An Approach of Change Impact Analysis in Web Systems

Zafar Mehboob (B.S. (Comp. Sci.), M.S. (Soft. Eng.))

A Thesis Submitted for the Degree of

MASTER OF SCIENCE IN COMPUTING SCIENCES (RESEARCH)

University of Technology Sydney
Faculty of Engineering & Information Technology

2013

Certificate of Original Authorship

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Production Note:
Signature removed prior to publication.

Signature of Student:

Date: 26-03-2013

Acknowledgement

This thesis was completed with the support and guidance of a great number of people whose contribution in various ways to the research and the making of the thesis deserved special mention.

I would like to express my sincere gratitude to Professor Didar Zowghi for her understanding, encouragement and personal guidance, provided me a good basis to conduct this research. I am grateful to Professor David Lowe for his detailed and constructive comments, and his valuable advice to improve the quality of this thesis.

I owe my gratitude to Professor Jaya Kandasamy and Dr. Paul Kennedy Paul for their support, guidance and constructive discussions to finish my research work.

I warmly thank to all the participants took active part in interview based study. I owe deep gratitude to Mr. Mohammad Aamir Khan who gave me the opportunity to conduct the case study in his organisation. I thank to Professor Marco Brambilla from Politecnico di Milano, Italy, for his valuable advice and friendly help in relation to exemplar system study. I am thankful to him for giving me insights into WebML modelling software and provide timely feedback on the WebML design Models. I am truly grateful to my father, mother, brothers and sister for their best wishes, prayers and support.

Lastly, and most importantly, I would like to dedicate this thesis to my father and my dearest mother who formed part of my vision and taught me to keep learning and taking on greater challenges. It would not have been possible to finish my research work without their understanding, patience, guidance, and continual support.

Preface

Some of the work described in this thesis has been published previously in several conference proceedings and international journals:

1. (Mehboob, Z. et al., 2007) describes preliminary research direction, based on the investigation from the state of the art, as discussed in Chapter 2
2. (Mehboob, Z. et al., 2008) describes research investigation toward change impact analysis in Web systems and research scope based on the investigation from the state of the art, as discussed in Chapter 2.
3. (Mehboob, Z. et al., 2009) describes a comparison study and the findings from the comparison of existing architecture level CIA approaches and their relevance in Web systems, discussed in Chapter 2.
4. (Mehboob, Z. & Zowghi, D. 2009) discusses the perspectives on Architecture level Change Impact Analysis in Web Systems Evolution, leads to the findings from the state of the practices in relation to change impact analysis approaches in Web systems projects, discussed in Chapter 4.
5. (Mehboob, Z. et al., 2011) presents a Process Model of Change Impact Analysis (PMCIA) for Web Systems Using Design Knowledge. This work is discussed in Chapter 5.
6. (Mehboob, Z., 2013) presents a comparison study for PMCIA with other similar approach proposed to perform CIA for Web systems.
7. (Mehboob, Z. & Khan, A. M., 2013) presents case study design in brief and investigation results in relation to validate PMCIA, discussed in Chapter 6 and 7.

Table of Content

List of Figures

List of Tables

Abstract

There are currently a high number of Web systems being developed; therefore, any consideration towards improving Web systems development practices will be worthy of research effort. Web systems are intricately interwoven into business processes so as to be supported by those systems. In other words, there are a multitude of fined-grained interconnections between business processes and the supporting architecture. Keeping in view the fine-grained interconnections, any change made in business processes can typically lead to more pervasive changes at the architecture (as compared to traditional software systems). Consequently, there is a much more fine-grained evolution at the architecture level in order to address the intricately interwoven characteristic. Partly as a consequence of the intricately interwoven characteristic, the connection between business processes and the architecture solution are also much tighter than for traditional software systems. Given this tight-connection, any change in business processes often leads to fundamental changes at the supporting architecture. Additionally, while addressing changes in business processes, a single change in architecture may have substantial impacts on other parts of the architecture and result in the ripple effects of the change being made to architecture.

The intricately interwoven characteristics, underpinning fine-grained evolution and the tight connection between business processes and the architecture design tend to differentiate Web systems from traditional software systems. To support the evolution and tight connection between business processes and architecture design, it would be beneficial to identify what needs to be modified in architecture to address business processes changes. Failure in attempts to identify change impacts on the architecture resulting from business processes changes leads to the problem where implementation (detailed design) begins before the impacts (on architecture) are adequately identified. As a consequence, many of the change impacts caused by business processes changes may go undetected or are identified very late and often lead to unnecessary re-work during the later stages of system development. We will refer to the identification of change impact on architecture before detailed design actually begins as "*early identification of change impacts*" in Web systems.

As presented in this thesis, the investigation from current state of the practices form a set of high level features/needs that a change impact analysis approach should support for Web systems. Keeping in view both the industrial perspectives and the issue where implementation (detailed design) actually begins before the change impacts (on architecture) are adequately identified, we proposed a systematic, structured and rigorous approach termed as the process model of CIA (PMCIA). We have captured various elements and necessary components derived both from the state of the art and state of the practice for the development of the process model of CIA. We have validated it with an industrial case study. The case study findings indicate that with the use of process model of CIA, practitioners indeed get support for early identification of change impacts in Web systems project and acquire an increased understanding of employing design information during change impact analysis.

Chapter 1: Introduction

1.1 Overview

Web Systems are characterised by continuous development so as to adapt to rapidly changing needs. Inadequate support during the development of Web systems is considered to be one of challenges that contributes to the failures of Web systems (Murugesan, 2008). A key area that is not well understood during Web systems development is the way in which changes in business processes can impact on the underlying architecture design. This understanding is particularly crucial, given that Web systems are intricately interwoven with business processes (Bucknell et al., 2008, Yusop, 2009) and the design of Web systems is highly tied with those business processes (Torres and Pelechano, 2006, Paiano et al., 2009a). In Web systems, the connection between business processes and architecture design is more stressed than traditional software systems because any potential change in business processes can lead to fundamental changes in the supporting architecture design (Lowe and Henderson-Sellers, 2003). The impacts of business processes changes on the architecture design need to be adequately identified before it becomes costly (due to unnecessary re-work) to address these impacts at subsequent stages of Web systems development. In this chapter we will provide an overview of a problem in relation to change impact analysis (CIA) for Web systems – where the detailed design and implementation actually begins before change impacts are adequately identified. We argue that this problem can be addressed in part through the development of a process model of CIA to support early identification of change impacts in Web systems. We discuss the key research questions which will be considered in this thesis. We also outline the structure of this thesis and finish by describing the contribution it makes to the body of knowledge in the area of Web systems development.

1.2 Motivation

In a span of two decades, Web systems have emerged as fully functional systems that provide many different types of services. Users of Web-based business have grown substantially in the last few years (Stats, 2009) and the Web has been employed across various domains including commercial, educational, governmental and entertainment sectors. According to statistical data reported by Gartner research, the number of users and investment in Web-based business has been growing at a phenomenal rate (Gartner, 2010b) and it has been forecasted the same for the next coming years (media, 2010, Gartner, 2010a). In particular, industry continues to commit significant expenditure in utilising the Web for developing an online presence for their business (Gallen, 2010) and for migrating key business processes to Web systems (Aversano et al., 2001, seacord et al., 2003). Web systems have evolved with the emergence of new technological innovations that have extended the role of Web systems from presentation of information to functionally complex Web-based business systems (Rossi et al., 2003, Torres and Pelechano, 2006, Paiano et al., 2009a). Web systems leverage the infrastructure of the internet and increasingly complex set of standards, protocols and technologies to provide sophisticated business solutions that merge Web-based front-ends with complex back-end softwares (Lowe and Eklund, 2001, Lowe and Henderson-Sellers, 2001a, Paiano et al., 2009a).

Software architecture is considered important for the success and quality of the systems being developed (Hofmeister et al., 2000, Bass et al., 2003). IEEE 1471 defines software architecture as, "*The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution*" (IEEE, 2000). However, architecture has not received a commonly agreed description in the context of Web systems. A number of views for architecture design has been described, such as the "4+1" model view of architecture (Booch, 2001b, Kruchten, 1995), different perspectives in Web Application Architecture framework (Kong et al., 2005a) and others (this will be described in Chapter 2: details of what is architecture and its different perspective in the context of Web systems). However, most of the work done in relation to Web systems architecture has a specific focus on modelling and representing architecture design to address different perspectives, e.g. user interface design, information design, functional design and application design, etc. Consequently, a number of Web design languages and approaches have been developed to address single or a combination of architecture design perspectives. A good discussion on Web modelling languages and approaches can be found at the following reference (Barna et al., 2003, Wright and Dietrich, 2008, Paiano et al., 2009b).

Current developments in the context of Web systems tends to support complex business processes, spanning multiple users and organisations (Bochicchio and Longo, 2004, Murugesan, 2008). These business processes are integrated with architecture design, thus, enabling the implementation of the functionalities, which are supported and delivered by Web systems (Aversano and Tortorella, 2009). The integration between business processes and architecture design implies that business processes needs to be tied with architecture design to extend the Web systems scope from information management to process management (Torres and Pelechano, 2006). Subsequently, the focus towards process management leads to a high number of Web systems development to support Web-enabled business. Despite the high number of Web systems being developed, there is often very little understanding of the way in which architecture design is integrated with business processes (Paiano et al., 2009a) and how Web technologies can be effectively utilised to support these business processes. The result is the developments of systems that often do not fully satisfy business needs and incur both a significant development cost and critical loss of business opportunities (Chaos, 2009).

There are many examples of Web systems ending up with high cost, an overrun schedule, and sometimes resulting in business failures. Cutter Consortium in 2000 found that the top problem areas of large-scale Web systems are the failure to meet business needs (84%), schedule delays (79%), budget overruns (63%), lack of required functionalities (53%), and poor quality of deliverables (52%). According to another report (Gallaher et al., 2004), it is estimated that a high cost incurred due to faults in Web systems, just in the capital facilities industry in the U.S. alone is $15.8 billion per year.

The reasons behind Web systems failure described by Ginige and Murugesan (2001b) include a lack of vision, flawed design and development process, and poor management of development efforts. Additionally, a lack of understanding of the ways in which changes in business process can impact on architecture design can potentially also play a major role toward Web systems failures. This lack of understanding mainly leads to problems in relation to change impact analysis – where the implementation actually begins before change impacts are adequately identified. As a consequence, generally project complexity and cost are increased and specifically time gets wasted in unnecessary re-work during subsequent stages of Web systems development. To address this problem, CIA approaches should allow Web developers to perform early identification of change impacts in Web systems.

One activity during Web systems development that has received increasing attention in the last decade is the modelling of Web systems by focusing on different design aspects (Paiano et al., 2009a). Among these aspects, presentation and free-navigation are surely important, but conditional navigation design (Distante et al., 2007b), information design (Ceri et al., 2000a) and transactional design (Distante and Tilley, 2005) must also be considered to adequately integrate business processes with system design. The need to focus on different design aspects of Web systems have emerged so as to address different elements of business processes including activities, workflows, control flows, operations and user's context (Paiano et al., 2009a). Each of these business process elements are taken into account during modelling of different design aspects of Web systems and thus it likely to build a connection between business process elements and architecture design (this will be discussed in detail: this connection as a highly tied scenario). Given the connection between business process and architecture design, any potential change in business processes can have substantial impacts on architecture design and these impacts need to be addressed before detailed design or implementation phase actually begin. It means that architecture designing can be considered as an appropriate phase to identify the possible change impacts (on architecture) resulting from business processes changes.

1.3 Problem Establishment

Research literature indicates that there is little research focus on identifying the impacts of business process changes on Web systems architecture/system design- we refer to it as early identification of change impacts in Web systems. There are a number of CIA methods, approaches and tools developed and reported in research literature. However, one of the areas which are not well supported by existing CIA approaches is understanding the way in which potential impacts of business process changes can be identified at architecture design. Failure in attempts to identify change impacts on architecture design lead to the problem where implementation begins before the impacts (on architecture design) are not adequately identified (Williams and Carver, 2010). As a consequence, many of the change impacts caused by business processes changes may go undetected or identified very late (Kagdi and Maletic, 2006) and potentially lead to unnecessary re-work during the subsequent stages of system development (Michael, 2004).

Web systems are characterised by fine-grained evolution during their development life cycle (Ginige and Murugesan, 2001a, Murugesan et al., 2001, Lowe, 2003), and therefore evolution in Web systems plays an increasingly important role in comparison with traditional software systems. Typical reasons for Web systems evolution are frequent changes in the environment (e.g. business partners and competitors), changes in business processes, user expectations, new operational environment needs, statutory and regularity requirements (Lana S. Al-Salema and Samahab, 2007), the availability of new technologies, methods, tools (Ramler et al., 2004), and the like. In order to aid Web systems evolution, systems need to be understood at different levels of abstraction and examined from different perspectives (Murugesan, 2008). We believe that one of the important perspectives is to understand the way in which the impact of business process changes can be identified on architecture/system design.

One crucial aspect during Web systems development that has been reported in various research literature is the ability to ensure that the connections between business processes and architecture design are adequately understood (e.g. Rossi et al., 2003, Knapp et al., 2004, Koch et al., 2004, Tongrungrojana and Lowe, 2004a, Bochicchio and Longo, 2005, Brambilla et al., 2006, Torres and Pelechano, 2006, Distante et al., 2007a, Distante et al., 2007b, Brambilla et al., 2008, Aversano and Tortorella, 2009, Paiano et al., 2009a). Further, understanding these connections and establishing the integration between business processes and architecture design ensure their compatibility and alignment (Aversano et al., 2005, Bucknell et al., 2008). A necessary first step in understanding the connection between business processes and architecture design is to explore the related characteristics of Web systems.

The characteristics of Web systems are both organisational and technical in nature. For example, organisational characteristics include a considerable degree of uncertainly in Web systems project domain and volatility of both the client needs and available technology (Lowe and Eklund, 2002). Fundamentally, however, there are some of the technical characteristics that are highlighted in Web systems (Burdman, 1999). Possibly the most significant is the combination of two factors: Web systems are interwoven with business processes (Bucknell et al., 2008, Yusop, 2009) and Web systems design are highly tied with business processes (Torres and Pelechano, 2006, Paiano et al., 2009a)

During the Web systems[1] development, each element of business processes are often connected to the compositions of design fragments (Paiano et al., 2009a, Brambilla et al., 2006), to the navigational concerns (Gordillo et al., 2006, Reina-Quintero, 2008), to the information units (Tongrungrojana and Lowe, 2004a), and to the details specific to the server side components (Kong et al., 2005b). The connection at the level business process elements to different aspect of system design implies that, in Web systems the connection between business processes and architecture design are much more fine-grained. Further, there are multitudes of interconnection between business process and system design. These multitude of interconnection are implemented by treating each business activity as a potential Web service, identify activity and Web service dependency, and model activity and Web service connection during architecture design (Inaganti and Behara, 2007). Fine-granularity and multitude of connections between business processes and architecture design guide us towards a characteristic where business processes and Web system are reported as intricately interwoven - a distinct characteristic of Web systems (Yusop et al., 2006, Bucknell et al., 2008). Conversely in traditional software systems, business processes are interwoven with architecture design (Nuseibeh, 2001) and there is a coarse-grained connection between business processes and architecture/system design (Nuseibeh, 2001, Pohl and Sikora, 2007).

One key activity during Web systems development that has received increasing attention in the last decade is Web systems modelling languages and approaches. Web systems modelling languages provide sophisticated notations to facilitate the understanding, specification, documentation, visualisation, communication and construction of important architecture design during the course of Web systems development (Tongrungrojana and Lowe, 2004a). Most of the Web systems modelling approaches were developed with the aim of successfully modelling information and

[1] Here by Web systems we mean systems that are developed to enable organisations to perform tasks and functions over the Web based on the underlying business processes; and utilise Web technologies as an integral element of a functionally complex system.

content (Baresi et al., 2001, Gomez et al., 2001, Schwabe et al., 2001, Ceri et al., 2002), whose main focus is to store, retrieve, transform and present information to the users. However, this concept has been changed by posing additional modelling requirements for Web systems that have supported these systems to be emerged from information management to process management applications (Distante et al., 2007a). In order to address process concerns, Web modelling approaches have been extended to model the integration of business processes with system design, thus enabling to adequately model Web systems.

Furthermore, in order to integrate business processes and system design, Web modelling approaches focus on different design aspects such as informational, navigational, and transactional. Example of these Web modelling languages and approaches are WebML (Ceri et al., 2000b), UWA+ (Bochicchio and Longo, 2004), UWAT+ (Distante et al., 2006), WFdHT (Brambilla et al., 2006) and WIED (Tongrungrojana and Lowe, 2004b), etc. Addressing informational, transactional and navigational design tends to create a strong connection between business processes and system design, given that information flows are derived from business workflows (Ranganathan and Ganapathy, 2002) transactions are derived from business operations (Distante et al., 2007a, Distante and Tilley, 2005); and navigation and user interactions are derived from business process definition, control flow and the user's context (Torres and Pelechano, 2006, Rossi et al., 2003). It implies that the design of Web systems is highly tied to business processes. This strong connection between business processes and architecture design is potentially due to the following reasons:

- Web systems that are based on underlying business processes support and guide the users through business workflow (Koch et al., 2004, Ceri et al., 2002). Integrating hypertext with business processes workflows means delivering user interface flow that permits the execution of business activities, and the embodied constraints as well as control flow from business processes that derive the navigation in the system (Brambilla et al., 2003). At systems design, control flow from one business activity to subsequent activities is represented by navigational semantics (Rossi et al., 2003) and business processes' constraints are enforced via hypertexts by employing conditional navigation (Brambilla et al., 2006). In Web systems, navigation tends to support the users to decide which process execution path to follow and which content to navigate through, and indeed it is business processes which derive the users both to path selection and next content in order to accomplish a process execution (Torres and Pelechano, 2006). Thus, Web systems allow both free (to a limited extent) and conditional navigation to support user navigation across the system, which is one of the most distinctive features of Web systems (Paiano et al., 2009a). There has been a considerable amount of work related to the specification and implementation of business processes in traditional software systems such as work-flow applications (Manolescu et al., 2000), and supporting the complex interactions between distributed processes (such as Business Process Specification Schema, www.ebxml.org); however, the interplay between processes and navigational paradigm is distinct in the context of Web systems (Rossi et al., 2003).

- Web systems are attributed by the information and information design. Information refers to the contents related to the features or services (focusing upon the core-business of the organization) that are supported by the Web systems (Janssen et al., 2008). While information design is the way by which the information is presented to the users (Janssen et al., 2008). The purpose of information design and modelling is the representation of relevant information to be handled by the system, and the provision of an overall organisation of the information structures (Baresi et al., 2006) and information flow (Ranganathan and Ganapathy, 2002). While connecting business process with system design, it is needed to model not only the information itself, but also representing information unit relationships, and external and internal information exchanges from business processes to architecture (Tongrungrojana and Lowe, 2004a). Indeed, information modeling depicts how information propagates across the organisation and exposes the potential for adding value through processing of information (Loshin, 2003). This information modelling tends to be much more complex than traditional data modelling (which used approaches such as entity-relationship and data flow modeling) since representing the information context becomes much more important in Web systems (Brambilla et al., 2003). The mapping between information flow and business processes (and workflow) support to bridge informational and functional aspect of the architecture (Tongrungrojana and Lowe, 2005). In particular, information structures are derived from business processes entities (e.g. users, inventory etc.) and information flow is derived from business workflow (Ranganathan and Ganapathy, 2002) and user interface flow (Kong et al., 2005c). It implies that there is a strong connection between business processes and architecture design of Web systems.

- Transactions design in Web systems is perceived as managing the interplay between business processes execution and system navigation, thus improving the user's experience in accessing the functionalities that Web systems offers (Distante et al., 2007b). A transaction is defined as a sequence of business operations with the associated execution flow that enables the user to accomplish a task and/or reach a specific goal (Distante et al., 2007b). In other words, a transaction is implemented in terms of business processes via sequences of operations (functional activities) and navigation steps (navigational activities) through the pages of the application (Distante and Tilley, 2005). It means that transaction design can be described in terms of operational and navigational capabilities of the system (Distante et al., 2007b). Consequently, modelling transaction in Web systems tends to create a strong connection between business processes and systems design.

Though many of the characteristics of the Web systems may be observed in traditional software systems; however, the intensity with which they apply and together the characteristics make Web systems distinguishable (Ramesh et al., 2002). Moreover, it has also been reported that some of the development characterisers are either unique or highlighted in Webs systems (Burdman, 1999).

As discussed before, one of the two characteristics of Web systems is that Web systems are intricately interwoven with business processes (Bucknell et al., 2008, Yusop, 2009). This interwoven characteristic tends to create a multitude of fined-grained interconnections between business processes and architecture design that are supported by Web systems. Secondly in Web systems, architecture design is highly tied with business processes (Torres and Pelechano, 2006) as discussed in the last couple of paragraphs. Given the multitude of fined-grained interconnections and tight connection between business processes and system design, it can be argued that potential change in business processes can lead to fundamental change in the underlying architecture design (Lowe and Henderson-Sellers, 2003).

Failure in attempts to identify change impacts on architecture design leads to the problem where implementation begins before the impacts (on architecture design) are not adequately identified (Williams and Carver, 2010). As a consequence, many of the change impacts caused by business processes changes may go undetected or identified very late (Kagdi and Maletic, 2006) and lead to unnecessary re-work during subsequent stages of system development (Michael, 2004). This unnecessary re-work mostly incurs due to poorly updated system design, where the impacts of business processes changes are not adequately identified on system design, thus leading to poorly implemented system. Boehm and Basili (2001) discovered through their research that finding and fixing a software problem after delivery was often 100 times more expensive than finding and fixing it during the requirement and design phase. Having an updated architecture design before implementation appeared to incur fewer defects (Cusumano et al., 2003). It has also been urged that software developers have less chance of introducing defects and fewer code changes (Cusumano et al., 2003) if an updated architecture design is available. One of the reasons for fewer defects at later stages of Web systems development is probably because projects could make an early adjustment based on change impact analysis results (as opposed to finding out later that some things were wrong) (Cusumano et al., 2003). It implies that early identification of change impacts allows keeping architecture design updated along with business processes changes, and support to reduce unnecessary re-works at subsequent stages of Web systems development. Indeed, a fruitful area for further investigation is to look at how to identify impact on architecture design caused by business processes changes? Analysing the impacts on architecture design resulting from business processes changes will allow Web developers/designers to address the problem in relation to change impact analysis – where the implementation actually begins before change impacts are adequately identified.

Surprisingly, most of the change impact analysis works have been done on technical aspects, which focus on the development of methods and tools for assessing the impacts of changes on code level. As reported by Jonsson (2007), one drawback of purely technical aspect is that such research often assumes the presence of a detailed, structured and supportive infrastructure (such as supporting tools, documents, activities to follow etc.) to apply change impact analysis approaches. This calls for an investigation of how CIA, specifically architecture level CIA (one of the area's where we are interested in) is currently practiced and what are the features/needs of the change impact analysis approach that it should possess in order to support early identification of impacts.

Related work done on architecture level CIA includes, for example, the work done by Zhao et al. (2002) by using architectural specification. Zhao et al. (2002) work focuses on how to find backward, forward, and unified architectural slices/chops from architecture specifications and to execute CIA process. Their process consists of three phases which include building the architectural flow graph, finding slices and chops over the graph, and mapping the slices and chops to architectural specification. However, Zhao et al. (2002) work restricts the changes and their impact analysis at the level of architecture only, not across multiple levels from business process to architecture design, etc. Their work also does not cover the steps and activities in details and other infrastructure support to carry out the proposed CIA process approach. Additionally, other related work proposed by Berg (2006) and by Tang et al. (2007) do not cover process aspect such as details of steps to follow, flow from one step to another, inputs/outputs, and activity descriptions. The paucity of process based research work in relation to architecture level CIA motivates us to focus on a process model of CIA.

Additionally, it has been reported by Jonsson (2007) that the basis, of which there are three, for the analysis of change impacts were gut feeling, experience and competence (as opposed to models or other documentation). Any analysis made based solely on subjective basis is bound to be person dependent. In the worst case scenario, this kind of analysis may be inconsistent with what others would produce. These aspects also direct us towards a need for process support during change impact analysis. Process support such as steps or activity descriptions could result in a more objective and well-founded analysis for change impacts, and also decrease the risk that something is missed during the analysis. The process aspect concern in our research has the same goal of objective and well-founded analysis to support early identification of change impacts.

Current research literature and industrial practices have highlighted a smaller focus on the process aspect of change impact analysis (Jonsson and Lindvall, 2005, Jonsson, 2007). Investigation of Web systems commercial practices also indicates that CIA approaches are mostly informal, unstructured and ad hoc (Mehboob and Zowghi, 2009). Other important aspects derived from industry perspectives are a lack of mutually agreed approaches, inadequate description of sets of steps, and lack of necessary guidelines and inputs required to carry out change impact analysis for Web systems. Based on the findings both from the state of art and the state of practice, we have focused on a process model of CIA as a possible extension of current CIA approaches to support early identification of change impacts in Web systems.

1.4 Research Objectives/Questions

The preceding discussion leads us to propose the following research question:

> *Can a change impact analysis approach that explicitly addresses characteristic of Web systems be used to support early identification of change impacts in Web systems?*

The research objectives deriving from the research question are:

Research Objective 1: To analyse the characteristics of Web systems that explicitly need to be considered during Change impact analysis (CIA)

Research Objective 2: To investigate scholarly/research literature for change impact analysis including the existing approaches, methods, techniques and tools, and to identify gaps in the scholarly/research literature while adopting CIA approaches in Web systems.

Research Objective 3: To investigate industrial practices of CIA and to identify the limitations of existing CIA (in practice) for Web systems.

Research Objective 4: Based on the reviews of scholarly/research literatures and practices, extend the current theory by proposing an extension of existing CIA approaches to support early identification of change impacts in Web systems.

Arising from the research objectives above, the research questions are defined as follows.

RQ1: What specific characteristics of Web systems need to be considered for CIA and why?

RQ2: What types of changes and their impacts are of interest in Web systems?

RQ3: Why is CIA important for Web systems?

RQ4: What are the relative strengths and weaknesses of the existing CIA methods, techniques and tools while adopting these CIA approaches in Web systems?

RQ5a: What does it mean by early identification of change impacts in Web systems?

RQ 5b: Why is early identification of change impacts important in Web systems?

RQ6: What are the industrial practices of CIA in Web systems?

RQ7: How do the existing CIA practices and approaches support early identification of change impacts?

RQ8: What kinds of information do Web developers currently use to perform change impact analysis?

RQ9: What are the high level features that CIA approaches should address in Web systems?

RQ10: Based on the review of scholarly literature and practices, is it possible to extend existing CIA approaches?

RQ11: Does the process model of CIA address the specific characteristics of Web systems?

RQ12: Does the process model of CIA support Web developers/architect in early identification of change impacts?

1.5 Proposed Approach and Justification

Commercial Web development practices are typically either ad hoc or reflect the disciplinary background of Web developers without adequately understanding the unique characteristics of Web systems (Lowe, 2003). Inadequate understanding of Web systems characteristics partially leads to Web developers/designers' inability to understand the ways in which potential changes in business processes can impact on the architecture/system design. Further, inability of this understanding results in identification of impacts during subsequent stages of Web systems development that consequently incurs unnecessary re-work and high costs during Web systems development. Thus, we refer to the problem in relation to change impact analysis – where the implementation actually begins before change impacts are adequately identified. To deal with this problem during the Web systems development, we have focused our research on "early identification of impacts". By early identification of impacts we mean the analysis of the impacted area of architecture design before it will be implemented into detailed design. We have limited our research scope of the "identification of impacts" to:

a. Impacts on architecture design caused by business process changes (direct impacts or primary impacts), and
b. Impacts on architecture design caused by the ripple effect of changes being made to the architecture design (indirect impacts or secondary impacts)

Research literature indicates that both primary and secondary impacts are important to identify, (Bohner and Arnold, 1996) and CIA approaches should cover the identification of both to ensure an adequate identification of impact set. Therefore, this research on "early identification of impacts" tends to cover both impacts on architecture design (i) caused by the changes in business process and (ii) caused by the changes being made to the architecture/system design.

Web systems development and engineering practices often require systematic, structured, rigorous and a quantifiable approach of 'how' to address certain aspects during system development (Knapp et al., 2004). Similarly, while proposing a CIA approach for Web systems, the question of 'how' guides us toward a process approach of CIA. As it has been discussed in Section 1.3, the work done in relation to the process aspect of CIA such as work proposed by Berg (2006), Zhao et al., (2002) and Tang et al., (2007) does not cover process aspects such as details of steps to follow, flow from one step to another, inputs/outputs and activity descriptions. The paucity of process based research work in relation to architecture CIA has created the motivation to focus on a process model of CIA.

The main objective of this research is to extend the theory and improve the current practices of CIA by investigating both CIA research literature and commercial practices in Web systems context. The core research question is developed to form the basis for multiple research questions that will be addressed by research methodology.

1.6 Contributions

The goal of this research was to develop a CIA approach for early identification of change impacts in Web systems. The major contributions of this thesis are briefly listed below and further discussed in detail in the conclusion chapter.

- The first contribution of this thesis is to report the investigation findings of current state of practices in relation to CIA for Web systems. This investigation gives an insight of industrial perspectives on the use of current architecture level CIA approaches and particularly, a set of high level features/needs that a change impact analysis approach should support for Web systems.

- The second contribution of this thesis is a process model of CIA specifically developed for and validated in a Web systems project. This process model supports a structure, rigorous and consistent approach toward change impact analysis focusing on steps, activities in each step, inputs and outputs and the validation of the impact set produced from the process model. The novelty of this process model is the extension of the current CIA approach to support early identification of change impacts in Web systems.

- The third contribution is the demonstration of how case study research can be applied to the validation of the process model of CIA, and its suitability for validating new approaches in real Web project context. Findings from the case study address the problem in relation to change impact analysis - where the implementation (detailed design) actually begins before change impacts are adequately identified, and thus it becomes prohibitively complex and expensive to address change impacts in subsequent stages of Web systems development. The findings also indicate that the likelihood of identification of impacts during subsequent stages of Web systems development will be substantially reduced as Web architects/designers will have a better understanding of impacts on architecture resulting from business process changes.

Additionally, the following minor contributions have also been made:

- The identification of key research areas in relation to the change impact analysis for Web systems, in terms of what has not been addressed by existing CIA methods and approaches.
- An improved understanding of Web systems characteristics considered important for change impact analysis.
- An improved understanding of early identification of change impacts-identification of impacts on architecture caused by the changes in business process as well as those caused by the changes being made to the architecture/system design.
- An example of how the proposed process model for CIA can be used for early identification of change impacts.
- The role and importance of employing design information during the change impact analysis.

1.7 Research Approach and Thesis Structure

A research methodology used in this research combines a theoretically based analysis, an interview-based investigation and a case study-based validation. We have captured various elements and necessary components derived both from the state of the art and state of the practice for the development of the process model of CIA.

The first stage of the research was a literature review which involved a thorough examination and critical analysis of scholarly literature, commercial practices, existing theory on and around the area of change impact analysis and Web systems development. Core research questions and sub-questions were then developed based on performing critical literature review and content analysis. In particular, the literature review contributed in guiding very initial scope (Mehboob, 2007, Mehboob et al., 2008) and further articulating context encompassing identification of impacts on architecture design of Web systems (Mehboob et al., 2009) and the possible use of design information while performing a change impact analysis activity. The primary objective of this review was to establish a theoretical foundation for our research, as discussed in Chapter 2.

In Chapter 3 we describe our research methodology. The discussion in this chapter covers the approaches, theoretical perspectives and measures taken to address the research reliability, validity and limitation of the methodology employed. Mainly the research methodology used in this research combines the theoretically based analysis, the interview-based investigation of current practices, and the case study-based validation of the research artefacts (process model of CIA). Additionally, research objectives, research questions and research strategies are mapped to make sure that research questions meet the research objectives.

The second stage of the research was the interview-based investigation of current practices in relation to the change impact analysis for Web systems. An important source of information for investigating the state of practice is from those who make decisions for architectural changes on a regular basis. Therefore, we have conducted an interview based study with Web developers/architects who have experience in architecture design changes and change impact analysis area. Firstly, the findings from these interviews were helpful to validate the theoretical foundation of our research (established from the literature review). Secondly, interview findings were also helpful so as to deeply understand high level features/needs for the possible extension of the current change impact analysis approach (Mehboob and Zowghi, 2009). These findings and the classification of high level features/needs are reported in Chapter 4.

The triangulation of results from the two methods (i.e. literature review-state of art and interviews-based investigation-state of practices) were important for the integrity of the research. Indeed, it was not possible to depend entirely on the literature due to the work needed to be grounded in practice, and also due to the general lack of adequate evidence and sufficient industry-based publications in the area of change impact analysis during Web system development. The results from Stage 1 (literature review) and Stage 2 (interviews-based investigation) of the research provided both the motivation and input for the development of the process model of CIA - the third stage of the research as an extension of current CIA approach (Mehboob et al. 2011). In Chapter 5, the background of the process model, development of the process model and its detailed description are provided.

A case study-based validation was chosen for the process model of CIA, as it allowed the validity of the research artefact to be evaluated in a situation where an experimental study was not suitable to be applied. According to the evaluation selection criteria described by Kitchenham (1996) and others (Bratthall and Jorgensen, 2002), we found a case study to be the most suitable choice for our research due to many reasons (these reason are described in Chapter 6). The case study design and methods for data collection and analysis are also presented in Chapter 6.

The process model of CIA is validated using case study design and the results from the case study work are presented in Chapter 7 (Mehboob and Khan, 2013). This chapter aims at demonstrating how the early identification of change impacts can be supported during Web systems, and thus describes the evaluation for the validity of research artefacts (process model of CIA).

Chapter 8 is the final chapter of the thesis, in which we describe the contribution of this research, further work and conclusions. The implication for researchers and practitioners are also presented in this chapter. Further, some relevant additional information based on our interviews and case study work is presented in Appendices A, B, C, D (for interviews) and F, G, H and I (for case study) respectively. In Appendix E, the exemplification as a proof for the demonstration of process model of CIA is presented.

Chapter 2: Literature Review

2.1 Introduction

In this chapter we review the relevant literature on and around the topic of Web systems and change impact analysis. The literature review begins with the investigation of Web systems, Web systems characteristics, nature of change in Web systems and Web systems development. This investigation resulted in the identification of Web systems characteristics that are important to be considered for change impact analysis (CIA). Further research investigation focused on CIA approaches and the importance of design information during CIA. It is intended that the literature review will provide a theoretical foundation for the research and the development of subsequent approaches to address our core research question. The core research question is: "*Can a change impact analysis approach that explicitly addresses characteristics of Web systems be used to support early identification of change impacts?*" We need to deconstruct the core research question and identify a set of research questions and these needs to be addressed by the investigation of scholarly research literature. They are as follows:

Can a change impact analysis approach...

(i.e. Can a change impact analysis approach that explicitly addresses characteristics of Web systems be used to support early identification of change impacts?)

RQ1: What change impact analysis approaches currently exist?

RQ2: What are the relative strengths and weaknesses of existing change impact analysis approaches?

... that explicitly addresses characteristics of Web system

(i.e. Can a change impact analysis approach that explicitly addresses characteristics of Web systems be used to support early identification of change impacts?)

RQ3: What characteristics of Web systems need to be considered for change impact analysis?

RQ4: What change impacts are we interested in to analyse in Web Systems?

... be used to support early identification of change impacts ...

(i.e. Can a change impact analysis approach that explicitly addresses characteristics of Web systems be used to support early identification of change impacts?)

RQ5a: What does it mean by early identification of change impacts?

RQ5b: Why is early identification of change impacts important?

We begin the literature review by investigating characteristics of Web systems in relation to CIA (Research Objective 1) and then focusing on existing CIA approaches while identifying the gaps with regard to the relevance of these CIA approaches for Web systems (Research Objective 2). It is important to note that the deconstruction exercise mainly helps us identify a set of research questions which underlie the core research question. The above mentioned ordering of the research questions is not important, and it is just a consequence of the structure the core research questions have. Therefore, before discussing RQ1 and RQ2, we discuss the specific characteristics of Web systems considered important in relation to CIA (*Research Question 3*) and the type of changes and their impacts in Web systems (*Research Question 4*). We aim not only to describe early identification of impacts (*Research Question 5a)* but specifically focus on why early identification of impact ought to be considered in Web systems (*Research Question 5b)*. Further, we describe existing CIA approaches (*Research Question 1)* and study the relative strengths and weaknesses of existing CIA approaches (*Research Question 2*) while adopting those approaches for Web systems. We also describe the importance of design decision information to be used during architecture level CIA. This will be followed by a summary of the entire chapter.

The results from this review of theory will be combined with the results from our investigation of the state of the practices of architecture level change impact analysis in Web systems context (Chapter 4). Investigation findings from the scholarly research literature and state of practices form the basis for the key components of our proposed CIA approach presented in Chapter 5.

2.2 Web Systems

Since the introduction of the Web in 1990, it has been evolving at a fast pace. The number of Web applications has been increasing as Web users and businesses realise the benefits that stem from globally interconnected Web applications. Companies, organisations, and academic institutions exploit the Web infrastructure to provide information and services to customers and users. When these Web applications are further enhanced by employing Web browsers, network technology and client server infrastructure, etc., they are termed as Web-based systems or Web systems (Pressman and Lowe, 2008).

The role of Web systems is considered important to the execution of business practices in many organisations. Considering various organisational businesses, Web systems tend to focus on different types of services (Mendes et al., 2006) and thus the scope and complexity of Web systems vary from small scale to large scale enterprise applications distributed across the internet, corporate intranet and extranets (Turban et al., 2010). Subsequently, Web systems have been developed as complex distributed applications which merge Web-based front-end with complex back-end systems, and provide the user the ability to perform various business operations through Web systems.

There are many descriptions of Web systems reported in research literature. Schwabe et al. (2000), considered a Web application as, "*a structured set of objects that may be navigated, and processed, in order to achieve one or more tasks"*. Another definition given by Mendes et al. (2006) is, "*An application delivered over the Web that combines characteristics of both Web hypermedia and Web software applications"*. Yusop et al. (2006), described Web systems as, "*systems which utilise Web technologies as an integral element of a functionally complex system which typically incorporates interfaces beyond the organisational boundaries*". Whilst the definition of Web systems varies, a particularly useful working definition is: "*Web systems are those systems that typically utilise Web technologies to provide a complex distributed front-end (often, though not universally, accessible through Web-browsers) combined with high performance back-end software systems that integrate the systems with critical business Processes*" (Lowe and Henderson-Sellers, 2003).

With the emergence of Web systems and their use in different sectors, there has been a considerable focus on different types of Web systems. However, there is no single or widely agreed categorisation. For example, at the level of user access, Web systems are widely categorised as internet, intranet and extranet systems. Ceri et al. (2002) have described five kinds of Web systems based on the core objective of these systems to address business models. The five classes of Web systems includes: (i) commerce site- core business is to sell products such as e-shops, e-malls, virtual marketplaces, e-auctions, (ii) content sites- gives users information about core businesses such as digital libraries, online magazines, recommending systems, (iii) service sites- offer services as a core business such as order-tracking site and so on, (iv) community sites- build socially shared businesses such as forums, chat rooms, newsletters, (v) context sites- helps to locate information (directories, search engines over local data).

Murugesan and Deshpande (2001) described six categories of Web systems based on functionality which includes *informational* (online newspapers, catalogues, newsletters, online books stores, etc.), *interactive* (membership forms, online games, etc.), *transactional* (online shopping, online banking, online air reservation), *workflow oriented* (online planning schedule, supply chain management), *collaborative work environment and online communities' marketplaces* (online discussion groups, online marketplaces, online auction, etc). Indeed, these categories are useful in understanding different functional contexts, and accordingly develop and deploy Web systems. Similarly, Mendes et al. describe three types of Web systems based on the delivery platform. These three types include *Web hypermedia application, Web software applications, and Web applications*. *Web hypermedia application* is typically characterised by authoring of information nodes, links to connect different nodes, anchor, navigation for access structure and the delivery of information over the Web. *Web software applications* are described as the applications executed by using Web infrastructure, for example, legacy information systems such as databases, online banking systems, knowledge bases, etc. *Web applications* are described as the application that employ characteristics of both Web hypermedia and Web software applications (Mendes et al., 2006).

Other authors such as Kalakota and Robinson (1999) and Turban et al. (2000) consider e-business (electronic business) and e-commerce systems as among the different types of Web systems. Laudon and Traver (2007) have described e-commerce as digitally enabled commercial transactions between and among organisations and individuals, involving an exchange of value across organisational boundaries. Whereas, Kalkota and Robinson (2003) have described that e-business encompasses internal and external electronically based activities, including e-commerce. However, Laudon and Traver (2010) have argued that e-business provides only infrastructure support for e-commerce application to perform commercial transaction, but does not directly generate revenue from outside businesses and customers (as e-commerce by definition does). It implies that the e-business definition refers to a broader spectrum not just facilitating commercial transactions, but also collaborating with outside business and providing services to customers.

With an increase in the number of Web-based businesses, Web systems are being considered as ubiquitous. There are various descriptions and types of Web systems reported in research literature based on different focuses and perspectives. However, the evolving trend in Web systems is towards complex functional systems to support various organisational businesses and services. Given the current focus of Web systems to support organisational business, we opt to use the description of Web systems provided by Lowe and Henderson-Seller (2003) as more suitable for our research study. We will refer to the same description of Web systems in the rest of this thesis.

To address increasingly complex functional Web systems, the importance of an adequate and well-structured development process also gets increased. As a result, a different set of development processes have been developed and reported by many researchers including Lowe and Eklund (2001), McDonald and Welland (2001), and Pressman and Lowe (2008). These development processes are constructed with a specific focus on a set of Web systems characteristics which distinguishes them from traditional software systems. In the next section we shall consider at some length, numerous works that describe Web systems characteristics as different from traditional software systems, and provide a foundation for Web systems as different systems.

2.3 Web Systems Characteristics

There are a number of characteristics of Web systems reported in the research literature that tend to differentiate Web systems from traditional software systems. It has been argued that the characteristics of Web systems as an application medium is not yet well understood (Murugesan et al., 2001). Specifically, there are a few characteristics of Web systems that are important to be considered in relation to the way in which Web systems change and the resulting impacts of that change on the system. In this section we scope out and describe those characteristics of Web systems and subsequently address ***RQ3: What characteristics of Web systems need to be considered for change impact analysis?***

In general, the characteristics of Web systems are both organisational and technical in nature. For example organisational characteristics include a considerable degree of uncertainty in Web systems project domains (Shina, 1999; Lowe, 2003), mutual constitution of problems and solution domains (Bucknell et al., 2008, Yusop, 2009), volatility of system requirements and available technology (Lowe and Eklund, 2002), co-evolution of business and solution designs (Yusop et al., 2005), and rapidly changing requirements (Lowe and Henderson-Sellers, 2001a).

According to Yusop et al. (2005) during the Web systems development, the business processes not only pose the requirements for the system to be developed but also these business processes change with the adoption or evolution of Web systems. These changes are such that not only do the system requirements change, but the very nature of the problem that was being addressed is changed. In effect, the business "problem" defines the nature of the technical system "solution" that is desired, but the solution itself changes the characteristics of the problem space. Subsequently, business processes and system solutions (Web systems) are mutually constituted (Yusop, 2009). This characteristic of mutual constitution could be considered both for Web systems as well as traditional software systems. However, this characteristic is likely to be more noticeable during Web systems development.

With Web systems, the technology, development skills and competing systems are changing so rapidly that the system domain is constantly evolving (Shina, 1999). This provides a significant degree of uncertainty in the system context, and consequently makes the determination of system requirements very problematic. On the other hand, business processes/models are evolving rapidly as organisations migrate to an increased reliance on Web-based solutions and technologies (Stein, 2000). As a result, system context and system solutions will often co-evolve considerably during the course of Web systems projects. The co-evolution of business processes and architecture design is thus more significant in Web systems, partly due to some of the technical characteristics of Web systems.

At a technical level, Web systems characteristics include such systems that are intricately interwoven with business processes (Bucknell et al., 2008, Yusop, 2009), business processes/model are tightly connected with architecture design (Lowe and Henderson-Sellers, 2003, Torres and Pelechano, 2006, Paiano et al., 2009), greater importance is placed on user interface (Lowe and Henderson-Sellers, 2001a), and a high demand is apparent on quality attributes (Lowe and Eklund, 2001, Lowe and Henderson-Sellers, 2001b, Lowe and Henderson-Sellers, 2001a). From these characteristics, possibly the most significant is the combination of the two: Web systems are intricately interwoven with business processes and business processes have a tight connection with architecture design. In fact, the intricately interwoven business processes with Web systems may lead to a tight connection between business processes and architecture design but the reverse (tight connection leads to intricately interwoven of business processes with Web systems) is not necessarily true. Therefore, we consider these two characteristics separately. In the next paragraph, we will discuss these two characteristics in detail and their role during Web systems changes and their resulting impacts.

To understand why Web systems are intricately interwoven with business processes, firstly we need to discuss business process. Davenport (1993) has presented a basic definition of business process as, '*a structured set of activities designed to produce a specific output*'. Furthermore Elguezabal (2005) has explained business process as, '*a collection of activities that are required to achieve a business goal and it is represented with an activity flow that specifies the orchestration needed to complete the goal*'. Other researchers have proposed the variation of the basic definition of business process while emphasising the fact that business process is a collection of various tasks which produce an output. More recently, Aversano and Tortorella (2009) have described business process as, '*A business process consists of the activities performed by an enterprise/organization to achieve a goal. Its specification includes the description of the business activities, related control and data flows'*. A business process can thus be described as a collection of elements including business activities, workflows, operations and control flows. In order to represent business processes and their underlying elements, significant research has been carried out to model business processes. For example, Business Process Modelling Notation (BPMN) has treated business activities, workflows, operations, control flows as different elements of business processes and introduced different types of modelling notations to represent each element (White, 2004).

In Web systems, business process elements are usually connected to different aspects of architecture design (Paiano et al., 2009a, Brambilla et al., 2006) including navigational design (Gordillo et al., 2006, Quintero, 2009), information design (Tongrungrojana and Lowe, 2004b), and server side components design (Kong et al., 2005a). As a result, there are a multitude of interconnections between business processes and the architecture design of Web systems. The multitudes of interconnection can be better understood from the research done in the area of Web services. Web services have emerged as a paradigm for implementing business process integration with Web systems (Aalst et al., 2007). Current practices of employing Web services help to decompose business processes until a detailed technical level of architecture design is reached and then Web services are integrated (with business processes) (Adam and Doerr, 2008). This is achieved from high-level business process modelling to a composition language which implements such processes with Web services technologies (De Castro et al., 2006). Each Web service attempts to complement individual business activity in order to provide the desired service to the users (Padmanabhuni et al., 2004) and subsequently model the underpinning connection between an activity and a service during architecture design (Inaganti and Behara, 2007). Since each business activity (from business process) is treated as an individual service (at architecture level); therefore, a considerable number of connections between business processes and architecture design are established. These multitude of connections between business processes and architecture design contribute towards Web systems characteristic where business processes and Web systems are reported as intricately interwoven - a distinct characteristic of Web systems (Yusop et al., 2006, Bucknell et al., 2008).

To adequately model and integrate business processes with architecture design, Web modelling approaches focus on a number of design aspects such as information, navigation, and transaction design. One of the reasons for focussing on different design aspects is to adequately understand, capture and represent the connection between business processes and the architecture design of Web systems. Taking into consideration three design aspects such as navigational, informational and transactional design, the connection between business processes and architecture design of Web systems will now be discussed.

Firstly, Web systems guide the users through business workflows and provide hypertext flexibility (Koch et al., 2004, Ceri et al., 2002) to decide which process execution path to follow and which content to navigate through. Additionally, control flow and constraints from business processes are also employed to design system navigation (Brambilla et al., 2003). For example, navigational semantics are derived from control flows (Rossi et al., 2003) and conditional navigations are derived from constraints (Brambilla et al., 2006). Navigation design allows both free (non-linear) and conditional navigation to support users while navigating the system and performing the related tasks, which is one of the most distinct features of Web systems (Paiano et al., 2009a). In traditional software systems, there has been a considerable amount of work done related to the implementation of business processes such as work-flow applications (Manolescu et al., 2000). However the interplay between business processes elements (such as control flow and business constraints) and navigation design is distinct in the context of Web systems (Rossi et al., 2003).

Secondly, Web systems are mostly attributed by rich information and subsequently demand a sophisticated information design. In Web systems, integrating business processes with architecture design is not only required to model the information itself, but also required to model the relationships between the information unit, external and internal information exchanges from business processes to architecture design and the way in which information is accessible and viewed by the users (Tongrungrojana and Lowe, 2004a). In Web systems, the information modelling is considered as much more complex than traditional data modelling (using approaches such as entity-relationship and data flow modelling), since representing the information and its flow becomes much more important in Web systems (Brambilla et al., 2003). The purpose of information design is the representation of relevant information to be handled by the system, and the provision for an overall organisation of information including information structures (Baresi et al., 2006) and information flow (Tongrungrojana and Lowe, 2004a, Ranganathan and Ganapathy, 2002). In particular information structure is derived from business processes entities (e.g. users, inventory etc.) and information flow is derived from both business workflow (Ranganathan and Ganapathy, 2002) and user interface flow (Kong et al., 2005a). Modelling of business processes elements (such as business entities and business workflow) at information design tends to create a strong connection between business process and the architecture design of Web systems.

Thirdly, transactions design in Web systems is perceived as managing the interplay between business processes execution and system navigation. Distante et al. (2007b) states: *"A transaction is defined as a sequence of business operations that enables the user to accomplish a task and/or reach a specific goal".* In other words, a transaction is implemented via sequences of operations (functional activities) and navigation steps (navigational activities) (Distante and Tilley, 2005). Thus, the resulting focus of transaction design is to model the operational and navigational capabilities of the system (Distante et al., 2007b). A major emphasis of modelling business operations at transaction design tends to create connections between business processes and the architecture design of Web systems.

To sum up, addressing different design aspects including navigational, informational and transactional together tend to create a strong connection between business processes and the architecture design of Web systems. This strong connection is potentially emerged while modelling different business processes elements (such as business control flow, constraints, workflows, operations, and business process definition) and their realisation at architecture design. For example, as discussed before, at architecture design information aspects of design flows are derived from business workflows (Ranganathan and Ganapathy, 2002); transactions aspects of design are derived from business operations (Distante et al., 2007a, Distante and Tilley, 2005); and navigation and user interactions are derived from business process definition, control flow and the user's context (Torres and Pelechano, 2006, Rossi et al., 2003). The discussion above supports the second characteristics of Web systems - i.e. there is a strong connection between business processes and architecture design of Web systems.

Many of the characteristics of Web systems may also be observed in traditional software systems, however, the intensity with which they apply distinguishes Web systems (Ramesh et al., 2002). Additionally, it has been reported that some characterisers are either unique or highlighted in Web systems (Burdman, 1999); therefore, adequate consideration of these characteristics will be beneficial for understanding and improving Web development practices. Investigation of the research literature indicates that the two characteristics of Web systems are important to be considered specifically for change impact analysis. These two characteristics include (i) Web systems are intricately interwoven with business processes (Bucknell et al., 2008, Yusop, 2009) and (ii) there is a strong connection between business processes and the architecture design of Web systems (Lowe and Henderson-Sellers, 2003). Further consideration of these two characteristics guide us toward the distinct dependency of architecture design on business process, to better understand the way in which business process changes can impact on the architecture design of Web systems.

The distinct dependency of architecture design on business process can be described in two ways. Firstly, given the multitude of connections between business processes and architecture design, the scope of the effect increases on architecture design incurred due to business processes changes. It has also been observed that slight changes in business processes can lead to massive system changes (Lana S. Al-Salema and Samahab, 2007) and can have a substantial effect on Web systems architecture design (Williams and Carver, 2010, Xiaoyu and Mei-Hwa, 2007). Conversely, in traditional software systems, business process changes can be either locally addressed at the business processes level or have a trivial effect on architecture design (Lowe and Henderson-Sellers, 2003). This is partially due to the coarse-grained connection between business processes and architecture design of traditional software systems. However in Web systems, the multitude of fine-grained connections between business processes and architecture design tends to distinguish the nature of dependency of architecture design on business processes.

Secondly, given the tight connection between business processes and architecture design, any potential change in business processes may fundamentally affect the architecture design of Web systems (Lowe and Henderson-Sellers, 2003). Here, the fundamental effect of change refers to the impacts on different aspects of architecture design. For example, business process changes are likely to impact on navigational design, informational design and transactional design. This means that the scale of business processes change impacts on architecture design could be of broader scope in Web systems. In the next section, we will discuss evolution, nature of change and types of changes in Web systems.

2.4 Evolution, Nature and Types of Change in Web Systems

In this section we specifically focus on types of changes and their impacts and address the research question ***RQ4: What change impacts are we interested in to analyse?*** In general, different opinions exist in relation to the concept of evolution during the system development life cycle. Some describe evolution as being encompassed by maintenance in a broader context (Sommerville, 2007), while others describe evolution as a separate stage in the system development life cycle (Bennett and Vaclav, 2000). There is also another view that describes software evolution as a way systems dynamically behave over their lifetime while those systems are being maintained and enhanced (Lehman and Ramil, 2001, Lehman and Ramil, 2002). Different descriptions of the system evolution imply that evolution can be seen as a much broader and encompassing term than software maintenance.

2.4.1 System evolution

In the context of Web systems, Lowe and Eklund (2002) compares the evolution of Web systems with landscape gardening as, "*The Evolution of Web applications are analogous to a garden changing as natural part of its cycle of growth*". They conclude that Web systems development differ mainly in a way that, "*We are no longer aiming to develop a finished product. Rather, we are aiming to create an organic entity that starts with an initial consistent structure, but continues to grow and evolve over time*". This implies that evolution in Web systems is considered to be an integral part of their lifecycle (Lowe, 2003) instead of large maintenance changes that normally occurred during traditional software system development. Further, the evolution in Web systems is much more fine grained (Lowe, 2003) when compared with the evolution in traditional software systems- where there tends to be coarse-grained response to software changes.

For Web systems, in many cases, it is hard to specify the complete requirements at the start of system development, because the underdeveloped system will either evolve over time or change dynamically during development. There are several reasons for changes in Web systems that typically include frequent modifications in the environment (e.g. business partners and competitors) and business processes (Lana S. Al-Salema and Samahab, 2007); new user requirements (stemming from intermediate development results presented to the user) (Lowe and Eklund, 2002); the availability of new technologies, methods, and tools; new opportunities to address business vision, strategies and objectives (Richard et al., 2002); and the need for refactoring and error correction throughout system development (Ramler et al., 2004).

2.4.2 Nature of changes

One of the important reasons for changes in Web systems is that business processes, and thus business requirements are continually changing to accommodate user expectations, new operational environment needs, statutory and regularity requirements, new business opportunities and the like (Lana S. Al-Salema and Samahab, 2007). This represents a key challenge during Web systems development where changes in business processes can have a high tendency to affect other systems artefacts such as architecture design, detailed design, code and test cases (Lana S. Al-Salema and Samahab, 2007). However, seemingly slight changes in business processes can effect on pervasive system assumptions, leading to massive system changes (Lana S. Al-Salema and Samahab, 2007). Some of the business processes changes can affect architecture design and have far-reaching consequences (Williams and Carver, 2010, Xiaoyu and Mei-Hwa, 2007) on the whole system.

It has been described in Section 2.3 that the ‘problem’ and ‘solution’ are mutually constituted during the course of Web systems projects. The concept of mutual constitution is also typical of other software developments, but it is particularly significant in Web systems. Whilst most traditional software systems development exhibits this characteristics, either the business process change impacts are sufficiently obvious to be taken into account relatively easily during initial development, or their scale is sufficiently constrained to allow them to be initially ignored and then to be addressed once the system has been implemented. With Web systems, particularly, the scale of impacts that business process changes can have on architecture design is sufficiently large and immediate and so they should not be overlooked during the initial stages of system development (Yusop, 2009). This is largely because of the degree and immediacy of the effects that business processes changes can have on architecture design (Lowe and Henderson-Sellers, 2003).

In Section 2.3 we also described that there are a multitude of connections from business processes to architecture design of Web systems. Given the multitude of connections from business processes to architecture design, changes made to business processes can effect on different aspects of architecture design including navigational design, informational design and transactional design. Hence, degree and immediacy of effects and wide spread effects on architecture design resulting from business processes changes tend to differentiate the nature of changes and their impacts in Web systems.

2.4.3 Types of change

A change may be classified and studied in several different ways. Bohner and Arnold classified the types of changes as, "*the type of change activity being performed, the kind of items being changed, and the motivation for the change"* (Bohner and Arnold, 1996). Type of change activities may include corrective, adaptive, and perfective changes (Mockus and Votta, 2000). Kinds of items being changed include business processes, requirement specifications, architecture design, code modules and test cases, etc. Motivation for change can be revealing from the reason of the change made (Bohner and Arnold, 1996). For example, typical reasons for design changes include business process improvements and business integration; design trade-off and elaboration; interface changes; scope and visibility issues; performance, timing and sizing issues; and feedback from prototypes. Similarly, typical reasons for program changes include bug fixes, algorithm-coding adjustments, arithmetic-precision modifications, data-structure modifications, initialization modifications, control and sequence changes, and parameter changes (Bohner and Arnold, 1996). More recently, Nurmuliani (2007) has described (requirement) that change classification comprises of change types, reason category and change origin. Here change types include addition, deletion and modification of requirements; reason category includes scope change, design complexity, product strategy, etc.; and change origin includes design review, product-integration testing, requirement review, etc.

It has been reported that most Web systems are directly derived from and influenced by strategic business vision and goals (Richard et al., 2002), and subsequently these systems are developed to support business processes presented by organisations (Aversano and Tortorella, 2009, Paiano et al., 2009). For this research study, the kind of item being changed is 'architecture design', where the identification of impacts are investigated resulting from business processes changes. Further, keeping in view the importance of business processes changes and their effect on architecture design, 'corrective and adaptive changes' will be considered as a type of change activity for this research.

The topic of evolution, nature of change and type of changes in Web systems are important, particularly in order to better understand the way in which changes in business processes can impact on architecture design. Research literature indicates that the nature of change is different such as the effects of business processes changes are sufficiently large and immediate on the architecture design. To manage changes in Web systems, system development approaches places a strong emphasis while addressing changes during the Web systems development. In the next section, we focus on the approaches that specifically support Web systems development. These approaches can be better examined by understanding Web systems design modelling and focusing on how business processes and their supporting architecture design are being constructed.

2.5 Web Systems Development

Web systems development often requires a holistic, systematic and disciplined approach to the successful development, deployment and evolution of high quality Web systems (Murugesan and Deshpande, 2001). The focus of Web systems development is to build a foundation for the systematic creation of these systems. This foundation consists of a body of theoretical and empirical knowledge for development, deployment, and support for continual evolution of Web systems. Deshpande et al. (2002) has highlighted the importance of Web engineering as a discipline to be adopted during Web systems development. Murugesan (2008) has argued that the essence of Web engineering is to successfully manage the diversity and complexity of Web systems development and hence, to avoid potential failures that could have serious implications.

There is a growing body of literature regarding the differences between the development processes of Web systems and traditional software systems. Numerous authors including Lowe and Henderson-Sellers (2001b), McDonald and Welland (2001), Murugesan and Deshpande (2005) and Mendes et al. (2006) have reported that Web development processes are different from traditional software processes in many ways. Deshpande et al. (2002) identified a number of differences between Web systems and traditional software systems. These difference typically include, *"compressed development schedules, constant evolution with shortened revision cycles, the embraced reality that 'content is king', insufficient requirement specifications and emerging technologies/methodologies"* (Deshpande et al., 2002). Other differences that have been reported includes: *"Web systems are developed in shorter timeframes, and with smaller budgets, meet a more generic set of requirements, and generally serve a less specific user group. They are often developed very quickly from templated solutions, using coarse-grained authoring tools, and by the efforts of a multi-disciplinary team*" (Lowe and Eklund, 2002). These differences point out the additional complexities of Web systems, and the challenges that Web developers face.

From the above mentioned differences and through a consensus among experts on the differences between Web systems and traditional software systems development (Deshpande and Hansen, 2001, Deshpande et al., 2002, Pressman, 2001), it is fair to say that Web systems development is now reasonably established as a sub-discipline. However, there are also opinions that most of the differences get exaggerated in Web systems that may not be very critical for traditional software systems (Turban et al., 2000, Pastor et al., 2002, Yusop et al., 2005).

A lot of effort has been put towards Web systems development processes, practices and tools; however, evolution, growth and maintainability of Web systems are still considered as challenging areas during Web systems development. Researchers, including Norton (2001), Pressman (2005) and Godfrey & German (20008), have considered the evolution of Web systems as an important area for further research, primarily due to the lack of adequate support provided by development processes (Murugesan, 2008, Paiano et al., 2009a). In his latest research work, Murgesan (2008) has presented a list of challenges including system evolution and maintenance as one of the challenges needed to develop sustainable Web systems. He further argued that different aspects of system development should be worthwhile to consider during their development. One of the important aspects during Web systems development is the way in which business processes changes can affect the architecture design of Web systems.

2.5.1 Web systems modelling languages

Web systems development is not a one-off event, instead it is a process with an iterative life cycle to manage continuous evolution (Murugesan, 2008). Web systems must be developed with a mindset towards rapid changes throughout the system lifecycle. Common Web system modelling approaches includes modular architecture design, use of re-useable components and Web application frameworks. Additionally a number of Web design modelling languages were proposed. An overview of design methodologies for Web systems can be found in (Christodoulou et al., 1998, Lowe, 2000, Scharl, 2000).

One key activity during Web systems development that has received increasing attention in the last decade is modelling languages and approaches catering aspects such as business modelling, functional modelling and information modelling. Most of these modelling languages provide sophisticated notations to facilitate the understanding, representation and construction of architecture design during the course of Web systems development. The earliest modelling languages and approaches such as The Object Oriented Hypermedia Design Model - OOHDM (Schwabe et al., 1996, Schwabe & Rossi 1998) and Relationship Management Methodology - RMM (Isakowitz et al., 1995) mainly focused on hypertext design. Most of the Web systems modelling languages and approaches were further developed with the aim of successfully modelling information and content, and focused on to store, retrieve, transform and present information to the users. However, the aim of modelling information and content has been further extended by posing additional modelling requirements to support business processes (Distante et al., 2007a). In order to address business processes concerns in relation to Web systems design, Web modelling approaches have been developed as a possible solution to model the integration of business processes with architecture design. Example of these Web modelling languages are WebML (Ceri et al., 2000b), UWA+ (Bochicchio and Longo, 2004), UWAT+ (Distante et al., 2006), WFdHT (Brambilla et al., 2006) and WIED (Tongrungrojana and Lowe, 2004b) etc. These modelling languages provide adequate support specifically to model the connection between business processes and the architecture design of Web systems.

2.5.2 Architecture design of Web systems

Architecture design is considered an important phase of system development that typically supports the transition from system analysis to implementation. Mostly, architecture design is represented by system abstractions or models in order to simplify and communicate complex system structure (Eichinger, 2006). According to Bass et al. (1998), architecture design of a system consists of its structure, the decomposition into components, their interface and relationships. Bass et al. (2003), has further defined architecture design as, '*the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them'*. DeMarco (1995) refers to architecture design as "framework of change" where architecture design forms the framework in which a software system can evolve. Starke (2002) summarised a few important properties of architecture design including, '*architecture describes structure*, *architecture form the transition from analysis to implementation*, *architecture can be looked from different viewpoints*, *architecture makes a system understandable*, *and architecture represent the framework for a flexible system'*. Further, Taylor and Hoek (2007) describe architecture design as, *"the structure of system/software and its attributes, such as considering what components and connectors comprise a system, and what constraints govern their interactions"*. Additionally, architecture design has been described in IEEE Standard 1471-2000 as, '*the fundamental organisation of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution'*. Given the number of architecture design descriptions, it can be concluded that the development of architecture design mainly deals with structuring various system components and deciding the linkages among those components.

Given the general definition of architecture, Kruchten (1995) has argued that Web systems architecture can build upon four views including the logical, process, physical, and development view. Grady Booch (2001) also has supported the Philippe Kruchten

(1995) argument that complex systems (like Web systems) cannot be understood from just a single viewpoint. He further proposed the adoption of 4+1 views (logical, process, physical, development and uses case views) of architectural design for Web systems by focusing on the interest of each stakeholder involved in the design process (Booch, 2001). Further, Web application architecture framework (WAAF) has been proposed to develop Web systems based on seven different perspectives. These perspectives include Business Architecture, User Interface Architecture, Information Architecture, System Architecture etc. (Kong et al., 2005a). Different perspectives and views as described by the proposed approaches reveal that Web systems architecture design tends to be complex, and needs to be modelled from different perspectives to better understand the potential changes and the impacts of those changes.

As discussed in the previous section, a number of Web system modelling approaches and languages have been developed and reported in the research literature. Most of those modelling approaches, for example OOHDM, WebML, WIED are used both for architectural design and detailed design specification. Another example is the Unified Modelling Language (UML). UML was specifically developed to manifest detailed design specification. However, UML has been classified by software engineering institute as an architecture modelling language (CMMI, 2006). A number of literatures have been reported on modelling Web systems architecture using UML (Conallen, 2002) and there is an overlap between the architecture design and detailed design of Web systems.

Web systems architecture are more likely to change than that of traditional software systems as Web systems architecture covers both information architecture and technical architecture (Lowe and Henderson-Sellers, 2003). However in Web systems, information architecture (which covers aspects such as information structure, information behaviour and its viewpoint and navigational structure) is substantially more sophisticated than traditional software systems (Lowe and Henderson-Sellers, 2003). In the next section we will explore information architecture and its importance in the Web systems context, particularly while considering business processes changes.

2.5.3 Web systems and information architecture

One of the important features of Web systems is that they use information intensively. Almost every industry from finance, manufacturing, media, government, education, telecommunication and utilities, has an increasing reliance on quality information for both survival and success. With the increased dependence on information of web-based businesses, many Web systems are being used as a public interface for organisations. As a result, the importance of information architecture has become quite intrinsic (Evernden and Evernden, 2003).

Information architecture provides the glue to allow information and high level functionality to effectively work in tandem to mediate among business processes/goals, users' requirements, constraints, and design solutions (Morville and Rosenfeld, 2006). Irrespective of the sophistication of functionality and the creativity of the interface, a Web-based business may likely fail without adequate alignment of information structures and information behaviour with the changing business processes and needs. Failure in attempt to address impacts on information architecture resulting from business processes changes may lead to problems that are more visible externally (and as such may affect the users' interaction and relationships with external stakeholders).

Given the evolutionary nature of Web systems that requires changes to functionality, the frequency and degree of information changes can also be high. User interests in systems get changed as end-user diversity increases (user goals and task become more diverse). Similarly information spaces (database, XML files, file folders, or other repository of external information) are likely to change as a result of evolving business processes. With so many moving targets, it is impossible to develop perfect information architecture, instead the focus should be to maintain an overall consistency of information structure and information behaviours along with business process changes. A possible way to maintain consistent information architecture is to adequately address the impacts on architecture design resulting from business changes. Additionally, the most obvious need to focus on the identification of impacts on information architecture is the strong connection between business process and the information architecture (Evernden and Evernden, 2003, Fielding and Taylor, 2000, Tongrungrojana and Lowe, 2004b) as discussed in Section 2.3.

2.5.4 Early identification of change impacts

In this section we first address the research question ***RQ5a: What does it means by early identification of change impacts?*** In Web systems, it is critical that business processes and architecture design be aligned so that business processes can be adequately supported by the developed system (Aversano et al., 2005). For any potential business processes changes, it is important to make necessary modifications at architecture design to ensure their possible alignment. It has been argued that sufficient tracing of impacts on architecture design resulting from business process changes is not particularly well established (Cesare and Serrano, 2006). An inadequate judgment of impacts on architecture design may result in inaccurate modifications that are required in response to business processes changes.

An adequate consideration of the connection between business processes and architecture design will allow Web developers to examine how changes in business processes can affect architecture design. The identification of impacts on architecture design resulting from business processes changes is worth considering, and in this thesis, we refer to the identification of impacts on architecture design resulting from business processes changes as 'early identification of change impacts in Web systems'. In the next paragraphs we will address the research question ***RQ5b: Why is early identification of change impacts important?***

Firstly, identifying the scope of impacts on architecture design resulting from business processes changes is essential during system development. Misidentification of this scope may result in inadequate modifications at architecture design or redundant re-design efforts, while unnecessarily extending the scope of possible changes at architecture design. Failure to adequately identify change impacts on architecture design leads Web developers to identify the impacts of business processes changes instead at subsequent stages of Web systems development such as detailed design and implementation stages. Consequently, at these stages it would be very late to refer back to the architecture design, particularly to address the impacts that are resulted from business processes changes. Instead, if the impacts resulting from business processes changes are analysed at the architecture design stages then it will allow Web developers to take appropriate and actions earlier (Rombach, 1990).

Secondly, a lack of understanding of the ways in which changes in business processes affect architecture design can potentially play a major role toward Web systems failures. This lack of understanding mainly leads to the problem in relation to change impact analysis – where the detailed design or implementation actually begins before change impacts are adequately identified. As a consequence, project complexity and cost increase, and specifically time can be wasted in unnecessary re-work during subsequent stages of Web systems development. A possible attempt to address this problem is to adequately identify the impacts of business process changes on architecture design before it becomes too costly (due to unnecessary re-work) during Web systems development.

To sum up, we have reviewed the characteristics of Web systems that ought to be investigated in relation to change impact analysis in Web systems, and the extent to which these characteristics guide us toward the dependency of architecture design on business processes. These characteristics include (i) Web systems are intricately interwoven with business processes and (ii) there is a strong connection between business processes and architecture design of Web systems. Additionally, we have investigated the nature of changes and their impacts in Web systems context. The ability to successfully address the impacts of business processes changes on architecture design depends on understanding both the nature of changes and their impacts, and the nature of dependency between business processes and architecture design. Consideration of these aspects guides us toward the identification of impacts on architecture design resulting from business processes changes. In this thesis we have referred to the identification of impacts on architecture resulting from business process changes as "early identification of change impacts in Web systems".

Different characteristics of Web systems and the resulting nature of dependency between business processes and architecture design (as discussed in Section 2.3), and the different nature of changes and their impacts in Web systems (as discussed in Section 2.4) reveals that early identification of impacts in Web systems is essentially important and needs to be addressed in Web systems. This will lead our focus to investigate the existing approaches of change impact analysis and their possible adoption in the Web systems context.

2.6 Change Impact Analysis

In this section we will begin our investigation by describing the 'scope of change impact analysis', 'change impact analysis approaches and tools' and 'definition of change impact analysis' while focusing on the current state of research in the area of change impact analysis.

2.6.1 Change management

Changes are inevitable and likely to be raised while a system is being developed. Importantly, functionality supported by the system may change at any point during the life of system development. These changes need to be identified, controlled and managed. Out-of-control change is recognised as a sign of project failure or unnecessary project delays (Hull et al., 2002). Many organisations prefer to establish a change management process to plan, monitor and control the requested changes. With a change management process in place, proposed changes that may come from different sources can be effectively assessed for their possible impacts on different project artefacts. The change management process typically includes four steps: (1) change initiation and classification, (2) change impact analysis, (3) change acceptance and rejection, and (4) change implementation and verification (Jonsson, 2007). It means that change impact analysis is part of the change management process. Decisions to reject or allow the changes and a plan to implement those changes can be generated from the results of the change impact analysis.

2.6.2 Overview of change impact analysis

Change impact analysis allows for a detailed examination of the consequences of a change being made to the software development artefacts, and thus change impact analysis provides extensive support in understanding changes and their impacts (Bohner and Arnold, 1996). Here software development artefacts refer to business process specification, system requirements, architecture design, detailed design, code, or test cases. To better understand the concept of change impact analysis, the definition and scope of change impact analysis are first discussed.

Turver and Munro (1994) define change impact analysis as, '*the assessment of change, to the source code of a module or on the other modules of the system. It determines the* *scope of a change and provides a measure of its complexity*'. This definition is limited only to the changes made and their impacts identification on source code. It tends to overlook other system artefacts such as architecture design, detailed design and system support documentations. Bohner and Arnold (1996) define impact analysis as, '*identifying the potential consequences of a change, or estimating what needs to be modified to accomplish a change'*. This definition is general and states that the scope of a change is not limited to any phase or activity. Pfleeger (1998) defines impacts analysis as, '*the evaluation of the many risks associated with the change, including estimates of effects on resources, effort and schedule'*. This definition focuses explicitly on risk and project control, and emphasises the non-technical aspects of impacts such as resources, effort and schedule. In this thesis we will follow the definition of change impact analysis provided by Bohner and Arnold (1996), both due to its generic focus and extensively cited definition in the research literature. In the following section we further review the terms: 'change' and 'potential consequence' that are adopted in the definition given by Bohner and Arnold (1996).

According to Bohner and Arnold (1996), "a change can refer to the addition, deletion and modification of system life objects (SLOs)". The term SLOs is specifically used by Bohner and Arnold and it refers to system life cycle objects which are affected by a change. SLOs may include abstract level objects such as requirements specification, architecture design, detailed design specifications, code, test cases or SLOs may include detailed level objects such as design entity, program function or variable and a step in text case execution. Lindvall (1997) describes change impact analysis specifically for software requirements context as, *"Requirements-driven impact analysis identifies the set of software entities that need to be changed to implement a new requirement in an existing system"*. It implies that Lindvall's definition of impact analysis covers the same scope but is limited only to the introduction of the new requirement as a system object.

A ***potential consequence*** is the effect of a change on the existing system (Bohner and Arnold, 1996). Potential consequences of making a change can be divided into primary and secondary impacts. Per Jonsson (2005) has referred to primary impacts as direct impacts that are identified by analysing the effects of a proposed change in the system. Whereas, secondary impacts are referred to as indirect impacts that are caused due to primary impacts. For example, a change from artefact "A" to an artefact "B" can be determined by considering direct link from artefact “A” to “B” and this impact is called primary impacts. Whereas, a change from artefact “A” to an artefact “C” can only be determined by considering the impact on “B” and its further consequence on “C”. The consequence on “C” is an indirect impact of the original change made to “A” and called secondary impacts. Bohner and Arnold refer to two forms of secondary impacts: (i) side effect and (ii) ripple effect.

A ***side effect*** is an “error or other undesirable behaviour that occurs as a result of a modification” (Bohner and Arnold, 1996). Side effects may cause damage to system and must be avoided.

A ***ripple effect*** is the “effect caused by making a change to a system which affects many other parts of a system” (Bohner and Arnold, 1996). The ripple effect is the consequence of the way system is structured, designed and implemented; therefore, the ripple effect cannot be avoided. However they must be identified prior to any modification made in the system (Jonsson, 2007).

Referring to the definition of change impact analysis by Bohner and Arnold, it is evident that “potential consequences of a change” or “what needs to be modified to accomplish a change” refer both to the primary impacts and secondary impacts. Therefore, it is important to note that change impact analysis covers both the identification of primary impacts (direct impacts) and secondary impacts (indirect impacts).

2.6.3 Basic change impact analysis mechanism

As illustrated in Figure 1.1, Bohner and Arnold (1996) describe three activities of CIA.

1) Examine software and change specification and identify starting impact set

The system developer starts this activity while examining the change specification or the change request form. This examination leads to a clear understanding of the change specification and its mapping at the decomposition of system objects. In doing so, the system developer scopes out the change impact by determining which system objects are initially affected by the change (Bohner and Arnold, 1996). The analysis of system objects supports Web developers by identifying the apparent impacts of a change in the system. This activity ends after the identification of the initial set of impacts that are apparent or currently known while employing expert judgement, previous experience and gut feeling. According to Bohner and Arnold (1996), the initial set of impacts comprises of system objects that are thought to be affected by a change and are referred to as the starting impact set (SIS).

2) Analyse and trace potential impacts- candidate impact set

With the initial set of impacts, further system objects' relationships to other system objects are studied. Mostly these relationships are implicit; however, change impact analysis assumes that there are relationships available that characterise the structure of the system or system objects. With these relationships, system developers can account for any potential ripple effects of a change. This activity extends the starting impact set (SIS) as identified in step#1. System developers identify candidate impact sets while focusing on the ripple effects of SIS during this activity. Bohner and Arnold (1996) have described the candidate impact set (CIS) as the set of system objects estimated to be affected.

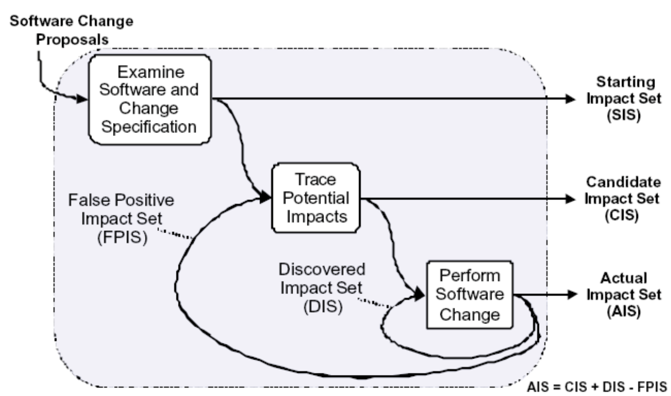


Figure 1.1 : Software change impact analysis – Adopted from (Bohner and Arnold, 1996)

The CIS can be determined by employing different impact analysis approaches (for example traceability analysis, dependency analysis and others (See Section 2.6.4).

3) Perform software changes- actual changes

The system objects identified (as CIS) in step#2 are modified by specifically performing changes in the system. The execution of this step validates the CIS and may further discover other impacted system objects in the system. This step is iterative in nature and used for discovering new impacted objects (i.e. discovered impact set) and falsifying the object in the candidate impact set (CIS). Therefore, when a change is being performed in the system, there is a likelihood that other impacts will be identified which have been overlooked during the initial estimation and only identified at the time of performing change. These newly identified impacts are in addition to the impacts identified in CIS. These discovered impacts can be categorised as Discovered Impact Set (DIS) and represents an under estimation of CIS. Similarly, there are few false impacts identified during the initial estimation, those are not true while the change is being performed. There is no role of these impacts during the change being made; therefore, these are considered as false positive impacts and represent an over-estimation of CIS. These impacts are categorised as False Positive Impact Set (FPIS).

Finally, the actual impact set (AIS) is the set of system objects that are actually modified and represent an accurate estimation of impacts. According to Bohner and Arnold (1996), the CIS in addition to DIS and minus deletion of FBIS represents the AIS.

Keeping in view the basic mechanism, it is understood that change impact analysis actually occurs throughout the system life cycle. As more changes are made, more impacts are determined and the set of known impacts may grow. Intuitively, CIA allows system developers to determine the system level artefacts that may be affected by the changes, and thus facilitate tracing the effect of that change during system development. CIA helps in gauging the cost of a specific change during cost estimation activity (Huang and Song, 2006). During project planning CIA helps in choosing among several solutions for implementing a given change - the one with lowest estimates of potential costs (Sherriff et al., 2007). While during testing, it alerts developers toward potentially affected program components requiring retesting; thus, reducing the risk associated with the changed software (Mark et al., 2007). Consequently, CIA can be used to guide regression test efforts by reducing the set of test cases to traverse a change and its impacts (Ryder and Tip, 2001). However, most research on impact analysis is focused on software code changes and system maintenance, although CIA undoubtedly plays an important role in the entire system life cycle. For example, Lindvall (1997) suggests a definition for requirements-driven impact analysis to analyse the effect of new requirements on an existing system. Similarly, the role of CIA is also important in other phases of system development including architecture/design (Zhao et al., 2002); (Tang et al., 2007, Berg, 2006) and testing phases (Mark et al., 2007, Orso et al., 2003).

2.6.4 Change impact analysis at architecture level

As reported by Bohner and Arnold (1996) change impact analysis supports a detailed examination of the consequences of a change being made, and thus it provides considerable leverage in understanding change impact on system level artefacts. Given that CIA extends support to identify impacts at system level artefacts, practitioners can use it to assess change impacts at different abstraction level such as at the level of the subsystem, architecture design, detailed design and code module (Mittermeir, 2001).

During architecture design analysis, it is important to understand the evolution of architecture components, the relationships between them and the way components communicates with each other (Mehta and Heineman, 1999). Bose (1998) has described the change to architecture design as the addition/deletion of components and the addition/deletion of connections between the components. Additionally, a number of researchers focus on describing changes to architecture component and develop change taxonomies that include adding, deleting, modifying, or substituting components, connectors, ports, and services (Aoyama, 2002, Nedstam et al., 2004, Robbins et al., 1996, Sadou et al., 2005).

While analysing the impacts of a change at architecture design, the influence of change can be classified as weak, average, or strong (O'Neal, 2001). However, Lassing et al. (Lassing et al., 1999a, Lassing et al., 1999b) have described and employed a subjective impact scale during architecture level change assessment. There are four levels of impact scale including no impact, impact on one component, impact on several components and impact on whole architecture design.

Given the connections or relationships between components that bind them together, Williams and Craver (2010) have argued that these architecture components can have causal dependencies and ordering dependencies. An example of causal dependency includes where one component causes behaviour of another component of architecture design. An example of ordering dependency includes where a specific ordering relation is being imposed between two or more component’s behaviours. Indeed, often these dependencies exist among architecture components but mostly these dependencies are implicit. Change impact analysis plays an important role in making these dependencies explicit in order to support the identification of components that need to be modified.

Change impact analysis is considered an important activity during change management process. Change impact analysis allows system developers to determine the artefacts that may be affected by the change and thus facilitate tracing the effect of that change to different levels of system abstraction. Based on the result of change impact analysis activity, CIA helps reduce the cost and the risk caused by unanticipated impacts, and thus facilitates decision making. Most of the CIA approaches have been developed to focus on the program code level. There are surprisingly few methods and approaches developed to assess the changes and their impacts at the architectural level. We will now describe these CIA approaches in the next section.

2.7 Change Impact Analysis Approaches

In this section we discuss existing change impact analysis techniques, their comparison and their generic properties. CIA techniques can be divided into two broad categories namely dependency analysis and traceability analysis. Jonsson and Lindvall (2005) states: "in *dependency analysis detailed relationships among entities, for example variables or functions, are extracted from source code"*. Whereas, *"traceability analysis is the analysis of relationships that have been identified during development among all types of artefacts"*.

Dependency analysis is suitable for analysing interdependencies of data, functionality, and component subject to the availability of explicit or implicit dependencies information, whereas traceability analysis is suitable for analysing relationships among requirements specification, architectural design, detailed design, code and documentation. Bohner and Arnold (1996) describe dependency analysis as, "*it involves examining detailed dependency information relationship among impacted system objects. It provides a detailed evaluation of low-level dependencies among objects but does little for SLOs at other level*". The definition of dependency analysis implies that it provides a narrow perspective of analysis and covers the depth of change impact analysis. Similarly, traceability analysis is described as, "*Traceability analysis involves* *examining relationships among all types of SLOs and it may cover many of the relationships among artefacts that a software project repository might store, however these relationships typically are not very detailed*" (Bohner and Arnold, 1996). Traceability analysis provides a broader perspective and covers breath of change impact analysis.

The research approaches in relation to modifiability, maintainability and quality assessment of architecture design has received increasing attention in the last few decades. Most of these approaches depend on scenario-based concepts and uses both traceability and dependency analysis. Very known approaches of software architecture assessment are the software architecture analysis methods-SAAM (Kazman et al., 1994, Kazman et al., 1996) and the Architecture level Modifiability Analysis-ALMA (Lassing et al., 2002, Bengtsson et al., 2004). However, these quality assessment approaches are limited to non-functional requirements such as usability (Rafla et al., 2006), modifiability (Lassing et al., 2002) and portability (McCrickard and Abowd, 1996). While investigating related research literature our focus is on change impact analysis approaches that are developed and proposed for architecture level analysis where we are interested in changes related to the functionality of the system.

2.7.1 Traceability analysis

Traceability analysis supports the analysis of impacts on system artefacts and across different levels of abstractions. Few approaches focus on traceability tables to identify and manage linkages that have requirements among them, and linkages from requirements and design elements (Kotonya and Sommerville, 1998). Given the tracing of potential effects, Ibrahim et al. (2006) have classified the traceability analysis as top down and bottom up traceability. Furthermore, some approaches use a traceability matrix to represent the relationships, and furthermore, the impacted objects are inferred by computing the transitive closure of that matrix (Jonsson and Lindvall, 2005). The traceability analysis employs tracing from one object to another and support only for breath of analysis; therefore, the traceability matrix and its instrumentation (e.g. reachability matrix) are of limited use to access secondary/indirect impacts (Jonsson and Lindvall, 2005).

The usefulness of traceability information is mainly subject to the consistency, correctness and completeness of the identified relationships and linkages information. Ramesh and Jarke (2001) have reported the failure of existing CIA practices to fully embrace the utilisation of semantic information from the documented traceability relationships. In the situation where traceability information is available down to the lower level of granularity (e.g. file level or function level in source code) it can be very efficient to use these relationships to determine change impact from requirement to code level. A possible drawback of this approach is that very large systems can have very complex and massive amounts of traceability information to be used and analysed while tracing the change-propagation (Jonsson and Lindvall, 2005). However, up to an extent the use of requirement management tools can facilitate traceability analysis subject to the availability of completely and consistently maintained relationships information.

Most of the techniques based on traceability analysis support impacts identification at a broader level of abstraction by focusing on the impacts within a system artefacts (e.g. among requirements) or between system artefacts (e.g. from requirement to architecture design and from code to test cases and system manuals). Indeed, traceability analysis approaches provide a broader perspective (horizontal analysis) and cover breath of change impact analysis. However, a broader perspective of traceability analysis makes it difficult to perform deeper analysis (vertical analysis) within system level artefacts.

2.7.2 Dependency analysis

Dependency analysis is the mostly used technique to analyse change impacts. Intuitively, this technique follows a vertical analysis approach. Broadly speaking, dependency analysis supports the identification of impacts within software artefacts (e.g. within source code or within architecture design) and attempts to assess the effects of a change by analysing the semantic dependencies among the objects/entities of a software level artefact. Various code based analysis models that employ the dependencies between program entities (such as files and functions) have been used for the purpose of CIA (Kagdi and Maletic, 2006). Besides code-level CIA approaches, a few architecture level CIA approaches have also been proposed.

Architecture level CIA approaches have mostly focused on employing static dependencies (focuses on all possible system inputs and behaviours) among architecture design entities. Few efforts have also been made toward the dynamic analysis of change impacts (focuses on static dependencies that have been involved in system executions), including the work proposed by Feng and Maletic (2006). Based on the dynamic analysis based approach, Stafford et al. (2003) introduced the chaining technique to analyse architecture level inter-dependencies during architecture design modification. This chaining technique used a table based approach to identify the chain set. A similar approach was proposed by Feng and Maletic (2006) to study the component composition (of architecture design) in order to determine the impacts of changes at architecture design. However, the approach proposed by Feng and Maletic is based on employing set of inputs, interaction sequences among components, and interactions triggered by users. Typically this approach is characterised by intra-component and inter-component impact rules. In general, architecture level CIA can be classified into three of the most representative architecture CIA approaches that largely encompass other available methods and techniques. These three architectural level CIA approaches include crosscutting in architectural design (Berg, 2006), architectural slicing (Zhao et al., 2002), and design rational analysis (Bratthall et al., 2000, Tang et al., 2007) respectively.

The CIA approach based on the crosscutting technique was proposed by Berg (2006) supported a mapping between two domains or two levels of system abstraction. These domains and levels are treated as source and target during the execution of change impact analysis. The crosscutting concept described by Berg et al. (2005) mainly covers inter-level impact analysis, i.e. at the same level of abstraction where different elements crosscut each other. Thus, the proposed approach is limited to architecture design level and does not support to identify the impact on architecture design resulting from business processes changes.

Another approach based on design rational has been proposed to predict change impacts in architecture design (Tang et al., 2005, Tang et al., 2007). The proposed approach employs probability calculus to quantify the dependencies between the architecture rationale and architecture elements (specifically called architecture rationale and element linkage- AREL). Based on a causal relationship between the architecture rationale and architecture elements, AREL provides a quantitative approach to support architecture level change impact analysis. However, this approach can be applied in a situation where a change is made to architecture and the impact of that change needs to be identified at the architecture design level.

Architecture slicing is another area of interest for change impact analysis. Architecture slicing has received more attention from the research community in comparison with crosscutting and rational-based approaches. Zhao introduced static architecture slicing (1998) and further proposed architecture dependence analysis techniques to support software architecture development by employing a specific Architecture Description Language (ADL) (Zhao, 2001). An extension of Zhao's earlier work (Zhao, 1998, Zhao, 2001) is reported in the following (Zhao et al., 2002). The work reported by Zhao (1998) failed to address the architecture complexity in situations where a large number of connections between architecture components are present. Kim et al. (2000) proposed another architectural slicing technique to generate a smaller number of components and connectors in each slice as compared to Zhao (1998) and called the technique dynamic software architecture slicing (DSAS) in the following (Kim et al., 2000). Further, an extension of Zaho's earlier work is based on architecture slicing and chopping technique (Zhao et al., 2002) and employ Wright (Allen, 1997) as another ADL. This approach supports identifying impact of changes in software architecture by analysing the semantics of Wright- an architecture design language.

Korel et al. (2003) present two types of slicing - deterministic and nondeterministic slicing by employing the EFSM (Extended Finite State Machines) approach and further report a tool to support the proposed slicing technique. Kagdi et al. (2005) introduced the concept of design model slicing as a means to ease maintenance through the understanding, querying, and analysing of large UML models. Primarily, their slicing approach extracts slices from UML class models and thus is limited to supporting only structural behaviour.

Mainly, CIA approaches based on slicing of architectural design have been investigated by many researchers including Stafford et al. (1998, 2003), Zhao (1998, 2001), Zhao and Xiang (2005), Kim et al. (2000) and Korel et al. (2003). Most of these architecture CIA approaches use architecture descriptions developed with some ADL. The research work on slicing UML architectural models by (Kagdi et al., 2005) and (Lallchandani and Mall, 2008) have also been reported in the literature. Consideration of a specific

ADL while performing change impact analysis at the architecture level has limited adoptability of these approaches to architecture design that are developed with an ADL or modelling languages like UML. This limitation toward architecture level CIA is true as most of the approaches employ the semantic, syntax and relationships vocabulary from that specific ADL to perform the dependency analysis. For example, approaches that are developed specifically for UML model employ different types of relationships such as aggregation, association, composition and generalisation/specialisation.

2.7.3 Change impact analysis process

Most of the change impact analysis research has been done on technical aspects such as the development of methods and tools for assessing the impacts of changes on different system artefacts. As reported by Pen Jonsson (2007), one drawback of focusing on purely technical aspects is that such research often assumes the presence of a detailed and supportive infrastructure (such as supporting tools, documents, activities) to apply change impact analysis approaches. It is thus important to investigate specifically what supportive infrastructure there is and how architecture level CIA is currently performed.

Related research, for example, the work of Zhao et al. (2002) focuses on a process-based approach of CIA and proposes three phases including develop the architectural flow graph, find slices and chops over the graph, and map the slices and chops to architectural specification. But this work does not cover the steps and activities to perform CIA in detail and other infrastructure support to carry out their proposed CIA approach. Other related work proposed by Berg (2006) focuses on the matrix representation of crosscutting dependencies and shows that crosscutting may have a large change impact. Similarly, Tang et al. (2007) represented the probabilistic causal relationships between design elements and decisions. However both of these approaches do not cover process aspects such as details of steps to follow, flow from one step to another for CIA, inputs/outputs, and activity descriptions of the process. The paucity of process based research work in relation to architecture level CIA motivates us to focus on a process aspect of the CIA approach.

The work reported by Per Jonsson (2007) based on commercial practices suggests that the analysis of change impacts are quite subjective and the top three basis includes gut feeling, experience and competence (as opposed to models or other documentation).

Any analysis made based solely on a subjective basis is bound to be person dependent. In the worst case scenario, subjective basis analysis may be inconsistent with what others would produce. Process support such as, steps or activity descriptions could result in a more objective and well-founded analysis for change impacts, and also reduce the risk that something is missed during the analysis. These aspects also direct us towards a need for a process support during change impact analysis to follow a consistent, rigorous and step-by-step CIA approach.

2.7.4 Change impact analysis tools

For architecture level change impact analysis, there are various tools that are used by organisations to analyse and identify the impacts on architecture resulting generally from changes made to system requirements. There are limited case studies and lessons reported in scholarly publications about organisations that have used change impact analysis tool in their web projects. Most of the articles reviewed are obtained from websites of companies (such as IBM, Mercury, IEAD, etc.) that developed tools to support architecture impact analysis. However, the lack of scholarly articles does not imply that it is not important to track and analyse issues that relate to Web systems development. It may simply imply that it is a field yet to be adequately explored by researchers.

Enterprise Architect™ (EA) is developed by institutes for enterprise architecture developments (IEAD). This tool provides traceability from requirements analysis to design, and further the implementation and deployment phases by using the Enterprise Architect's Relationship Matrix and Hierarchy View. Another proposed tool was the Impact Professional that illustrates the way in which the structural impact analysis for architecture design can be done. However, this tool is limited in its scope to architecture level change impact analysis only. Additionally, this tool only supports the architecture design modelled using ArchiMate language (Zhao et al., 2002) and thus limited the adoption of Enterprise Architect™ for a specific architecture design specification.

Other commercial tools developed, such as Clarity™ by IBM, Mercury Change Impact Testing™ by Mercury, Lattix Architecture Management System™ by Emenda, support the analyse change impact analysis among different system artefacts including requirements, design, source code and test cases by employing the traceability analysis approach. Most of these traceability-based tools are suitable for Web development projects, primarily tracking and identifying the effects of a change at the system artefacts level (Hadar and Hadar, 2006) but not deep enough at the architecture level. Additionally, no empirical validation could be found in scholarly publications on how these tools are used and how useful change impact analysis results were achieved.

2.7.5 Existing Web systems CIA techniques

Most of the techniques that have been reported in the research literature for Web systems change analysis are focused on program code analysis and further orient towards testing activities. A relatively early work by Ricca & Tonella (2000) proposed performing statistical analyses on structural and historical dependencies so as to be used for regression testing. Similarly, another approach for dynamic extraction of model information has been proposed to be used during statistical analysis and testing (Ricca, 2004). This approach is based on the input values to cover all internal states of applications. However, the proposed method is limited to input values that can be used to cover all the relevant behaviour of the Web Systems (Tonella and Ricca, 2002). Dynamic slice (at code level) is another technique that focuses on the execution of system behaviour but is limited to a specific operational profile and for a given input as described in the following (Ricca and Tonella, 2001).

Similar to traditional software, slicing analysis techniques for component based systems (such as Web services based systems) was introduced by Pan et al (2005). With this approach analysis and change-prediction complexity are reduced significantly as complexity is encapsulated within architecture layer but this approach is limited only at the logic layer of the three tier Web systems architecture. Additionally, other proposed slicing approaches may also be applicable to an individual tier. For example, techniques to analyse impact for GUI were proposed by Galliers et al. (1999) and for maintaining functionality at the logic tier by Xiaoyu (2006). Another dependency approach has been proposed by Xiaoyu and Mei-Hwa (2007) to assess the change impacts during testing activity. However, their approach is highly reliant on the dependence models that need to be updated after each iteration, which may hinder its applicability.

A number of approaches have been proposed for maintaining traditional system programs/code (Bohner and Arnold, 1996, Bohner, 2003, Gallagher and Lyle, 1991, Huang and Song, 2006, Law and Rothermel, 2003), object-oriented programs (Ibrahim et al., 2006, Lee et al., 2000, Lindvall, 1997, Ryder and Tip, 2001, Sneed, 2001), component-based software (Bohner, 2003, Jonsson and Lindvall, 2005, Feng and Maletic, 2006), and database-driven applications (Basson, 2001). However, there is less research focus on architecture level CIA approaches. Investigation of scholarly research literature indicates that there is a considerable level of shortcomings in most of the architectural level CIA approaches to support impacts identification on architecture design resulting from business process changes - an important aspect in Web systems. That is the focus of this thesis.

Firstly, most of the architecture level analysis approaches identify changes and their impacts at the architecture design level where changes are being made to architecture and their impacts are further identified at the architecture design level. This means that these approaches are limited to the level of architecture only, and not across multiple system artefacts such as from business process to architecture design. While limiting changes and their impact analysis at the architecture design level, there is a tendency to overlook those impacts on architecture that resulted from business process changes. One exception is the work proposed by Boer et al. (2005), to analyse changes that could affect different domains, for example, business and infrastructure within the context of an enterprise. However, the approach proposed by Boer et al. (2005), employ traceability analysis, a horizontal analysis based approach, and therefore support impacts identification at the abstract level of system artefacts. In general, there is less focus given by most architecture level CIA approaches so as to take into account the dependency between business processes and architecture design. As a result, these approaches do not particularly provide adequate support to analyse the dependency that architecture design has on business processes.

The current research focus of architectural level CIA reveals that most of the approaches developed support software system domains, where designing activity begins as a predecessor of the requirement analysis phase and change identification, and their assessment are typically initiated at the architecture design level. In this thesis, we are motivated to analyse change impact on Web systems architecture that are driven by changes in business process for a number of reasons including:

1. Web systems solution is intricately interwoven with business processes
2. Tight-connection between business processes and architecture design
3. Increased scope and scale of impacts at architecture design resulting from business processes changes
4. Sophistication of architecture design and externally visible problems due to inadequate identification of change impacts

In this we refer to the identification of change impacts on architecture design resulting from business processes changes as "early identification of change impact in Web systems".

Secondly, one class of the CIA approaches focus on the traceability analysis to identify change impact from one system level artefact to another system level artefact. Traceability analysis follows the horizontal analysis approach, covers breadth of CIA and tends to support change impacts identification at a coarse-grain level of system artefacts. Another class of CIA approaches focus on the dependency analysis to identify impacts within a system artefact i.e. from one component of system artefact to other component of system artefacts. Dependency analysis follows the vertical analysis approach, covers depth of CIA and tends to focus at fine-grain level of system artefacts. To address the identification of impacts on architecture resulting from business process changes, it will be beneficial to employ both traceability analysis (from business process to architecture design) and dependency analysis (within architecture design). Most of the existing CIA approaches focused either on traceability analysis or dependency analysis but failed to integrate both traceability analysis and dependency analysis in a single approach to support adequate identification of change impacts. Subsequently, combining both the traceability and dependency analysis will address the problem where the implementation (detailed design) actually begins before change impacts are adequately identified in Web systems.

Thirdly, most of the change impact analysis research has been done on the technical aspect assuming the availability of detailed and supportive infrastructures (such as supporting tools, documents and activities) to apply the change impact analysis approaches. Additionally, most of the CIA approaches do not cover process aspects such as details of steps to follow, flow from one step to another, inputs/outputs, and activity descriptions for CIA approach. Indeed, process support for CIA could result in a more objective and well-founded analysis for change impacts, and also reduce the risk that something may be missed during the analysis.

Fourthly, most of the architecture level CIA approaches use the semantic of a specific ADL to examine the possible dependency among the components of architecture design. Limiting CIA approaches to a specific ADL greatly hinders the adoptability of these approaches for other candidate architecture design specifically developed with different architecture languages. For example, the technique developed by Boer et al. (2005) uses the different relationships semantics proposed by ArchiMate modelling language, thus limiting its applicability to the architectures that are developed with ArchiMate (Zhao et al., 2002).

Additionally, while considering architecture level analysis, it is observed that most of the methods and approaches proposed for change impact analysis at the architecture level are at the inception or refinement phase of their maturity. This may be due to the lack of adequate industrial experiments, empirical works or case studies to validate these methods. Additionally, most of the methods are neither characteristically developed nor adequately validated with Web systems. As discussed in Section 2.3 and 2.4, Web systems are different from software systems, and it is important to consider these differences while adopting existing architecture level CIA approaches in the context of Web systems.

We have investigated a number of CIA approaches and tools that have been developed specifically by employing traceability and dependency analysis. This has identified shortcomings in most of the architecture level CIA approaches that make it difficult to adopt in Web systems. Most of the CIA approaches do not adequately address both the distinct dependency that architecture design has on business processes and the different nature of change impacts in Web systems. Hence, there is a need for a more rigorous process support during change impact analysis.

To support impact analysis at the architecture design level, it is necessary to identify the possible dependency between architecture design components. In the following section we will investigate design decisions information and its support to reveal possible dependency between architecture design components.

Software architecture is mostly developed by taking a number of design decisions to achieve functional and quality requirements. These design decisions are considered as the outcome of a design process during the construction and the evolution of software architecture (Farenhorst et al., 2007). The importance of design decisions for software architecture has received early attention by Perry and Wolf (1992). Furthermore, Kruchten et al. (2005) have reported that architecture knowledge is a collection of design decisions and design specification. Jansen and Bosch (2005) have also reported architecture design as a set of design decisions and emphasised both documenting design decisions (Jansen et al., 2009) and re-using them (Jansen et al., 2008).

Substantial research has been carried out in relation to managing and documenting design decisions and their underlying information. Reasons for documenting design decisions refer to the design recovery process, documenting forward and backward traces between requirements and architecture design (Dueñas and Capilla, 2005), re-using previous decisions during architecture modification (Trujillo et al., 2007) and traceability during the maintenance phase (Capilla and Nava, 2008). Re-using design decisions during architecture modification is important to achieve the quality of architecture (Trujillo et al., 2007) and to better handle architecture level changes. Design decisions can thus be considered as a good source of information to be employed during architecture level modification.

Design decisions are described in terms of underlying information such as design elements, design rationale, design rules, design constraints, trade-off and others (Choi et al., 2006a, Falessi et al., 2008a). Documenting and managing design decisions information in repositories are argued as complementary activities that are carried out in parallel to typical architecture design tasks (Kruchten et al., 2005). One of the objectives of documenting design decisions and codifying them in a suitable form is to re-use these design decisions when needed. Documentation, management and retrieval of design decisions and their underlying information has received considerable attention in the literature.

Some authors have focused on proposing templates and emphasising a set of descriptions while capturing design decision information (Burge and Brown, 2008, Dueñas and Capilla, 2005, MacLean et al., 1991). Other use ontology to structure design decisions and specify attributes that are used to document design decision information (Jansen and Bosch, 2005, Wang et al., 2005). Most of these approaches and tools focus on documenting a set of design decisions information in the form of design rationale, design rules, design constraints, etc. (Jansen and Bosch, 2005)

As a possible support for architecture level changes, design rules and design constraints deserve more attention as they impose certain behaviour on architecture design (Capilla et al., 2006). Design rules and design constraints shape the architecture design both by encouraging (from design rules) and limiting (from design constraints) certain behaviours (Jansen et al., 2008). This encouraging and limiting behaviour guides us toward the underpinning of relationships that exist between the design entities[2] of architecture (Mattsson et al., 2008). It means that, the relationships between designs entities can be identified by employing design rules and design constraints. Further, design rules and design constraints can assist us in determining the possible inter-dependencies between design entities.

Design rules are a prescription for future design decisions and are considered as mandatory guidelines during design activity (Jansen and Bosch, 2005). At a granular level of architecture design, design rules tend to relate one design entity to other design entities; however, these relationships are mostly implicit within the description of design rules (Mattsson et al., 2008).

Design constraints are also the prescription for future design decisions; however, they are described as the opposite side of design rules. Design constraints describe what is not allowed in the future design of architecture. i.e. They prohibit certain behaviours (Jansen and Bosch, 2005). Design constraint captures constraint that a single design entity poses on other design entities and is documented as a part of the description of design decisions that are taken. Similar to design rule, design constraint tends to relate

[2] The terms "design entity" and "architecture component" have been used interchangeably in the research literature. However to be consistent with the research done in relation to design decisions, we will use the term design entity for the rest of the thesis.

one design entity to other design entities at a granular level of architecture design; however, these relationships are mostly implicit within the description of design constraints (Mattsson et al., 2008).

Most of the existing approaches document design decisions are in textual form (Jansen et al., 2007) either by utilising design decisions management tools (Capilla et al., 2006), design patterns (Harrison et al., 2007) or templates (Jansen and Bosch, 2005). Interestingly, the practices to document design rules and design constraints have also been widely used and adopted in the industry (Babar and Gorton, 2007, Capilla et al., 2006, Jansen and Bosch, 2005). Through an extensive investigation of existing design decisions approaches and tools developed for software engineering domain, we have identified that most of the design decisions approaches (Babar et al., 2006, Capilla et al., 2007, Dueñas and Capilla, 2005, Falessi et al., 2008a, Jansen and Bosch, 2005) and tools (Babar and Gorton, 2007, Capilla et al., 2006, Falessi et al., 2008b) provide support specifically to capture and document design rules and design constraints. For example, Tyree and Akerman (2005) define a template to capture design decisions information including design constraints and the same template has been further used by industry practitioners (Capilla et al., 2007, Falessi et al., 2008b, Tyree and Akerman, 2005). Similarly, Janssen et al. (2008) have proposed an approach to document design rules and design constraints during design decisions. Tool based approaches to capture design decisions information, proposed by Baber and Gorton (2007) also acknowledge the importance of recording design constraints. Table 1 presents a summary of design decision approaches and tools, form of design decisions representation, their industry validation, and information captured by these approaches and tools. The summary provided in Table 2.1 guides us toward the availability of design rules and design constraints that can be extracted from design decisions repository to be used for the purpose of CIA.

Table 2.1: A summary of design decision approaches/tools and their documented information constructs.

DD Approaches/Tools	Industrial validation	DD representation	CIA supporting information
ACCA (Wang et al., 2005)	1 experience reported in banking applications	Meta-model of concern	Assumption, alternatives,

	(Wang et al., 2005)	Traceability map (CT-map) (Wang et al., 2005)	arguments captured on CT-map (Wang et al., 2005)
DDV(Lee and Kruchten, 2008a)	1 experiment with industry dataset in laboratory setting (Lee and Kruchten, 2008a)	DD Template (Kruchten, 2004)	Decision relationships, design rules, constraints (Kruchten, 2004)
ADDSS (Francisco Nava, 2007)	4 experiences (Capilla et al., 2008b), URJC & IESE(Capilla et al., 2008a) and (Capilla et al., 2010)	Meta-model	Constraints, assumptions, dependencies, rationale
PAKME (Babar and Gorton, 2007)	2 experiences (Babar and Gorton, 2007), (Babar et al., 2008)	Data-model	Pattern and tactic used, Rationale, constraints and related decision alternative
Archium (Lee and Kruchten, 2008b)	3 experiences Athena (Jansen and Bosch, 2005), GRIFFIN (Griffin), CLC-4-TTS (Chen et al., 2007)	Meta-model	Design rules, constraints, related design decision consequences
AQUA (Choi et al., 2006b)	1 experience (Choi et al., 2006a)	Model of architecture design decisions	Design variable (problem), decision values (solution), risks, and constraints.

In software development projects, information from design decisions repositories are often used to support architecture analysis, re-using design decisions for architecture modification/improvement and to bridge between design decisions and resulting architecture. Similarly, design decisions reveal a set of information such as design rules and design constraints considered important to support change impact analysis. The explicit focus to utilise design rules and design constraints will assist in the identification of possible dependency between design entities and thus supports impact identification at the architecture design level.

2.8 Chapter Summary

In this chapter, we have investigated Web systems, their characteristics, nature of changes and their impact on Web systems. Further, we investigated existing change impact analysis, their strength and weakness, relevance of these CIA approaches for Web systems and use of design decision information to support CIA.

Investigation of Web systems characteristics reveals that there is a special dependency of architecture design on business process that needs to be addressed while performing CIA in Web systems. Further, this investigation guides us toward proposing early identification of change impacts in Web systems, that is, the impacts identification on architecture design resulting from business process changes.

By having a specific focus on the nature of changes and their impacts in Web systems, it has been concluded that the impacts on architecture design resulting from business process changes need to be addressed adequately before it becomes too expensive to address. Failure in attempts to adequately identify the impacts on architecture design resulting from business process change may result in inadequate modification at architecture design and further lead to misalignment of the architecture design with business processes. Therefore, during Web systems development, architecture design can be suitably considered as an appropriate system level artefact to identify the possible impacts resulting from business processes changes.

An important point in achieving successful Web systems development is to adequately align business processes with the architecture design. Thus, it is vital to ensure that the underlying approaches for CIA should support the identification of impacts on architecture resulting from business process changes i.e. early identification of change impacts in Web systems. Investigation of existing CIA approaches reveals that a set of CIA approaches that employ traceability analysis opt to support an abstract level of impacts identification on system level artefacts. One drawback of traceability based approaches alone is that they only support the broader perspective and cover the breadth of change impact analysis. In these approaches, there is no specific focus on the dependency that a design entity can have on other design entities in an architecture design. Other sets of CIA approaches that employ dependency analysis have limited the impacts identification on a single system level artefact such as architecture design and program code. Specifically, from the investigation of architecture level CIA approaches that are based on dependency analysis, it has been observed that only the dependencies between architecture design entities are taken into account for the purpose of CIA. As a result, the connection between architecture design and business processes is primarily overlooked while adopting these CIA approaches for Web systems.

In conclusion, the current research focus toward CIA approaches tend to be less useful for early identification of change impact in Web systems, where impacts on architecture resulting from business process changes needs to be adequately identified before the implementation actually begin. Additionally, there is a lack of a step-by-step, activity based, structured and rigorous approach towards CIA to be adopted for Web systems. This, in turn, requires both an appropriate CIA approach (particularly one that supports early identification of impact during Web systems development) and a process support for carrying out CIA activity.

Previous research highlights that the CIA methods are developed by both an understanding of dependencies between components or sub-systems (Jacyntho et al., 2002) as well as the nature of change impacts and processes (Walter and Scott, 2006) which derives from a set of requirements through design and ultimately to the implementation. This focus may lead us to the supposition that if the nature of dependency, nature of change impacts, and development process are fundamentally changed, then the CIA methods should also be changed accordingly. This is particularly relevant in the context of Web systems, given that the literature indicates that these systems often have different nature of change impacts, different set of development processes, and a specific set of characteristics that reveal different nature of dependency in Web systems. As Web systems become ubiquitous it has become increasingly important to be aware of various underpinning requirements- which are different from traditional software systems and hence the approaches for CIA should address them appropriately.

Given the inadequacy of existing CIA approaches to be adopted in Web systems and the lack of process support, there is a pressing need to develop a process based approach of CIA that facilitates early identification of impacts in Web systems. Additionally, employing design decisions information can support impacts identification on architecture without limiting the CIA activity for any specific ADL implementation or other implementations of the architecture modelling language.

In conclusion, major findings from the review of scholarly research literature highlighted the inadequacy of existing CIA approaches while addressing aspect including:

(i) Linkage between business processes and architecture design of Web systems
(ii) Identification of impacts on architecture design resulting from business processes changes,
(iii) A process level support for CIA, and
(iv) Employment of both traceability and dependency analysis
(v) Addresses the problem where the implementation (detailed design) actually begins before change impacts are adequately identified in Web systems

There are two general opinions on whether or not current software engineering techniques and methods are suitable to be adopted for Web systems. One opinion advocates the need for a new "Web engineering" discipline that handles Web systems peculiarities (Murugesan and Deshpande, 2005, Ramesh et al., 2002, Athula, 2002, Murugesan et al., 2001) and subsequently proposes new approaches, methods and practices. On the other hand, a number of researchers argue that Web systems are a special class of software systems and a natural evolution of information systems (Liu, 2007). Thus, the current traditional approaches, methods and practices from software engineering are still applicable. Our research takes a middle position between these two opinions. Web systems can be perceived as an evolution of information systems but there are also differences when compared with traditional software systems. Holck (2003) is convinced that the shortcomings of traditional approaches should not cause them to be rejected but rather to be extended or supplemented with new ideas and concepts. Therefore, this is not to say that existing software engineering approaches show no value for Web systems, instead, they need to be adapted to address the resulting requirements toward CIA approaches.

The findings from the literature are significant to this study as they provide direction for the next phase of the research, which is selecting research methodologies to validate the findings from the scholarly research literature and from commercial practice. In the next chapter, the research methodology will be presented. Pertinently, from the literature review, we are able to focus on research objectives, research questions, research methods and measurement required for the data analysis and collection activities.

Chapter 3: Research Methodology

3.1 Introduction

This chapter describes the mapping of research objectives to research questions and research strategies to ensure that the research strategies are appropriate for the research questions and objectives. We present the research methodology, which comprises of three stages as shown in Figure 3.1. Each stage describes the investigated research literature topics, research activity, data collection and data analysis methods. In Stage 1, we conducted an extensive literature review of the state of the art and commercial practices as reported in the scholarly research literature. The review of literature was aimed at providing the necessary theoretical foundation for research. In Stage 2, we have gone beyond mere understanding of the state of the practice and have also derived the potential list of high level features that a CIA approach should address. In Stage 3, we have used a case study for the design and validation of the process model of CIA. This chapter begins by describing the nature of our research by taking into account exploratory types of research. Furthermore, in this chapter we describe the research design; map among research objectives, research questions and research strategies employed; describe the research methods selected, and assess the reliability and validity of research methods employed.

3.2 Research Study

Research objectives generally describe the purpose of the study and the possible research opportunity for the phenomenon being studied. The focus of our research objectives is twofold, the first is to develop a better understanding of the current state of the art (existing CIA approaches and their relevance in Web systems) and the state of the practice strengths (what approaches developers mostly employ in Web systems specifically for early identification of change impacts), and especially the limitations of those practices. Second, based on this understanding, we investigate how to extend existing CIA approaches to support early identification of change impacts in Web systems.

Our research study begins by investigating related scholarly/research literature, synthesizing new understandings from linkages between the research literatures, and representing a valid and necessary theoretical foundation for our research in Stage 1, as shown in Figure 3.1. Further, the investigation from the literature review supports and identifies gaps in scholarly/research literature while adopting CIA approaches during Web systems development. From this gap analysis, a number of shortcomings were identified in existing CIA approaches, while adopting those approaches in Web systems. Specifically the literature review highlighted the inadequacy of existing CIA approaches to be adopted in Web systems in order to support impact identification on architecture design resulting from business process changes- early identification of change impacts in Web systems (see Section 2.7), which has laid down a foundation for the identification of the main research questions for this research.

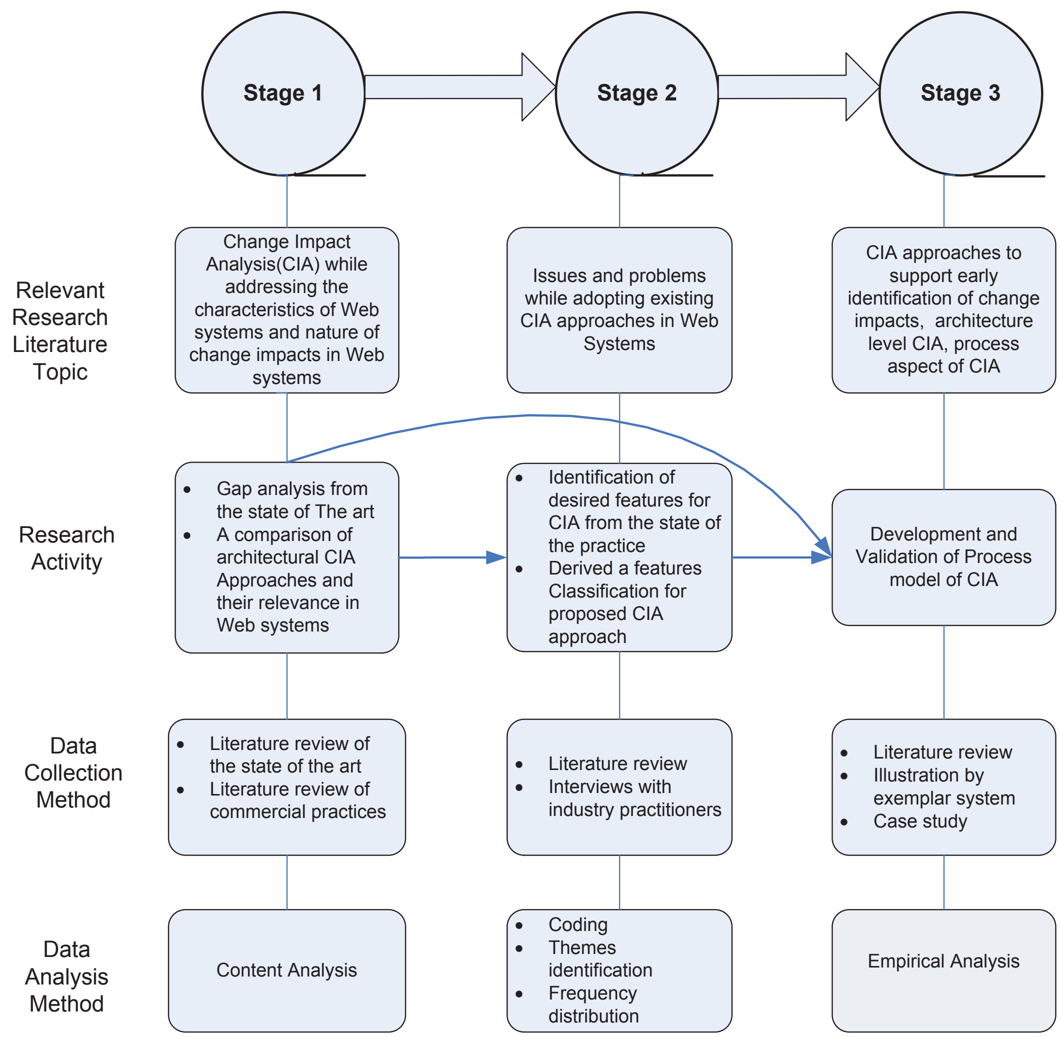


Figure 3.1: Three stages of the research methodology

The research study proceeds further by investigating industrial practices of CIA, seeking industrial perspectives on the gap identified from the state of the art (Stage 1) and to identifying the desired features that a CIA approach should possess. During the investigations both in Stage 1 and Stage 2, we examined the area of CIA while focusing on the characteristics of Web systems and the nature of change impacts in Webs systems.

The investigation results both from the state of the art and state of the practice guided us to develop a process model of CIA in Stage 3 of the research methodology. Further, process model of CIA is illustrated by an exemplar Web system as a proof of demonstration and further validated by a case study. Our research study is completed by the development of a process model of CIA, as a possible solution to address the research questions and further the illustration of the proposed solution.

Most of the previous studies on the topic of change impact analysis (Bohner, 1996, Badri et al., 2005, Canfora and Cerulo, 2006 , Kagdi and Maletic, 2006, Orso et al., 2004, Zimmermann et al., 2005) have adopted a perspective, where generally an attempt was made to test theories and hypotheses, quantify measure of variables, and draw conclusions about the phenomenon. Previous studies have not attempted to understand the relevance of CIA in the context of Web systems. In this research, firstly, we focus on deeply understanding the existing CIA approaches, practices and their relevance in the context of Web systems. Secondly, we focus on the investigation of potential problems/issues faced by Web developers during CIA and the possible solutions to address those problems/issues. To highlight our focus and subsequently address our research question(s), we have employed a range of research methods at different stages of the research methodology.

In the next section, we will describe the purpose of research to better understand the nature of research being conducted.

3.2.1 Purpose of research

Robson (2002) has reported four purposes for research including exploratory, descriptive, explanatory, and emancipatory in order to better understand the nature of research. In the following section, we discuss the exploratory research and its relevance in our research (and the irrelevance of others).

Exploratory Research

Exploratory research is concerned with exploring a particular situation and phenomenon while asking questions, looking for insights, and viewing possible research opportunities from a different perspective. It means that exploratory research is useful when researchers are particularly interested in investigating and gathering baseline information that forms the essential basis for future research, or to study one concept that has not been described in great detail in the literature, or to seek the feasibility of or needs for a more extensive study. Our research study is mainly exploratory in nature, in a way that we have focused on a different-than-usual perspective of change impact analysis, and thus investigated one concept that has not been described in great detail in the literature (i.e. early identification of change impacts in Web systems). Robson (2002) reported that the design of exploratory research is mostly flexible. A flexible design research allows the researcher to investigate all aspects of a phenomenon and to explore all kinds of emerging ideas revealed from data collection and analysis methods (Fitzpatrick and Kazer, 2006).

Empirical Research

Merriam-Webster (2003) defines empirical as:

"(1) originating in or based on observation or experience; (2) relying on experience or observation alone often without due regard for system and theory; (3) capable of being verified or disproved by observation or experiment."

The above definition states observation and experience are two important aspects of empirical research. Indeed, experience is valuable so as to get a better understanding of the strength and weaknesses of the underlying phenomenon. Similarly, observation is important so as to enable researchers to perceive and solve real problems instead of problems that the researcher believes to be important (Colin, 1993). While considering the execution of empirical research, Wohlin et al. (2000) have described three main methods including survey, case study, and experiment. Survey and case study typically employ both qualitative and quantitative approaches, whereas experiment employs only the quantitative approach. To validate the process model of CIA, we have selected a case study (see Section 3.8.8) in an industrial setting. We analysed the case study results empirically; therefore, the research presented in our study can be considered as empirical for Stage 3. The details of the case study are presented in Chapter 6.

3.2.2 Types of research questions

Along with the focus on the nature of research and underlying purpose, a research study should also be well synchronised with (i) the type of research questions asked and (ii) the issue of what constitutes a valid answer for the research question (Easterbrook et al., 2008).

Firstly, we will discuss the type of research questions. According to Easterbrook (2007) there are two classes of research questions including knowledge questions and design questions. Mostly, exploratory study (as ours) focuses on the class of knowledge questions to gain a better understanding of the nature of the underlying research and then focuses on design questions to propose possible solutions. In our research, Stage 1 and 2 is about understanding existing approaches (knowledge question), and Stage 3 is about designing an alternative as a possible solution (design question). Given the class of knowledge questions, Meltzoff (1998) has described different types of research questions including existence questions, description and classification questions, descriptive-comparative questions, frequency and distribution questions, descriptive-process questions, and relationship questions, etc. (Meltzoff, 1998). Our core research question is stated as "Can a CIA technique that specifically addresses the characteristics of Web systems be used for early identification of change impacts? And this tends to partially aligned with the descriptive-process question that deals with the aspects how does X normally work? What is the situation by which X happens? In what sequence do the events of X occur? In our research descriptive process questions support us to explore and develop a deeper understanding of a process (X) and this leads us to propose a design of that process (X). Additionally, in order to address our core research question, in Stage 1 of the research methodology, we have conducted an extensive literature review of CIA approaches and their commercial practices, characteristics of Web systems and nature of change impacts in Web systems. The literature review facilitated us to set the research direction and provided us with the research opportunity for further improvements and future research. In Stage 2, we have investigated industrial practices to gain relevant knowledge and a better understanding of the research problem. In doing so, we investigated how Web developers perform CIA in the context of Web systems? What problems do Web developers face during CIA- especially in relation to early identification of change impacts? What are the features expected to be addressed by CIA approaches?

Another class of research questions is design questions where the goal is to design an approach to carry out some activity in an effective way or in a different setting (Easterbrook et al., 2008). However, design questions presuppose that relevant knowledge questions have already been addressed so that researchers have adequate knowledge about the nature of the research problem. In this research study, after developing a better understanding, from current state of the art and state of the practices, we gained sufficient knowledge about the nature of the research problem and possible solutions. Typically, a long term research program involves a mix of knowledge questions and design questions where researchers investigate the problem including how best to solve it and which solution suits best (Wieringa and Heerkens, 2006).

Secondly, we will discuss the issue of what constitutes a valid answer to the research question. While addressing the issue of valid answers to research questions, Easterbrook (2007) urge that research cannot be conducted without any philosophical stance as that typically guides us toward the judgment criteria for a valid answer of a research question. Additionally, a philosophical stance is important to consider as it may potentially influence the selection of research methods that leads to acceptable research results in response to the research question. Given the importance of the research instance, there are four dominant stances including positivism, constructivism, critical theory, and pragmatism (Creswell, 2009). Positivism refers to employing precise theories in order to extract a hypothesis that can be verified and tested. Constructivism refers to the possible emergence of theories from rich collected qualitative data related to human activities. Critical theory refers to research that actively seeks to challenge existing perceptions about practices. Whereas pragmatism refers to the research knowledge/results, given a potential dependency of these research knowledge/results on the adopted research methods (Menand, 1997). Pragmatism is reported as less dogmatic in comparison with the other three as in a way it provides more freedom to the researcher to choose the research method(s) they believe shed light on the research question (Menand, 1997). Moreover, pragmatism strongly refers to a mixed method approach where various methods can be used to address the problem under study. In our research, we have the freedom to select a range of research methods which we believe sheds light on the research question(s) that we posed for our study. Both the selection of research methods and appropriate evaluation of method results lead to acceptable evidence in relation to our research question and thus support the validity of the answer. Our research study deals with the exploration and then the confirmation of a phenomenon in the Web systems context. The two aspects (exploration and confirmation) of our research study are described below.

Within a single research study, there can be both exploratory and confirmatory aspects where exploration is often called for at the outset and confirmation near the end (Miles and Huberman, 1994). In this research, first we have focused on exploratory study to explore the state of the art by using the literature review and content analysis as a data collection and analysis method respectively. Additionally, we have used interview as a data collection method to explore the state of the practice and identify a set of features that CIA approaches should address that were deemed essential from the practitioner's perspective. For exploratory study, it has been reported that close ended devices are more likely to be inappropriate (Orlikowski and Baroudi, 1991); therefore, we have selected interviews as an approach to explore the dynamic of our research investigation. Since our interview was semi-structured and mostly comprised of open-ended questions, subsequently, we have analysed the data both qualitatively and quantitatively. The details of the selected data collection and analysis methods are given in Section 3.7.

We have also focused on illustrating the process model of CIA with an exemplar system, and therefore a well-structured instrument design of proof by demonstration is a logical choice. Proof by demonstration is an opportunity to demonstrate the capabilities of the research artefact on a small sample and in a controlled manner. It can also serve to inform an alternative analysis for an organisation while adopting a new approach. Proof by demonstration also helps to determine whether the new approach is appropriate for use (by intended users) and how easily it can be executed, adopted and configured. Based on the experience of proof of concept; overall exercises and results can serve as a guide for the user to adopt the developed approach in a real-life project setting.

3.3 Research Design

Research design describes the more appropriate choices for the selection of methods including data collection and data analysis methods. There are two main types of research methods mostly used in software engineering research (i) Quantitative and (ii) Qualitative. Here it will be important to briefly describe the different research methods to make an informed selection of those methods for our research study.

3.3.1 Quantitative research

Mostly, the aim of quantitative research approach is to test a pre-determined hypothesis and produce generalised research outcomes. Therefore, quantitative research is considered more useful for answering 'what' questions. Mostly, researchers employed research methods such as experiments and surveys to carry out quantitative research. Neumen (2000) has reported that statistical data, tables, or charts form the basis for quantitative data analysis and further describe how these data support the validation of hypothesis. In this research, we have used a quantitative approach to analyse close-ended interview questions both at Stage 2 and Stage 3 of the research methodology.

3.3.2 Qualitative research

The aim of the qualitative research approach is to provide an understanding and explanation of complex psychosocial problems. Qualitative research is most useful for answering 'why'? and 'how' questions. It has been reported by various researchers that the role of qualitative research is significantly important if the research is exploratory in nature or if the research is an attempt both to understand the experience and find the significance of a given situation to a group of individuals (Strauss and Corbin, 1998, Symon and Cassell, 1998). In the same pursuit, we have used a qualitative research approach both to understand the current practices of CIA in a Web systems context and to find the adequacy of CIA approaches to support early identification of change impacts in Web systems. Mostly, in qualitative research the data is collected through interviews, observations, site notes or other varieties of pictorial or written materials, which are then analysed by a coding approach to determine themes/concepts/ideas (Strauss and Corbin, 1998). We have used the same approach, which allowed us to analyse data gathered during interviews. Additionally, during the case study we have used pre and post-study questions and analysed open-ended questions qualitatively. In the following, we have described two main reasons for using the qualitative research approach in our research study.

First, a qualitative approach provided us a better coverage of our research scope (Easterbrook et al., 2008) and thus led us to new and even unpredicted issues/problems and their possible solutions that were not obvious at the onset of the research.

Second, a qualitative approach enabled us to deepen our findings of the phenomenon under study. Subsequently, the qualitative research helped us to understand the underlying human-related components in phenomenon that are rich and comprises of many details and perspectives.

The use of both qualitative and quantitative research approaches at Stage 2 and Stage 3 guide us to focus on the concept of mixed approaches research and its relevance in our research study.

3.3.3 Mixed approaches research

Mixed approaches research is an attempt to legitimise the use of multiple approaches in answering research questions, rather than restricting or constraining researchers' choices (Johnson and Onwuegbuzie, 2004). Mixed approaches research can be defined as the class of research where researchers mix or combine different research approaches. For example, researchers use quantitative and qualitative research techniques, methods, approaches, and concepts in a single study (Bazeley, 2002). According to this principle, researchers collect multiple data using different strategies, approaches, and methods in such a way that the combination is likely to result in complementary strengths and non-overlapping weaknesses (Brewer and Hunter, 1989). Greene et al. (1989) reported five major purposes for conducting mixed approaches research including (1) triangulation (2) complementarily (3) initiation (4) development and (5) expansion. However, triangulation is mostly used for combining methodological approaches, data sources, or data analysis methods. In this research, the intent of using triangulation is to decrease, negate or counterbalance the deficiency of a single approach, thereby increasing the ability to interpret the findings.

3.3.4 Triangulation

Triangulation is useful for combining different methodological approaches during the investigation of the same phenomenon. Neuman (2000) has described that there are several types of triangulation, such as triangulation of measures, triangulation of observes, triangulation of theory, triangulation of data, and triangulation of method. In this research, we used triangulation of data, through mixing qualitative and quantitative styles of data analysis. A description of our triangulation approaches is given below.

(1) In this research, the use of open-ended and close-ended questions both for the interview and pre and post-study (during case study) meant that we have analysed the data both qualitatively (open-ended questions) and quantitatively (close-ended questions). For close-ended questions, the responses received from participants were tabulated in a spread sheet and analysed with basic statistical methods such as frequency analysis. However for open-ended questions, we have considered the elaboration of answers from interview questionnaires typically as a function of qualitative data. We have used a qualitative method and utilised interviews as an instrumentation of qualitative approach. Further, interview data was used to determine themes by coding technique. Subsequently, frequency distribution as a statistical analysis method was further used to reveal data that was not really evident in the details of qualitative analysis. This triangulation of data helped us in addressing the limitation that exists for both research approaches and could neutralize the biases of both approaches (Creswell, 2009).

(2) In triangulation, there are two possible ways to carry out the study. One is using two methods sequentially and another way is using the two methods in parallel or both simultaneously. We have used the same concept of using data analysis methods (qualitative and quantitative) sequentially with a view to provide corroborating evidence for the conclusions drawn.

(3) Triangulation facilitates us to tackle the limitations and principle issues underpinning a single approach, thus substantially providing a basis for deriving “facts” from a number of interlinked research questions in our research study (see Table 3.1, mapping from research questions to research study).

3.4 Mapping Research Objective, Questions and Strategies

As Crotty (1998) has pointed out, it is important to define the mapping between the objective of research and the questions that the research will be addressing to ensure the trustworthiness of the research and its findings. Table 3.1 provides the mapping among research objectives, research questions and research strategies employed during our research study.

Table 3.1: Mapping Research Objectives, Questions and Strategies

Research Objectives	Research Questions	Research Strategies
Research Objective 1: To analyse the characteristics of Web systems that need to be considered during Change impact analysis (CIA). Research Objective 2: To investigate scholarly/research literature for change impact analysis including existing approaches, methods, techniques and tools, and to identify gaps from the scholarly/ research literature while adopting existing CIA in Web systems.	RQ1: What specific characteristics of Web systems need to be considered for CIA and why? RQ2: What type of changes and their impacts are we interested in with Web systems? RQ3: What are the relative strengths, weaknesses of the existing CIA approaches, methods, techniques and tools while adopting these CIA approaches in Web systems? RQ4: Why is CIA important in Web systems? RQ5a: What does it mean by early identification of change impacts in Web systems? RQ5b: Why is early identification of change impacts important in Web systems?	Literature review
Research Objective 3: To	RQ6: What are the industrial	Literature review of

investigate industrial practices of CIA and to identify the limitations of existing CIA (in practice) in Web systems.	practices of CIA in Web systems? RQ7. How do the existing practices and approaches support early identification of change impacts? RQ8: What kind of information do Web developers currently use to perform change impact analysis? RQ9: What are the high level features that CIA approaches should address in Web systems?	commercial practices and practitioners interviews
Research Objectives 4: Based on the review of the scholarly/research literature and practices, extend the current theory by proposing an extension of existing CIA approaches to support early identification of change impacts in Web systems.	RQ10: Based on the review of scholarly literature and practices, is it possible to extend exiting CIA approaches? RQ11: Does the process model of CIA address the specific characteristics of Web systems? RQ12: Does the process model of CIA support Web developers/ architect in early identification of change impacts?	Development of the research artefact and illustration with an exemplar system (as proof of demonstration) and a case study. Discussion

3.5 Research Methods

A number of research methods can be applied to a research study in order to deeply understand research problems (Creswell, 2009) and subsequently provide an adequate solution to the problems. Mostly, during research design, it is desirable to select research methods by tapping into its strengths while mitigating its weaknesses (Bazeley, 2002). Indeed, the validity of the research results depends on how and in what ways the research design addresses the weaknesses of the selected research methods (Easterbrook et al., 2008), as we have discussed in Section 3.3.4. In this research, we have used triangulation of data that results in complementing the strength and compensating the weakness of both quantitative and qualitative approaches.

As discussed before our research methodology (see Figure 3.1) consists of three stages and has employed different research methods for data collection and data analysis in all three stages. We achieved an agreement across methods and investigation approaches used in our study with other experienced researchers of qualitative, qualitative, and mixed-approach research. In Table 3.2 we describe the types of data collection, analysis approaches and methods used. In the following subsections, we describe in detail each research method and approach used in each stage of the research methodology.

Table 3.2: Data collection and Analysis methods

Stage	Approaches		Methods
1	Data collection	Qualitative	Literature review
	Data analysis		Content Analysis
2	Data collection	Qualitative and Quantitative	Interviews
	Data analysis		Coding and theme identification
3	Data collection	Qualitative and Quantitative	Case study and interviews
	Data analysis		Empirical analysis

3.6 Research Methods for Stage 1

In Stage 1 of our research, we focused on the investigation of the current state of the art of change impact analysis (CIA) in a Web systems context. In achieving this focus, we addressed research objective 1 and research objective 2 and developed a set of underlying research questions. We have used the literature review and content analysis as a data collection and analysis methods respectively in Stage 1 of the research methodology. This investigation is further complemented by reporting a study of comparing architecture level CIA approaches and their relevance in a Web systems context (Mehboob et al., 2009) in Stage 1 of the research methodology, as shown in Figure 1.

3.6.1 Justification of methods selected

We have selected the literature review to gain a comprehensive knowledge of the related areas, to keep us well-informed with what is current in the related areas, make new insights and contributions, and to use as a stepping stone for subsequent stages (Stage 2 and Stage 3). Content analysis technique is used for making valid inferences from the identified research literature.

3.6.2 Data collection methods

In Stage 1, a broad spectrum of scholarly research literature on Web systems, CIA approaches and other related area was investigated at a detailed level. For example, considering Web systems, a more focused analysis of research literature was carried out including change in Web systems, architecture design of Web systems, and the characteristics of Web systems considered important for change impact analysis. The data used in this phase was therefore relevant research literature which was qualitatively analysed to address the research objectives and research questions. We have treated the literature review as a first part of the problem analysis as shown in Figure 3.1. However, the literature review was primarily done simultaneously with all three stages and has been extended to continue throughout the whole research study.

3.6.3 Data analysis methods

Content analysis is different from a basic literature review in that the researcher searches for predetermined themes and uses the material as data (Stockdale and Standing, 2002). Mostly, content analysis is used as a systematic examination of the scholarly research literature and for formulating inferences from supporting scholarly research literature (Leedy and Ormrod, 2004). As a data analysis method, we have focused on analysing the data qualitatively where the main objective is to investigate the characteristics of Web systems and change impacts in Web systems, existing CIA approaches, and their relevance in Web systems. We have used content analysis to synthesize the literature while addressing research questions. Content analysis facilitated us to analyse the presence, meanings and relationships of concepts/ideas within texts or sets of texts (e.g. books, thesis, journal and conferences paper, industry research papers, etc.) Additionally, content analysis facilitated us to make inferences about the messages within the integrated view of textual data and their specific contexts. As a result, we have identified a number of shortcomings toward existing CIA approaches and concluded those CIA approaches cannot be adequately adopted in Web systems context.

3.7 Research Methods for Stage 2

In Stage 1, the findings that resulted from undertaking the investigation for research objective 1 and 2 led our focus to research objective 3, and subsequently guided us to develop research questions for Stage 2, where we examined the industrial practices of CIA in a Web systems context. The research questions developed were then used to guide the development of questionnaires for interviews. In this stage, we selected interviews as a data collection method, and coding and theme identification as data analysis methods. In Stage 2, the investigation of the state of the practice gave us an insight into the industrial perspectives of CIA and supported us to reveal a list of high-level features that CIA approaches should address (Mehboob and Zowghi, 2009), in Stage 2 of the research methodology as shown in Figure 1.

3.7.1 Justification of methods selected

Interviews have been reported as a suitable research technique to be used for the acquisition of related domain knowledge. An interview is primarily based on revealing information from the practitioner's experience (Agarwal and Tanniru, 1990). Additionally, an interview supports the depth of information received and provides a better choice for the effort required during data collection (Kvale, 1996). Keeping in view, acquisition of knowledge, depth of information received and comparatively a better choice, we have selected interview as a data collection method for Stage 2. Since interview questions in Stage 2 were mostly open-ended questions the data was mostly analysed qualitatively (see Section 4.4.2). It has been reported by David and Sutton (2004) that coding and theme identification is possibly the single most significant act in the process of qualitative analysis. Therefore, we have used coding techniques that involved the identification of common themes within the interview data being analysed. Additionally, for close-ended questions, we have used a quantitative analysis approach. The data collection and analysis methods for Stage 2 are described in the next sections.

3.7.2 Data collection methods

Interviews enabled us to directly inquire the practitioners about their thoughts and subsequently provided the freedom for practitioners to describe in detail their opinions and beliefs related to our research questions. Face-to-face in-depth interviews were carried out and a list of questions were used during interviews as shown in Appendix C. In-depth interviews were used, as the aim was to get a deeper experience and perspective from Web developers while adopting CIA approaches in their Web system projects (see Section 4.4.1). Therefore, interview questions were mostly focused on gaining a better understanding of the potential issues/problems that participants faced during CIA practices and their possible solution. We recorded all the interviews. All recorded interviews were listened again and again and every possible effort was made to correctly convert each recorded interview into an interview transcript. For the purposes of confidentiality, all the interview transcripts (including questionnaire responses) were given a numerical identification code, based on the sequence they were received. In Stage 2 of our research, the data was mainly analysed qualitatively (as discussed in 3.7.3) whereby a search was conducted to identify the common themes that emerged from the participants' responses (see Section 4.3).

3.7.3 Data analysis methods

Numerous options for the analysis of qualitative data are reported in the research literature. In the broadest sense, for qualitative data mostly researchers use some type of approach to code the text and further to identify themes present in the text data (Ryan and Bernard, 2003). In this research, the interview transcripts were used for the purpose of coding and identification of themes. For the responses received from open-ended questions, the data mostly comprised of textual descriptions as provided by the participants. The textual descriptions were analysed using content analysis as proposed by Miles and Huberman (1994). After performing content analysis, interview data was coded and compared to identify themes and the significance of their similarities or differences. Here, the identification of themes was carried out by using a coding scheme that was mainly guided by the research questions (Harris, 1996). Further, codes were given meaningful names, indicating the themes underpinning the coded text. The subsections below describe five steps used to analyse the interview data.

Coding & theme identification

Coding is the process of analysing text data for the identification of themes and then making similar passages of text labelled with a code. These codes can easily be retrieved at a later stage for further comparison and analysis (Patton, 2002). Creswell (2009) has described three focuses to code the text including the identification of (i) themes, topics (ii) ideas, concepts (iii) terms, phrases and keywords found in the data. Our coding approach was focused on the combination of identifying terms, phrases and keywords as well as themes/topics found in the text data. Further, our coding approach is developed based on the five criteria reported by Ryan & Bernard (2003). These five criteria include (i) kind of data types, (2) required expertise, (3) required labour, (4) number and types of themes to be generated, and (5) issues of reliability and validity. These criteria are discussed below.

Kind of data types -We have found a rich narrative data throughout the interview transcripts. As transcripts texts became larger and more comprehensive, we looked for transitions and linguistic connectors between texts. Further, we also looked for the missing data from long responses of open-ended questions as it facilitated us to identify new themes by the elicitation of data.

Expertise -We have selected the approach to look for repetitions, similarities, differences and transitions in the interview transcripts text. This approach is easy to apply and requires less expertise in language skills.

Labour -We have taken into consideration the frequency of words/phrases and applied quick sorting to identify quite a few themes as it required less labour. However, we avoided those techniques that require more labour and produce fewer themes such as word co-occurrence (that is based on the idea where a word's meaning is related to the concepts to which it is connected) techniques as reported by Ryan & Bernard (2003).

Number of themes - For the text from interview transcripts, we have employed repetition, similarities and differences, transitions and connectors that occur frequently in open-ended interview responses and this supported us in producing more themes. Whereas for text processing techniques, we used cutting and sorting to identify an intermediate number of themes. We found that cutting and sorting were the most versatile techniques that facilitated us to sort the coded text into a comprehensive list at different level of abstractions (e.g. process level support for CIA, tool support for CIA, traceability and dependency analysis for CIA, etc.) and it has been the mostly recommended technique for theme identification (Creswell, 2009).

The first task we had when using interview transcripts was to consider how to break the text down to the unit of analysis. There are number of ways doing this including by research section (whole interview, responses to interview questions); by grammar (sentences, paragraphs/passages, etc.) by formatting (lines, page) and by theme (themes, ideas), etc. We chose passages of text from the interview data as a unit of analysis. Before coding the interview data, the researcher familiarised himself with the material by studying it in different ways- for example not just reading it line by line, but also dipping in and out, looking at what was missing, and looking for the flow of interview responses e.g. struggle to answer and the questions avoided, etc. We have adopted the concept provided by Ryan and Bernard (2003) that the themes can be considered as a conceptual linking of expressions, however, there were many possible ways in which a expression from the interview text can be linked to themes.

Approaches to coding initiation

There are two possible ways described by Creswell to initiate coding. The first is to use priori-code such as pre-existing theories to identify themes and the second is to let new codes emerge from the text as the researcher analyses it (Creswell, 2009). It has been reported that priori codes and their associated themes come from the characteristics of the phenomenon being studied, from literature reviews, theoretical orientation and personal experience (Creswell, 2009). Additionally, researcher decisions about what topics to cover and how best to query participants about selected topics are a rich source of priori-code (Ryan and Bernard, 2003). We have used a combination of (i) priori-codes as identified from the literature review (ii) intuitive feelings of the researcher about the possible codes and (iii) grounded codes that emerge from the transcribed interview text. The process of coding was initiated when we started to translate the key words, phrases and ideas into more abstract concepts (themes), which then became short descriptions for our codes. In fact, the first pass at identifying themes came from the interview questions and we coded it accordingly. We gave these codes a label and a short description while remaining aware of the fact that themes will emerge during subsequent analyses. We realised subtle but significant differences among themes and we distinguished them in the coding list. For example, we labelled each code as C1, C2…C15 along with their short descriptions.

Typically when we started coding, we used some codes description from the research topic selected (such as business process changes, impacts on architecture, steps of CIA, information used, etc.) and also sought other themes (represented by new codes) that were revealed from the transcribed interview text. There are many variations of techniques that can be adopted for identifying a preliminary set of themes. Below we have described some observational techniques to identify themes from the transcribed interview text and processing techniques to process the coded text.

Techniques to identify themes

We approached interview data in a way so that we remained open minded about what can be coded, and started to observe important themes in the interview text. The most common procedure recommended in the research literature is 'constant comparison' in a way that every time we selected a passage of text and coded it; we compared it with all those passages we had already coded that way. We adopted the approach that involved searching for similarities and differences by making a systematic and constant comparison across passages of text. The search for similarities and differences, and constant comparison ensured that our coding was consistent and allowed us to consider the opportunity that either some of the text coded a particular way didn't fit or that there were other dimensions in the passages that might also be coded in another way. In addition to the constant comparison we have used a combination of other text observational techniques described by Ryan and Bernard (2003) including key-words-in-context (KWIC), search for missing information, transitions, connectors, pawing and unmarked-text to identify themes.

Throughout the coding and theme identification activity, we have followed a strategy that if a theme was identified from the interview data that did not quite fit with the existing codes then we looked for a new code for it. As we coded the interview text data we were likely to create new codes, and therefore we needed to go back and check the passages we coded previously to create this code. We have also checked that if there were any more passages that were coded at the newly created code i.e. something we came across later and changed the way we wanted to code interview text data. In fact, we had not noticed a new theme in the interview text until we had coded a number of interviews. It was a lengthy process and at one stage we found that previously coded passages did not need to be coded at the newly created code but we could not know this without checking. By using both the identified themes from the interview data and a few obvious responses for CIA received from participants, it was possible to extract all the explicit and implicit information, revealing the list of features that CIA approaches should address. For example, it happened on several occasions that participants explicitly mentioned the feature that a CIA approach should address and the weakness of the current adopted approaches. These features intents were specifically coded and in the theme generation activity was merged with the existing theme.

Following are the text processing techniques that we used as a way to process the coded text resulting from observational techniques.

Cutting and sorting

The technique of cutting and sorting was adopted to cut the coded texts from interviews transcripts and then we collected all those coded texts on a question by question basis. Organising these coded texts and re-reading them together, was an important part of the text processing and analysis. Similar to theme and code identification techniques, we used analytic/theoretical approaches where processing of the coded texts were carried out by analytical thinking of the researcher while focusing on what is emerging from the collection of coded text and what is not.

After cutting the coded texts from interview transcripts, eventually, we found a large number of codes and we found it necessary to sort them into some order, group or category. One possible way was either using a hierarchical or non-hierarchical coding to categorise these codes (Ryan and Bernard, 2003). We used a non-hierarchical coding (flat coding) approach. Interestingly, when re-organising the coded text by the non-hierarchical approach, we observed two important aspects. First, we found that we were able to start categorising our codes where the categories were formed based on comparison and analysis between interviewees' responses. Second, we found that taking into consideration the categories of similar codes, several new possibilities of codes further emerged. Strauss and Cobin (1998) refer to the identification of new codes as 'dimensionalising'. The code and the underlying dimension became observable through review and were revealed through subsequent comparative analyses. We have found that several codes can be grouped together as types or kinds of something. For example, we have found several codes description such as (i) tracing from one system component to another, (ii) linkages between system artefacts and (iii) mapping among system components. In those cases we moved them together and put those codes under a major code: the kinds or types of which they all can be represented e.g. traceability. These final level themes were listed as a list of features that CIA approaches should address.

Results Triangulation and Filtering

A triangulation and filtering approach was selected in order to get a more representative and desired set of features for CIA in Web systems. Our triangulation approach employed two sources of information (i) the list of features generated from interviews and (ii) gap analysis results from the scholarly research literature. Further, a filtering approach was adopted to reduce the inconsistencies and redundancies in the list of features and to develop a final, most representative and desired set of features that CIA approaches should address.

3.8 Research Methods for Stage 3 (Development, Illustration and Validation)

In phase three, we focused on the possible extension of existing CIA approaches and exemplification of extended approach to address the research objective. The research objective (Objective 4) for Stage 3 was developed based on both the findings both from the review of scholarly/research literature and industrial practices. In achieving this objective, a process model of CIA has been developed as an extension of existing CIA approaches. Further, the execution of the process model of CIA has been illustrated through an exemplar Web system.

Iivari and Lyytinen (1998) considered that a development approach was a constructive type of research. It is grounded on the philosophical belief that development always involves the creation of some artefacts "…*which may be either purely conceptual artefacts (models, frameworks and procedures) or more technical artefacts with a 'physical' realisation (e.g. software)*". Furthermore, Hevner et al. (2004) has described the possible artefacts of system development research as construct (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices), and instantiations (implemented and prototype systems). Further, Burstein and Gregor (1999) argues that the development of model, system or prototype, both being subjects of software development research, imply that the researcher of that research artefact will carry certain views and values throughout the research.

In the following discussion we take the view that research follows a pattern of "problem, hypothesis, analysis, argument". In this view, problems exist in a research domain and are encountered by observation. Once a hypothesis is formed, analysis can be used as an attempt to confirm and to generalise on hypothesis. This analysis may take the forms as varied as formal proofs, developed systems, and opinion surveys (system development in IS research). The results of the analysis become the argument (and evidence) in defence of the original hypothesis. Note the view of research methodology permits system development to be a perfectly acceptable piece of evidence (artefacts) in support of a "proof" where proof is taken to be any convincing argument in support of a worthwhile hypothesis. System development can be considered as a "proof-by-demonstration" (Nunamaker et al., 1990). Here the role of system development is pivotal due to the fact that the developed system serves both as a proof by demonstration for the fundamental research and provides artefacts that becomes the focus of expanded and continuing research (Oates, 2006). In our research study, development of research artefacts is used with other research methodologies including literature reviews and interview based studies to form a comprehensive research methodology.

Judicious selection of compatible settings approaches and an illustration method for the execution of research artefacts is a critical part of the research methodology. However, selected approaches and methods vary from one research study to another. Indeed the selection of approaches and methods is primarily based on the research question(s) asked, the research artefact developed and the notion of validation. For example, if there was at the outset a serious question about whether something could be achieved at all, simply exhibiting an implementation is enough validation. However, if the question is how to improve on current practice, the same implementation would probably be completely unsatisfactory as the sole evidence. In other words, "Look, it works!" is sufficient proof only if the original question was whether it could be done at all. As ideas mature into practical technologies, the character of open questions does change. Whereas clear articulation of the problem may be a significant contribution at the outset, the questions, as a technology/approach enters practice, may involve formal analysis and evaluation of effectiveness relative to other technologies/approaches. Since the essential questions change, so will the idealised research settings, development of suitable research artefacts, and the appropriate validation techniques.

3.8.1 Development of the process model of CIA

In this research we aim to develop a process model of CIA which focuses on early identification of change impacts in Web systems whilst taking into consideration the characteristics of Web systems. Failure in attempt of early identification of change impacts leads to a typical problem in Web systems where detailed design or implementation actually begins before the impacts on architecture design are adequately identified, as discussed in Chapter 2. The proposed model should be able to address this problem and provide support for early identification of impacts, and subsequently also address features that practitioners considered important for a CIA approach in Web systems.

The proposed model, referred to as PMCIA (Process Model of Change Impact Analysis), was developed to support early identification of impacts in Web systems, by enabling correct and adequate identification of impacts on architecture design resulting from business processes changes. The development of PMCIA is based on:

(i) Investigations of the state of the art suggest that current architecture CIA approaches may not be suitable to adopt in a Web systems context and indicates an extension of CIA approaches for Web systems to support early identification of change impacts in Web systems.

(ii) Investigations of the state of the practices imply that significant attention and effort is still needed in this area. The research findings based on the industrial perspectives of existing CIA approaches in a Web systems context suggest a list of high level features that CIA approaches should address. Subsequently, investigating the state of the practices provides guidance on how to extend current CIA approaches.

(iii) Investigation of the relationships and connections between business processes and architecture design; as well as among architecture design entities by taking into consideration the related characteristics of Web systems. Subsequently, the proposed solution has employed both the traceability relationships between business processes and architecture design (resulting from traceability matrix) and the dependency relationship among architecture design elements (resulting from architecture design information).

(iv) Consideration of the necessary infrastructure for the process model of CIA, that fill the gaps identified both from the state of the art and state of the practice, by addressing a set of features that a CIA approach should possess. This detail, in relation to the concept and development of PMCIA, will be presented later in Chapter 5.

3.8.1.1 Justification of method selected

In order to fill the gap identified from the state of the art and in response to the feature identified from the state of the practice, we receive a clear direction for the development of a process-based approach as an extension to existing PMCIA approaches. Nunamaker et al. (1990) has described a list of research classifications including the development type of research. The development type of research is the systematic use of scientific knowledge directed toward the production of useful materials, devices, systems, models or methods, including design and development of prototypes and processes (Nunamaker et al., 1990). In our research study, the knowledge received both from the state of the art and the state of the practice and their systematic use leads us to the development of the process model of CIA. Specifically, a number of features that are deemed essential by industrial practitioners guide us to the question of how. For example, these features include the structured and consistent execution of analysis task, information re-use and effectiveness of impacts identification, which arguably are the fundamental aspects of the process approach of CIA. Thus, the natural choice is the development of PMCIA.

3.8.1.2 Data collection methods

As described in Section 3.8.1, the investigation results both from the state of the art and state of the practices are used for the development of PMCIA. Therefore, two data collection methods are also intuitively considered during the development of PMCIA. The first is the synthesis and critical analysis of the state of the art (related research literature) on the topic of Web systems and their characteristics, architecture design developed to support business processes in Web system and change impact analysis approaches. The detail of literature review is covered in Section 3.6.2. Another data collection method included was the investigation of state of the practices, commercial and industrial practices of CIA in Web systems. The detailed data collection method used for the investigation of commercial practices is covered in Section 3.7.2

3.8.1.3 Data analysis method

There are two ways data analysis was performed during the development of PMCIA. The first was the mapping of the findings both from the state of the art and the state of the practices, so as to seek research direction and any overlap between these two investigations (if any). The findings from the review of scholarly research literature also highlighted the inadequacy of existing CIA approaches while addressing few aspects (see Section 4.5.2). Interestingly, these five aspects were also covered by the investigation of the state of the art, which confirms that our listed high-level features are indeed relevant and reflects a reasonable coverage of features that CIA approaches should address.

Secondly, for a more focused research direction, we have analysed and then classified high-level features into five specific focused areas. The purpose of classification of high-level features is to provide focused areas for analysing the group of high-level features together in relation to change impact analysis. The classification supports to extend the CIA approaches by providing a structure which can be used for a clear research direction (in the same area of research). This classification is covered in detail in Section 4.7 and those focused areas were taken into consideration during the development of the process model of CIA.

3.8.2 Illustration of the process model approach

The execution of process model tends to gather the preliminary demonstration of PMCIA while understanding either the developed PMCIA approach support for early identification of change impact in Web systems. We focused on the proof by demonstration approach by using an exemplar Web system to execute the process model of CIA. With the exemplar Web system and by comparing the results received both from PMCIA and without PMCIA, we studied the results and reported the findings. In the following section, we briefly discuss the proof by demonstration as an appropriate approach and further details of it. The details about Proof by demonstration for PMCIA by using an exemplar Web system is presented in appendix E.

3.8.2.1 Justification of methods selected

Nunamaker et al. (1990) stated that if the proposed solution of the research problem cannot be proven mathematically and tested empirically, or if it proposes a new way of doing things, researchers may elect to develop a system to solve the research problem and demonstrate the validity of the solution based on a suggested new method or technique through proof-of-demonstration.

Proof by demonstration is the first opportunity to test the technical capabilities of a research artefact and experience how it may operate in a real software project, providing opportunities for users to gain practical experience. Illustration of research artefacts as proof by demonstration requires certain care to be taken so as to avoid (or at least recognising) biases of the researcher that may influence the demonstration activity. In order to reduce the probability of bias we have adopted a set of activities in relation to proof by demonstration that can be divided into two distinct phases including (i) preliminary activity and (ii) conduct of proof by demonstration.

Preliminary Activities

Defining the purpose, goals or objectives for proof by demonstration

Defining the purpose and goals for proof by demonstration is essential to its success. The goals, we set out for this study are given below.

Goals for by demonstration

- Determine if the adopted approach (PMCIA) meets the intended needs
- Acquire experience and information for a real-life project rollout
- Gain a general acceptance of use

Specifying these objectives will help us to identify aspects that we should monitor during the exercise run.

What should a proof by demonstration accomplish?

- Clarify users' understanding of research artefact (PMCIA)
- Verify the adequacy of process specifications
- Validate the usefulness of the process model
- Verify that the process model is technically stable
- Verify the execution and overall functionality of the process model
- Identify and address obstacles (if any) for the full scale implementation
- Produce samples of all inputs and outputs

A well-defined proof by demonstration exercise, with carefully developed purposes, goals, and objectives will make success an easier task to accomplish. In the next sections, we address the need for establishing the success criteria at the outset of proof by demonstration, and the effect this has in assuring smooth deployment in real-life Web system projects.

Establishing success criteria

In relation to the proposed PMCIA, proof by demonstration needs to achieve the following criteria in order to declare its success.

- Easy use for each step and activity of PMCIA
- Support to identify impacts on architecture design resulting from business process changes
- Correct and adequate identification of impacts

Outlining the benefits of proof by demonstration (and risks of not conducting one)

Proof of concept provides an early visibility of PMCIA execution in the Web system project.

- Conducting proof of concept exercises will potentially help Web developers to understand the execution of PMCIA and the benefits of its use. This will further assist in real-life Web system projects.
- The proof of concept provides a tangible way for the potential value of PMCIA use and why it is necessary (or worth the effort). Thus, the results of proof of concept are evidence of the PMCIA's immediate and tangible value.

Proof of concept reduces the risk of investment by identifying technical risks (e.g. compatibility problems with existing systems and infrastructure), procedure changes (workflow issues), and information required as inputs. Information gained as a result of proof of concept will mitigate any possible associated risks and prepare for full deployment.

Conduct of Proof by Demonstration

Defining the scope and duration of a proof by demonstration

The proof by demonstration provides an opportunity to employ PMCIA for an exemplar system within the set environment. Limiting the proof of demonstrations' scope to a manageable size (an exemplar system depicting a typical Web system) allows sufficient execution of PMCIA to:

- Determine whether PMCIA is appropriate to be used for the intended purpose (early identification of change impacts in Web systems)
- Assess how easily it can be adopted
- Ascertain how it can best be deployed in real a Web systems project.

Proof by demonstration is particularly useful for complex implementation of research artefacts such as a process model, where the proposed approach is new, the implementation represents a significant change in the development task (operating procedures), and when an adequate organisation as a case study may be difficult to find. A proof by demonstration provides the practical experience necessary before rolling-out a process model in a real-life project in an organisation. This permits the project team to validate the research artefacts as well as providing an opportunity to identify any tailoring (if required).

3.8.3 Validation of process model approach

The validation activity mainly focused on the question of either a developed CIA approach support for early identification of change impact in Web systems or not in a commercial setting. A number of data collection and analysis methods were used during the validation activity. In the following section, we briefly discuss a case study as a research method, and other data collection and analysis methods. A detailed discussion about the case study design adopted in our research is presented in Chapter 6.

3.8.3.1 Justification of methods selected

In Stage 3, we have selected a case study as the research method, a pre-study and post-study interview as the data collection method and analysed the data both qualitatively and quantitatively. Wohlin et al. (2012) argued that a case study is suitable for industrial evaluation of new approaches, because unlike in an experiment, both scale and variables are inherent to the situation rather than to the experimental setup. Additionally, Yin (2002) has described that a case study should be selected when (i) the researcher wants to provide an explanation of a phenomenon (ii) the subject of the investigation cannot be moved into the laboratory and cannot be controlled (iii) the phenomenon is contemporary and many different sources are available for investigation. We have employed both Wohlin et al. (2012) and Yin's (2003) description in support of our argument, that validation of the process model of CIA cannot be done like a controlled experiment in a laboratory; therefore we have selected a case study for the purpose of validation.

We have focused on a set of research questions (see Table 1) derived from the objectives of Stage 3. Further, these research questions guided us in the selection of the subject and data collection method for our case study (Easterbrook et al., 2008). There are two broad classes of sampling: (i) purposive sampling and (ii) random sampling. Purposive sampling is used to serve a very specific need or purpose, they don't truly represent the population and thus are less desirable than random (probability) sampling (Gilbert, 1996). Random sampling advocates that the sample is selected in a way that each and every unit of the population has an equal and positive chance of being selected. As a result, a sample determined by random sampling is likely to represent the overall population (Levy and Lemeshow, 2008). Therefore, we have used random sampling with the aim to enable us to assertively generalise results from the small sample to the larger population. For our case study, we have selected a sample of a single case that met the criteria for our case study research, and therefore the selected sample of a single case is relevant to our study. We believed that a single case was sufficient to be a typical case of common situations and possibly can be done within the stipulated research time constraint.

3.8.3.2 Data collection methods

Mostly, case study supports an in-depth study of a single case instance or unit of analysis in which certain attributes are studied (Easterbrook et al., 2008). As a sample of a single case (or unit of analysis) for our case study, we have selected a Web system project which allowed us to validate the developed process model of CIA. We will use pre-study and post-study interview questions to develop an understanding of the practice before and after the enactment of the process model of CIA. Besides pre-study and post-study interview questions, we will also use the following data collection methods.

Project documents and archival records

As a supportive source of data, we have also analysed project documents, archival records from different development phases and other relevant organisational data. We have also examined actual change data collected, design change decision repositories and the development and updates of relevant documentations.

Observation in case study organisation

During the case study, observations were made in the organisation by studying existing CIA practices, enactment of the process model of CIA, participation in project meetings, and how the process model of CIA supported early identification of impacts in Web systems.

3.8.3.3 Data analysis methods

Mostly, data analysis in case study research consists of examining, categorising, interpreting, and combining quantitative and qualitative data to address the research questions (Yin, 2003, Creswell, 2009). In our research study, we employed qualitative and quantitative analysis for open-ended and close-ended questions both from pre-study and post-study interviews. By analysing the data qualitatively and quantitatively, we have focused on research questions for Stage 3 (see Table 1) and ensured that the data analysis results adequately address the research questions posed for Stage 3.

3.9 Research Reliability and Validity

Due to different philosophical and methodological approaches used to investigate engineering and human activities in software engineering research, the importance of reliability and validity of research have been increased substantially. The reliability and validity of research guides the appropriateness of the methods selected and the quality of data (Easterbrook et al., 2008). A number of steps are usually taken by researchers to minimise the threats to validity and reliability through a careful design of the research study, including the instruments used and the process of data collection and analysis (Stake, 1995). The following sections describe how we have handled the issues of reliability and validity in our research.

3.9.1 Reliability

In this research, researchers aimed to address the concern of reliability by ensuring that the data collection methods used during the interviews produced stable and consistent results. This was achieved by conducting pilot testing of the interview questionnaire with participants who were external to the research, but who matched the established selection criteria. Pilot testing for both Stage 2 and Stage 3 were useful activities as they provided suggestions which led to the adjustment of the questionnaires structure, addition, deletion and modification of questions. Additionally, the interviews and case study procedure are documented in detail and the sequences of all activities are clearly described to ensure reliable, consistent and stable results (see Section 3.7, 3.8, Chapter 4 and Chapter 6).

Another concern of reliability is the representative reliability (Neuman, 2000), where researchers ensure that the sample selected are representative of the population. In our research we have used Web developers/architects and a case study from a Web systems development organisation (see Chapter 7). The sample size for Stage 2 was of twelve Web developers/architects having at least three to four years of Web systems development experience, and who especially have experience of architecture level change impact analysis in a variety of Web system projects. This strict selection criteria reduced the number of interviewees; therefore, the researchers have chosen subjects who represent different roles (including developer, architect/designer and system analyst - but each one had extensive involvement during change impact analysis activity as a decision maker to address change impacts in their Web systems projects) and belong to different organisations in order to provide the most extensive and comprehensive material for the analysis. In Stage 3, the researchers selected a case study system as a typical case of common situations and which possibly could be done within the stipulated research time constraint.

Creswell (2002) has identified three types of validity as internal validity, external validity, and construct validity. It has been argued that there is no ultimate demonstration of validity as there are many ways of considering, specially qualitative data in a study (Dey, 1993). However, one can maximise the validity by maximising the clarity and agreement as we have discussed below.

3.9.2 Internal validity

Creswell (2007) refers to internal validity as a threat to draw correct inferences from the data resulting from experimental procedures, treatments, experience of subjects and the ability of the researchers. In this research, internal validity concerns were addressed by the use of several methods: (1) through triangulation; (2) checking the interview findings with the participants; and (3) after transcribing the interviews, checking that the interview conversations matches the transcribed interview transcripts.

In Stage 2 and Stage 3 of our research, we have used triangulation of data by combing the qualitative and quantitative data analysis. Corroborating different approaches has provided us greater confidence than that held in singular conclusion. In Stage 2 the questionnaires were comprised of open and close-ended questions. The results from the closed-ended questions were analysed quantitatively using basic frequency analysis, while results from the open-ended questions were analysed qualitatively where themes from the participants' responses were examined. We have adopted the approach to maximise the validity as described by Ryan & Bernard (2003). First, we made the researcher judgment quite explicit and clear during themes identification for the reader so as to understand the researcher's conclusion. Second, we addressed the validity as a key to the agreement across methods and investigation approaches used. Having an agreement across methods built our confidence by knowing we have identified suitable themes in the same way other multiple investigations do. All these different approaches and sources helped us in ensuring internal validity. Additionally, as recommended by different researchers (Lincoln and Guba, 1985), we have utilised interviewees as an opportunity to examine and comment on the interview findings. Feedback on interview findings helped us in ensuring accuracy of the data. In Stage 2, we also used an approach where the researcher listened to the recorded interviews on audio and compared them with the transcribed interview transcripts.

In Stage 3, internal validity was applied to the proposed approach and relevant queries were proposed by Burstein & Gregor (1999) for the process model of CIA as:

- Does the process model of CIA work? Does it meet its stated objectives and requirements?
- Were any predictions made in the study and how accurate were they?
- Have rival methods/approaches been actively considered?
- Were the conclusions concerning the worth of the process model of CIA considered to be accurate by the system evaluators? If not, is there a coherent explanation for this?

3.9.3 External validity

External validity is related to the researcher's concern about whether the research study findings can be generalised or other researchers can apply the findings to their own research (Leedy and Ormrod, 2004). For the interview study, in Stage 2 researchers used a representative sample: - a strict selection criteria that reduced the number of participants to twelve, and chose the participants who represented different roles and belonged to different organisation in order to provide the most extensive and comprehensive material for the analysis. In Stage 3, considering the external validity as an approximate validity with which conclusion is drawn about the ability to generalise, we have focused on queries proposed by Burstein & Gregor (1999) for the process model of CIA as:

- Are the findings congruent with, connected to, or confirmatory of prior research questions?
- Are the outcomes of the process model of CIA described in conclusions generic enough to be applicable in other settings?
- Does the researcher define the scope and boundaries of reasonable generalisation from the study?

Indeed, based on above facts, our research findings will not be generalised to populations. i.e. all Web development organisations, but they will contribute significantly to the current practice and theory. However, the findings can be claimed to be applicable to an organisation with similar settings (and Section 7.2). The change impact analysis problems and features that were identified guide us that process model of CIA - as a solution to problem - can be suitable to some extent and degree in most of the Web development organisations. Additionally, for the interview study and case study, the steps and the approach used are described in sufficient details and clearly documented; therefore, they can be adopted and repeated by other researchers and practitioners in their organisations.

3.10 Ethical Considerations

Approval from the Ethics Committee of the university was sought for the data collection and analysis activities conducted during both the interview study and case study. The questionnaires used were submitted to the Ethics Committee together with the complete ethics application document, as specified in the university guidelines for ethical research. The approval document may be observed in Appendix D. However, a formal consent (see Appendix B: research consent form) was signed and obtained each time from the concerned participants and each had the freedom to withdraw from this research at any time and/or to contact my supervisor or the university at any time. During this research, organisational details, participants' personal details and responses were kept confidential and were known only to the researcher and supervisors involved.

The responses provided were kept in the strict confidence and in accordance with the University of Technology Sydney's Privacy Principles.

Where the participants' data was used for scholarly publication purposes, the names and personal details of the participants were not included so as to protect the identity of the participants. In addition to this, throughout the research, every possible effort was maintained to ensure that participants were protected against harm and provided with informed consent, and that the research was conducted with integrity.

3.11 Chapter Summary

This chapter described the research methodology adopted for our research study. We have discussed exploratory and confirmatory approaches as a selected research approach, which then required the data to be collected and analysed. We have listed a number of questions which needed to be answered from the core research questions. Some of these questions have already been addressed through our analysis of the scholarly research literature in the previous chapter (Chapter 2), others will be addressed through the development of a process model of CIA, and following the validation of process model of CIA by a case study.

From a practical standpoint, it is expected that if a new system or method is constructed it should provide a better solution to the problem than existing system or methods. In our research study, we have shown from the literature review and industrial investigations that we are aware of any known partial solutions to the problem and subsequently have demonstrated how their proposed new method improves on these known approaches/methods.

Research design and the three stages of research methodology were discussed. Each of the stages has its own research objectives, and addressed its own set of and employed its own research strategy. The research objectives were integral in driving the research questions and the formulation of research strategies. This chapter addressed research threats in terms of reliability, internal validity and external validity.

The interview study and case study design, data collection and data analysis methods have been adopted according to research objectives, research questions and strategies as proposed at the start of this chapter. Multiple sources of data collection, including the literature review, interviews and the case study have been used, and the data was analysed both qualitatively and quantitatively where required. The way the data was managed and analysed was also explained in detail in this chapter. In the next chapters (Chapter 4 and Chapter 7), the findings from the data analysis activities will be discussed in greater detail.

Chapter 4: Features of a Change Impact Analysis Approach for Web systems

4.1 Introduction

A comprehensive review of the research literature investigated the current state of the art of CIA approaches and their relevance in a Web systems context and was presented in Chapter 2, Stage 1 of this research. Particularly taking into consideration the characteristics of Web systems and the nature of change impacts in Web systems, this review revealed the significance of analysing change impacts in Web systems.

From the investigation results of Stage 1, it was ascertained that there are a number of shortcomings in the existing CIA approaches while adopting them in a Web systems context. Specifically, a review highlighted the inadequacy of existing CIA approaches in early identification of change impacts (impact identification on architecture design resulting from business processes changes) in Web systems. It is necessary to obtain insight of the current industrial CIA practices and approaches while adopting them during Web systems development and to seek possible confirmation and supporting evidence for the literature review findings.

In Stage 2 of our research, the focus is to identify how Web developers carry out CIA, so as to gain an in-depth industrial perspective of CIA approaches, and the possible limitations of those CIA approaches while adopting them in the Web systems context. This investigation fundamentally informs us about the current state of the practice, inadequacy of CIA approaches and subsequently tends to confirm the literature review findings. For the research activities in Stage 2, face-to-face interviews were conducted to get an understanding of the state of the practice and further used to derive a list of features that CIA approaches should address. Necessary descriptions and a possible classification of those features are also presented.

In this chapter, we describe in detail the research activities conducted for Stage 2. This chapter begins by providing, the objectives of the interview study in Section 4.2. We then describe the interview study setting in Section 4.3, discuss the data analysis results and findings in Section 4.4, present the classification of features that a CIA approach should address in Section 4.5 and conclude with the chapter in Section 4.6, where a summary of the research activity and findings are presented.

4.2 Interview Objectives

The objective of the interview study is to answer the research questions presented in Table 4.1. Specifically, the intention was to look at not only what practitioners thought about the CIA practice to support early identification of change impact in Web systems, but also what they believed would be the features of an approach to support early identification of change impact in Web systems. Furthermore, it was anticipated that the interview results would provide confirmation of the gaps identified from the literature review.

4.3 Research Questions

The interview study was adopted with the aim of gaining an industrial perspective of CIA approaches while adopting them in the Web systems context. We have presented research questions to support the research objectives of Stage 2 and the corresponding questions from the questionnaire to address those research questions. The research strategies formulated for this stage are also described. The key focus of research questions is to consider how well existing CIA industrial practices and approaches support Web developers to carry out change impact analysis, particularly with regards to supporting early identification of impacts and the features that CIA approaches should address.

Table 4.1: Stage 2 research objectives, research questions, corresponding questions from questionnaire, and research strategies

Research Objectives	**Research Questions**	**Questionnaire Questions (see Appendix C for question)**	**Research strategies**
Research Objective 3: To investigate the industrial practice of	*RQ6:* What are the industrial practices of CIA in Web	Q1, Q2, Q3, Q4, Q5, Q6(a), Q6(b)	Literature review, Interview study

CIA and to identify the limitations of existing CIA (in practice) in Web systems.	systems? *RQ7:* How do the existing practices and approaches support early identification of change impacts? *RQ8*. What kinds of information do Web developers currently use to perform change impact analysis? *RQ9*. What are the features that CIA approaches should address so as to be adequately adopted in Web systems?	Q7(a), Q7(b), Q8, Q11, Q12 Q9, Q10	

4.4 Interview Study Setting

It has been reported that Web systems architecture's flexibility to change is more stressed than that of traditional software systems as Web systems architecture covers in detail both an information architecture and technical architecture (Lowe and Henderson-Sellers, 2003). As discussed in section 2.5.2, one of the important features of Web systems is that they use information intensively. Web-based businesses have increased their dependence on information, and as a result the importance of information architecture has become quite intrinsic (Evernden and Evernden, 2003). Irrespective of the sophistication of the functionality and the creativity of the interface, Web-based businesses are likely to fail without adequate alignment of information architecture with the changing business processes.

Possibly the most obvious reason to focus on the identification of impacts at information architecture is the recognised tight connection between business processes and information architecture (Evernden and Evernden, 2003, Fielding and Taylor, 2000, Tongrungrojana and Lowe, 2004b). Failure in attempt to address impact on information architecture design resulting from business processes changes may lead to problems that are more visible externally (and as such may affect users' interaction and relationships with external stakeholders). Given the importance of information architecture and demands to make adequate modification at information architecture design resulting from business processes changes, we have specifically focused on the information architecture while interviewing the participants.

The subsections below describe the four steps used to design and to conduct interviews.

4.4.1 Establishment of the participant criteria

In this step, we establish the criteria for participant selection for our interview study. As the interview questions were designed specifically to seek insight of impacts identification on architecture design resulting from business processes changes, we required typically those participants who have a number of years of experience in Web systems development - at least three to four years of experience. To gain a broader view during the interview study, we required the participants to have experience in various Web system projects of different size, from a broad spectrum of development areas (involving the internet, intranet, and extranet), of different application domains (educational institutions, hospitality, government and software vendors), and in various company sizes. Furthermore, participants needed to have particularly experienced performing change impact analysis at architecture design. Since the criteria were strict, the number of participants was reduced and only a limited number of interviews were conducted. Additionally, participants who represent different roles and belong to different organisations were also considered to provide the most extensive and comprehensive material for the analysis. The roles of the participants ranged from developer to architect or designer, but each one had extensive involvement in change impact analysis activity as a decision maker to address the impacts on architecture design of Web systems.

4.4.2 Questionnaire development

We used both open-ended and close-ended questions for our interview questions. Open-ended questions were selected as the most appropriate type because of the depth and nature of the topics covered, and possible variety and range of participants' experiences (Patton, 2002). Since open-ended questions avoid imposing any restriction on the participants, there were many different ways participants might choose to answer a question. No matter how carefully we word the question, open-ended questions might leave room for misinterpretation and result in an irrelevant or confusing answer. Thus, in order to restrict the participants and to elicit the information as foreseen, we also used a few close ended questions.

However close-ended questions were less in number because closed-ended questions might offer choices/options at hand. One drawback of offering choices/options at hand seemed to be making suggestions or leading questions, and thus it might influence the answers received according to participant's knowledge/thoughts. Conversely, open-ended questions allowed the participants to describe what was meaningful or important to him/her using his/her own words rather than being restricted to predetermined categories. As a result participants might feel more relaxed and thus more realistic data obtained.

By addressing the research questions for our interview study, as described in Section 4.2, the interview questions were developed. Additionally, while developing interview questions, we have focused on the review of published literature on CIA, Web systems, CIA approaches, potential issues toward CIA approaches and their possible solutions.

Table 4.2: Division of interview questions via topics

Section	**Topic**	**Questions**
1	General Information	2
2	Process information	2
3	Current practice	2

4	Issues/problems	3
5	Features/Solutions	2

Our interview questions comprised of twelve questions including eight open-ended questions and four close-ended questions, divided into five sections for each of the main topics covered as shown in Table 4.2. Additionally ten questions were developed that were related to projects and company information (demographic questions). The demographic questions were designed to screen participants and to refine the data sets relevant for the final analysis. Thus, the total number of questions were twenty two (twelve questions were related to the area of investigation and ten questions were demographic questions) and were determined by a desire to keep the estimated time required to complete the interview to approximately sixty to ninety minutes. The wording of the questions was also carefully developed in accordance with ethical considerations to ensure the participants could not be identified, and to keep the responses anonymous. As with all interviews, the wording of the questions was paramount, and all possible measures were taken to be as concise as possible and to avoid ambiguity.

4.4.3 Contacting potential participants

In our interview study, the sampling of the participants can be characterised as convenience sampling (Robson, 2002). Participants were selected mainly based on recommendation and availability. However, a few participants were approached based on their role in identifying impacts on architecture design in Web systems projects. A list was developed comprising of sixteen participants known to the researcher that matched the criteria. All sixteen participants were contacted via e-mail as an invitation to participate (See Appendix A). Whereas, finally ten replies were received and all ten participants agreed to participate and signed the consent form (See Appendix B).

4.4.4 Pilot study and refinement

A preliminary version of the questionnaire was piloted internally to determine any overlaps in the questions, relevancy to the desired topics, consistency in phrasing, and clarity of explanation. This was achieved by conducting a face-to-face interview, using the questionnaire with two participants who were external to the research, and matched the established selection criteria. The pilot study participants had the added advantage of being someone who had significant experience in interviewing Web developers in their PhD research and being interviewed as a Web developer. We found upon analysing pilot-test interview data that a couple of the questions were interpreted in different ways by pilot-test subjects and thus provided us the opportunity to improve the questions accordingly (for example Question 5, see Appendix C). Based on the results received from the pilot study and the feedbacks from supervisors, the questionnaire was refined and formally documented.

Following the above four steps, researchers conducted all interviews face-to-face and probes were used to get participants to expand and clarify their responses. Additionally, when answering questions posed during the interviews, the participants were asked to respond based on their experience in Web systems development, not just limiting them to a single more recent Web project. At the beginning of the interview, descriptions about early identification of change impacts, information architecture and other technical aspect were explained to participants.

4.5 Data Analysis

This section briefly described the analysis of the interview data (a detailed description has been covered in Section 3.7) in accordance with the research questions presented in Table 4.1. We have structured the presentation of data analysis in the subsequent paragraphs sequentially based on question numbers.

Regarding the industrial practices of CIA in Web systems (Research Question RQ6), participants were asked several questions including the goal(s) of analysing change impacts (Q1), the way in which changes in business processes affect information architecture design (Q2) and its significance (Q3), type of change impacts on IA (Q4) and their significance (Q5), and the approach used to identify change impacts (Q6). Most of the participants indicated that CIA supports the reduction of unnecessary re-work and few participants stated that CIA supports decision making and change estimation during Web systems development. The participant in general stated that changes in business processes substantially affect different abstractions of IA (including structure and behaviour of information) and considered to adequately address those affect very significant both for user satisfaction and ultimate success of Web-based business. Mostly, the participants described that there was no formal approach adopted, but a combination of ad hoc approach, previous experience, gut feeling and intuition were used to perform CIA activity.

To understand what information and practices currently used for the identification of change impacts (RQ8), the participants were asked questions such as: is there any approach that help Web developers to identify impacts on architecture design (such as IA) resulting from business processes changes (Q7), need for an approach that assists them in identifying impacts on IA (Q8), when impacts on architecture design are identified (Q11) and when does architecture design change frequently (Q12). All the participants indicated a lack of any approach to identify impact on architecture design resulting from business processes changes and considered it to be of the utmost importance to have such an approach to support CIA. The participants also informed us that impacts on architecture design were overlooked and mostly identified once the detailed design and implementation had begun. Participants also described that architecture design tends to change frequently, as soon as business processes were modified.

Regarding the kind of information used to perform CIA, the participants were asked to explain the information they used to identify impacts on architecture design (Q9) and what other information they believed needed to be employed (Q10). Most participants described the use of limited traceability/dependency information at the abstract level of artefacts. Additionally, change request information and change historical data were mainly used for the purpose of CIA. However, it was repeatedly stated by the participants that there was inadequate design information employed for the identification of impacts on architecture design. Additionally, architecture design decisions and previous change impact analysis results were also considered as essential information to support impacts identification on architecture design.

All interviews were transcribed and every part of the interview text was then analysed both quantitatively and qualitatively. For close-ended questions where options were provided, participants could select more than one option as there may be more than one option that is relevant. The responses from participants were tabulated in a spread sheet and analysed with basic statistical methods such as frequency analysis. For the responses received from open-ended questions, the data was analysed qualitatively to identify the common themes and to compare for similarity and differences in the participants' responses. During the analysis of the interview data, where the responses provided richer information about the questions being asked, quotes from the participants were also included.

During the analysis of the interview data, important insights were identified when each response for questions was coded to identify phrases and words. The words and phrases are mapped back to the interview questions to ensure a consistent and logical relationship between the responses and the questions. The patterns from within the interview data were then identified are referred to as themes.

The approach of coding adopted for data analysis involves close reading of the text. While deeply reading the interviewees' text data, we articulated that there were themes that emerged when we addressed the question, what was this sentence or expression an example of? As discussed by Ryan and Bernard (2003), themes are the classification of more discrete concepts and we found that some themes were broad and comprehensive constructs that were linked to many different kinds of expressions from the participants' responses. Additionally, a few themes emerged that were more focused and linked to very specific kind of expressions. As described in Section 3.7.2, from the analysis of the interview data we have identified a number of themes that supported us to derive a set of features that CIA approaches should address.

4.6 Analysis Results

This section describes data analysis results from the research activities conducted in Stage 3. From the analysis of the interview's data, several useful insights were obtained and discussed below.

The data analysis results of demographic questions indicate that 60% of our participants were from Australia and 40% from Asia. The participants' experience in the Web systems development varies between three years and five years with a median of four years. A median score for working as a Web developer, designer or architect is three years and a median score for working with one organisation (current or previous) is three years. An average number of co-workers on the current (or last) Web systems project are eight peoples. 95% of the respondents have IT related tertiary qualification. These demographics give us confidence that we have gathered data from participants who are experienced in Web systems development, especially in identifying impacts on architecture design.

It has been identified from the interview data that overall change impact analysis activity was unstructured and unclear, and specific impact analysis tasks, namely change identification, tracing the location of impacts and identifying their ripple effect were not well supported by the adopted CIA approaches. Subsequently, a list of features was derived from the interview data and the descriptions of those features are presented in Table 4.3.

4.6.1 Features for CIA

The thematic analysis of interview data reveals a list of fifteen features to be addressed by CIA approaches in Web systems. A list of these features along with their descriptions is presented in the following table.

Table 4.3: Features for CIA approaches derived from the interview data

Feature	Descriptions
1	Link business processes with their information architecture (IA) design entities to keep the traceability consistent between them
2	Identify how changes made at one abstraction of architecture design affect design entities on other abstraction of architecture design
3	CIA activity should be better guided and instructed (guidelines)
4	Activity descriptions (steps) required to carry out CIA
5	Analyse impacts early (before detailed design) to reduce un-necessary rework
6	Visualization of IA inter-level (e.g. structure to behaviour) and intra-level (e.g. structure to structure) dependencies among design entities of architecture design
7	Employ repository of change information and old impact analysis results as a knowledge base

8	Support feedback mechanism used to evaluate the results of impact analysis (impact set) after retesting affected software
9	A consistent and systematic way to identify impacts on architecture design
10	Avoid overestimation (unnecessary change impacts) and underestimation (overlooking change impacts)
11	Mechanism to extract information related to the dependencies among design entities of architecture
12	Identify how changes made in architecture design affect other parts of architecture design within a single abstraction (structure to structure, behaviour to behaviour)
13	Employ architecture design decisions repository/information
14	Identify which design entities are necessary to modify after making a change
15	Identify change impacts on IA resulting from business processes changes

4.6.2 Frequency distribution analysis of features for CIA

Frequency distribution facilitates to organise and summarise data in a form that allows further interpretation and analysis (Cohen and Holliday, 1996). Frequency distribution shows how the data clusters around a central value and the degree of difference between identified data (Cohen and Holliday, 1996). Given the importance of frequency distribution, we have used this approach to prioritise CIA features that are derived from the interview data. The frequency of occurrence for each feature was examined and we found that several features can be categorised or grouped together. In that case we moved them together and put them under a major feature - the kinds or types of which they all can be represented. We have prioritised features on the scale of 5-1 (5= highest, 1= lowest) as shown in Figure 4.1. Based on the frequency of the features occurrence, the list of identified features from Table 4.3 was reduced from fifteen to ten features. We have called these ten features as high-level features for CIA and it represents a more comprehensive set of features that CIA approaches should address. We tagged each feature with prefix FA for all ten high-level features (FA1-FA10) as shown in Table 4.4. The enumeration of features was made only for the purpose of referencing them later. The numbering logic is used to structure features and their sub-features. For example, feature FA-2 has sub-features as FA-21 and FA-22, etc.

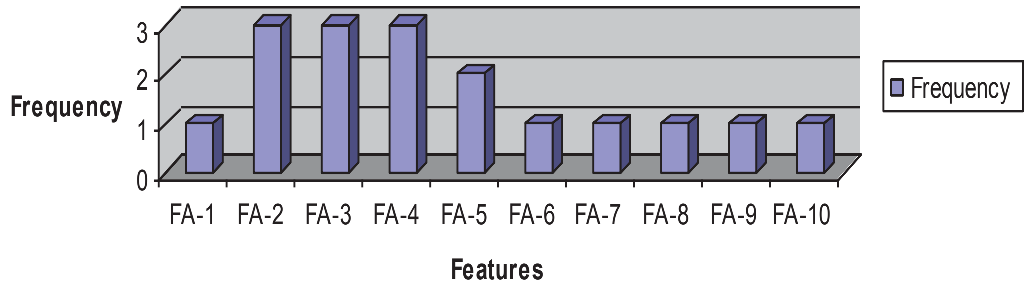


Figure 4.1: High-level features based on the frequency distribution of fifteen identified features

It can be observed from Table 4.4 that high-priority features are concerned with the impacts identification and the execution of the analysis task respectively, which arguably are the fundamental aspects of a process level approach towards CIA. Whereas, medium-priority features are concerned with activity-orientation, consistency and the management of CIA activities. These results guide us toward the process level support for CIA in order to facilitate change impact analysis in Web systems. A detailed discussion on process level support will be covered in Chapter 5.

Table 4.4: High-level Features of Change Impact Analysis

Features for Change Impact Analysis	
FA-1 link business processes and information architecture (IA) design entities to keep the traceability consistent between them	
FA-2 Impact analysis at IA level facilitates to:	
	FA-21 Identify how changes made at one abstraction of architecture design affect design entities on other abstraction of architecture design (structure, behaviour, location, etc)
	FA-22 Identify how changes made in architecture design affect another part of architecture design within a single abstraction (structure to structure, behaviour to behaviour, etc)
	FA-23 Identify which design entities are necessary to modify after making a change
FA-3 Process development support for:	
	FA-31 Activity descriptions (steps)
	FA-32 Each activity (from FA-31) should be better guided and instructed (guidelines)

FA-4 Process working support for:	
	FA-41 Managing multi-abstract architecture changes - changes at different abstractions of architecture design
	FA-42 Mechanism to extract information related to the dependencies among design entities of architecture (IA)
	FA-43 Visualization of inter-level (e.g. structure to behaviour) and intra-level (e.g. structure to structure) dependencies among design entities of architecture
FA-5 Process management support for:	
	FA-51 A consistent and systematic way to identify impacts on architecture design
	FA-52 Support feedback mechanism used to evaluate the results of impact analysis (impact set) after retesting affected software
FA-6 Avoid overestimation (unnecessary change impacts) and underestimation (overlooking change impacts)	
FA-7 Employ a repository of change information and old impact analysis results as a knowledge base	
FA-8 Employ architecture design decisions repository/information	
FA-9 Analyse impacts adequately on architecture design before detailed design to reduce un-necessary rework	
FA-10 Identify change impacts on IA resulting from Business processes changes	

As discussed before, the analysis of the interview data revealed a set of features, and finally ten high-level features were developed by the re-organisation of fifteen identified features. Additionally, findings from the review of scholarly research literature also highlighted the inadequacy of existing CIA approaches while addressing aspect including (i) linkage between business processes and architecture design of Web systems, (ii) identification of impacts on architecture design resulting from business processes changes, (iii) a process level support for CIA, and (iv) employing both traceability and dependency analysis (v) to address the problem where the implementation (detailed design) actually begins before change impacts are adequately identified in Web systems (see Section 2.8). Interestingly, these five aspects were already among the ten high-level feature of CIA, which confirms that our listed ten high-level features were indeed relevant and reflects a reasonable coverage of features that CIA approaches should address. However, these ten features vary in terms of closeness to impact analysis. For example some features are concerned with change requests, some are concerned with actual CIA activity, some concerned with process detail and others with process level support, and some are concerned while tracing the location of impacts.

Overall, deficiencies in formalising and employing a consistent, structured and step-by-step approach of change impact analysis were identified during all interviews. Since CIA approaches were not formalised, no tool support was available at the time of studies. Therefore, instead of tool support, a major focus of features was on the basic process functions such as activity description and checklist support (FA-31), systematic and consistent approach (FA-51) information reuse (FA-7, FA-08), process guideline (FA-32) and validation of process results (FA-52).

4.7 Classifications of Needs for Change Impact Analysis

From the findings of the literature review and interview study, we have identified a set of high-level features to be addressed by CIA approaches in Web systems as shown in Table 4.4. For a more focused research direction, we have classified high-level features into five specific focused areas, and these are presented in Table 4.5. The purpose of the classification of high-level features is to provide a framework for analysing the group of high-level features together in relation to change impact analysis.

- Traceability & dependency analysis
- Process level support
- Effectiveness of analysis
- Design information reuse
- Understanding specific Web systems characteristics

Table 4.5: Classification of high-level features of CIA

Classified Focused Areas	**Features for Change Impact Analysis**
Traceability & dependency	FA-1 link business processes and information architecture (IA) design entities to keep the traceability consistent between them

analysis	FA-2 Impact analysis at IA level facilitates to:	
		FA-21 Identify how changes made at one abstraction of architecture design affects design entities on other abstraction of architecture design (structure, behaviour, location, etc.)
		FA-22 Identify how changes made in architecture design affects another part of architecture design within a single abstraction (structure to structure, behaviour to behaviour, etc.)
		FA-23 Identify which design entities are necessary to modify after making a change
Process level support	FA-3 Process development support for:	
		FA-31 Activity descriptions (steps)
		FA-32 Each activity (from FA-31) should be better guided and instructed (guidelines)
	FA-4 Process working support for:	
		FA-41 Managing multi-abstract architecture changes - changes at different abstractions of architecture design
		FA-42 Mechanism to extract information related to the dependencies among design entities of architecture (IA)
		FA-43 Visualization of inter-level (e.g. structure to behaviour) and intra-level (e.g. structure to structure) dependencies among design entities of architecture design
	FA-5 Process management support for:	
		FA-51 A consistent and systematic way to identify impacts on architecture design
		FA-52 Support feedback mechanism used to evaluate the results of impact analysis (impact set) after retesting affected software
Effectiveness of Analysis	FA-6 Avoid overestimation (unnecessary change impacts) and underestimation (overlooking change impacts)	
Design information	FA-7 Employ a repository of change information and old impact analysis Results as a knowledge base	

reuse	FA-8 Employ architecture design decisions repository/information
Understanding specific Web systems characteristics	FA-9 Analyse impacts early (before detailed design) to reduce unnecessary rework
	FA-10 Identify change impacts on IA resulting from Business processes changes

This classification will be used in Chapter 5 while taking into consideration the focused areas during the development of the process model of CIA. The following sections explain each focused area in detail and link the related features to each of them.

4.7.1 Traceability and dependency analysis

Traceability is related to establishing and maintaining traceability links between software system artefacts (Bohner and Arnold, 1996). The necessary discussion on traceability and dependency analysis has been covered in sufficient detail in Section 2.7, Chapter 2. Traceability from business processes to architecture design entities (FA-1) was regarded as valuable during all interviews. Furthermore, the dependency links were also considered important for maintaining consistency among different abstractions of IA (FA-21) by all the interviewees. Additionally, dependency information among design entities were also regarded as valuable (FA-42) for the purpose of CIA.

Indeed, identifying change impacts on architecture design resulting from business processes changes (FA-10), managing change impact in a multi-abstract architecture design (FA-41), analysing the ripple effect of a change to other architecture design entities (FA-23) and achieving an understanding of change impacts in Web systems (FA-10) were among the most problematic tasks identified by the participants directly and indirectly related to a traceability and dependency analysis.

During interviews participants repeatedly stated that business processes changes were frequent in Web systems and led to the situations where the traceability of those changes down to the level of architecture design were difficult to study. It was also observed from the interview data that the dependency analysis is difficult to perform due to unavailability of adequate dependency information among design entities. It has been articulated from the interview data that there are three important levels of support required from a traceability and dependency analysis.

1. To adequately manage traceability links between business processes and architecture design entities. Given that the creation and maintenance of traceability links through experience or based on gut feeling was regarded as unreliable, the importance of traceability links between business processes and architecture design entities has also been presented as a desired feature of CIA approach (FA-1).
2. To find out the locations of change impacts. This aspect deals with the identification of change impacts within and across different abstractions of architecture design and are presented as a desired feature of CIA approach (FA-21, FA-22).
3. To examine the ripple effect. This aspect emphasises the effects caused by a change made to architecture design. It typically forms a narrow perspective of the architecture analysis so as to identify which design entities are necessary to modify after making a change (FA-23).

One of the characteristics of Web systems is reported as business processes being interwoven with the architecture design of Web systems. Given the importance of this characteristic and it roles in CIA (see Section 2.3), the traceability links from business processes to architecture design entities is highly desirable (FA-1) for adequate identification of change impacts on architecture design (FA-1). Further, the task of identifying change impacts at multi-abstract architecture design (FA-41) may become complex largely due to tight-connection between business processes and architecture design of Web systems - another characteristic of Web systems.

4.7.2 Process level support

One of the main findings from the interview data is that commonly agreed approaches for change impact analysis did not typically exist. Impact analysis activity and its execution were left to individuals to perform and were mostly carried out in an ad hoc way. During architecture design modification, change impacts were not adequately analysed, processed and managed by Web designers/architects. However, when Web systems were being implemented or deployed in the production environment, the person responsible for system maintenance managed the change impacts as he saw fit or followed an informal approach.

The need to develop a process approach for CIA in Web systems is highly desirable, although a few features also emerged such as the process should support a set of guidelines and be lean and not fixed. The activity of CIA was described as including "ad-hoc", "haphazard" and "unsystematic". For example, a participant stated:
…some time we adopted CIA approaches according to the situation, we sensed the situation and used our own expert judgment, no explicit rules were there as such and a lot of intuition used.

The interview data also revealed that most of the adopted CIA approaches were typically represented by few activities to perform. Mostly these activities consisted of high level tasks such as identifying and reviewing change request, examining associated document (if any), consulting and discussing changes with stakeholders for the identification of possible impacts, and implementing changes according to identified change impacts. Therefore, developing a process approach is important so as to ensure that all the activities are completed and executed with consistency. As a participant described:
…it is important to find a way that enables best practices to be encouraged and adopted, thereby building on the current state of practical knowledge.

The paucity of a formalised CIA approach was identified from all the interview data and can be considered as a process level support (FA-3, FA-4, FA-5) for CIA. However, in a few organisations official procedures were there for reporting issues while validating a particular change, but these procedures were not developed to address change impacts at the architecture design level. Instead these procedures were more focused toward issues/bugs identification resulting from the changes made at the code level. Deficiencies such as ad hoc, haphazard and unsystematic ways of CIA activity imply that a consistent and systematic approach is required (FA-51) for CIA.

Surprisingly, approaches reported in the research literature for CIA (Berg, 2006, Tang et al., 2007, Zhao et al., 2002) were not part of the Web systems development in most of the interviewees' studied. Consequently, the Web designers/architects adopted their own approaches. Additionally, since the architecture of CIA approaches were not formalised, no tool support was available at the time of study. Therefore, the requirements for tool support also primarily focused on the need for basic process functions including activity description (FA-31), process guidelines (FA-32), information extraction and representation (FA-42, FA-43) and process and its validation (FA-51, FA-52).

In the CIA approach proposed by Bohner and Arnold (1996) they have mentioned the use of change history data, which is a way to learn from past experiences, but they have said nothing explicitly about the importance of evaluating the impact analysis activity. From the interview data, we have derived the feature of process validation and posit this feature as an improvement of Bohner's CIA approach by adding the evaluation of the result of impact analysis (impact set) after retesting the affected system (FA-52).

4.7.3 Effectiveness of analysis

Effectiveness can be defined as an ability to achieve the desired results. During CIA it is mostly desirable to perform analysis in a way that all relevant change impacts can be identified within the constraints which are particle and reasonable from the system development's point of view (Bohner and Arnold, 1996). Additionally, effectiveness also means maintaining a balance between neither over nor underestimation of change impacts (Kagdi and Maletic, 2006, Lindvall, 1997). It means that effectiveness refers not only to identifying all relevant change impacts, but also avoiding overestimation or underestimation of impacts related to a change. Overestimated or underestimated impacts, indeed, result in false impacts. Consequently, these false impacts cause redundant work in the case of overestimation and at worst an unexpected behaviour of changed system in the case of underestimation.

It has been observed from the interview data that most typical improvements related to the effectiveness of identified impacts required are in terms of reducing the likelihood of false impacts. Effectiveness of analysis facilitates to reduce the processing overheads resulting from false impacts and misleading modification based on false impacts. For example, one of the participants during the interview described:

…mostly we underestimate the impacts and therefore underestimating the consequence of making a change can directly effect on cost/effort estimation and further on release planning.

Another participant stated:

…overestimations are mostly the result of providing a wide coverage of change propagation. Therefore, redundant work is done to process the false impact entities that are not actually affected when the changes are being made.

The above statements also suggest that CIA approaches adopted in Web systems context must employ mechanisms for an effective analysis in order to reduce unwanted impacts and thus avoid expensive and complex modifications afterwards. Therefore, it is important to address those impacts that may not be real impacts, but rather misunderstood impacts related to a change.

Specific characteristics of Web systems that were described in Chapter 2 can play an important role in the effectiveness of analysis in two ways. Given that Web systems business processes are interwoven with architecture design; consequently, potential changes in business processes lead to fundamental changes in architecture design and subsequently both business processes and architecture design co-evolve and emerge together (see Section 2.3). Additionally changes in one architecture portion (and thus to underlying design entities) may lead to modifications in another portion of an abstraction and across different abstractions of architecture design. At this stage, it was observed that there was no effective way to support identification of impacts on architecture design resulting from business processes changes. As a result, participants mostly followed an ad hoc approach or delayed impact identification until later stages of system development. As a result, most of the identified impacts are not real impacts and thus contribute to either underestimation or overestimation of change impacts.

Additionally, interview data revealed that there was both a lack of availability of dependency information to carry out the impact analysis (FA-42) and any structured and consistent approach is not followed (FA-51). Both the lack of availability of dependency information and failure to follow any structured and consistent approach results in identification of those false impacts and consequently leads to un-necessary re-work. The effectiveness of analysis for the architecturally sophisticated Web systems was usually considered high (Lowe and Henderson-Sellers, 2003); therefore, the unnecessary re-work at later stages of Web systems development should be minimised. The above discussions guide us toward the effectiveness of analysis (FA-6) to avoid both the underestimation and overestimation of change impacts.

4.7.4 Design information reuse

Design decisions information is considered as very important architecture knowledge (Tyree and Akerman, 2005, Bosch, 2004). Design decisions information is mostly stored in a form of repository which consists of tacit and referenced information about design decisions taken and the architectural solution that implement those decisions (Avgeriou et al., 2007). Design decisions information can also play a crucial role while identifying change propagation on the architecture level and exploring the dependencies between architecture design entities (Tyree and Akerman, 2005, Bosch, 2004).

During Web systems evolution, at the stage when business processes and information architecture solutions begin to evolve, one of the participants described:

…as business processes get changed, relatively extensive design decisions are taken to address these changing requirements. Due to less-focused utilisation of these design decisions information during CIA, most of CIA approaches do not adequately support impact identification at information architecture level.

Identification of change impacts may become more complex while addressing other characteristics of Web systems such as tight-connection between business processes and architecture design. As a result of tight-connection, changes in business processes often lead to fundamental changes at architecture design (Lowe and Eklund, 2001). Additionally, the nature of Web systems architecture is like any change in business processes that may impact on different abstractions of architecture design (see Section 2.3, Chapter 2). The above discussion implies that the identification of change impacts on architecture design cannot be performed solely based on change requests specifications. Existing architectural CIA approaches mainly focus on changes to the architectural component and connectors (Zhao et al., 2002, Tang et al., 2007), but fail to employ design decision information (FA-8) which is considered important from the analysis of interview data.

4.7.5 Understanding specific Web systems characteristics

Interview data revealed that problems Web designers/architects faced during CIA primarily stemmed from inadequate understanding of Web systems characteristics. Two main characteristics contributing difficulties during CIA were identified as:

- It has been reported that business processes are interwoven with architecture design of Web systems. However, there is a lack of understanding of this characteristic partially due to the fact that the traceability from business processes to IA and its design entities is not explicitly well-established. This leads to the problem of inadequate analysis of change impacts and failure to identify impacts on architecture design resulting from business processes changes.
- There is a tight-connection between business processes and architecture design of Web systems. Consequently, a potential change in business processes may typically lead to fundamental changes in architecture design developed to support those business processes. A lack of understanding of this characteristic may complicate the impact analysis activity further, as Web developers/architects have to track several combinations of design entities in order to analyse the effect of changes at architecture design. This leads to the problem of overlooking impacts on architecture design resulting from business processes changes. Indeed, most of these overlooked impacts were identified at later stages of Web systems development.

In support of tight-connection between business processes and architecture design of Web systems, one of the participants expressed:

… from our experience we know that nature of (Web systems) architecture is like that- any change in business processes may effect on information architecture, however this change in business processes need to be directly reflected at architecture level…for example a change in business work flow need to be reflected in terms of information flow and navigational flow.

Both of the above Web systems characteristics are important to consider in relation to CIA as discussed in Section 2.3. Adequate understanding of these Web systems characteristics facilitates identifying impacts on architecture design resulting from business processes changes (FA-10) - early identification of change impacts in Web systems. Additionally, an understanding of Web systems architecture can provide an adequate visualisation of dependencies between architecture design entities (FA-43) and support to manage a multi-abstract architecture design (FA-41). Overall, we have concluded from the interview data that the specific Web systems characteristics need to be recognised in order to increase our understanding for the purpose of CIA.

4.8 Chapter Summary

Before conducting face-to-face interviews, it was postulated that there is a need for possible extension of CIA approaches to support early identification of change impacts in Web systems. The analysis of the interview data revealed that, indeed, existing CIA approaches that can be adopted in the Web systems context are inadequate and there is a need for an extension. Additionally, analysis of the interview data revealed a list of high-level features that CIA approaches should address. Commonly agreed change impact analysis approaches to support early identification of change impacts in Web systems do not explicitly exist. In a few companies, official procedures exist for reporting issues while managing changes at the detailed design or coding level, but do not provide much support for architecture design. Mostly, architectural level impact analysis were left for the execution of the person responsible for the system maintenance and he managed the change impacts as he saw fit and therefore it is reported as ad hoc. Indeed, the identification of impacts on architecture resulting from business processes changes were not adequately analysed, processed or managed by the Web designers/architects.

Overall CIA approaches for early identification of impacts in Web systems were unstructured, inconsistent and unclear. Furthermore, a systematic approach for impact analysis is largely not part of Web system development, so Web developers/architects tend to develop their own approaches. However, this is not unexpected given that most of the practitioners depend primarily on their experience, expert judgment, and intuition during change impact analysis. Key sentences from the previous section are worth repeating here, given its significance. Since the CIA approaches were not formalised, no tool support was available at the time of the studies. Therefore, instead of tool support, major focus was on the basic process functions such as CIA activities descriptions and checklist support, traceability and dependency information tracking, information reuse, process guideline, and process validation.

The current state of practice is generally not encouraging, and would imply that significant attention and effort is still needed in this area. This was not unexpected given the gaps identified from the review of related literature and work reported for the relevance of existing CIA approaches in a Web system context (Mehboob et al., 2009). In fact, a number of features for CIA approaches were derived from the interview data and may suggest that there is in reality a strong demand for them in practice today. Participants' support on the need of new CIA approaches versus improvement of the existing ones was split, although all participants during interviews directly and indirectly acknowledged the need for the extension of CIA approaches for Web systems. However, it was notable that a structured, consistent and rigorous approach was highly desirable, one that intuitively guides us toward a process based approach for CIA.

Classification of high-level features identified the focused areas of research. In summary, the main points identified in this chapter are as follows:

1. The need for a possible extension of CIA approaches, specifically, to address the characteristics and nature of change impact in Web systems are unanimous, and there is a congregative consensus on how this can be achieved.
2. High-priority features are concerned with impacts identification and the execution of the analysis tasks respectively, which arguably are fundamental aspects of change impact analysis.
3. Information to trace changing business processes down to the level of IA design entities, and design decisions information to exploit dependencies among design entities are considered essential respectively for traceability and dependency analysis.

Given the paucity of research focus on handling the characteristics of Web systems related to CIA, nature of change impacts in Web systems, and based on the findings from industrial practices of existing CIA approaches, we will develop a process model of CIA. Hence, the next stage of the research, Stage 3, is directed toward the development of a process model of CIA, as an extension of existing CIA approaches, based on the comprehensive list of high-level features for CIA.

Chapter 5: A Process Model of Change Impact Analysis

5.1 Introduction

In Stage 1 of this research, we investigated two important characteristics of Web systems in relation to change impact analysis, as reported in Chapter 2. An adequate consideration of Web systems characteristics guides us towards the identification of impacts on architecture design resulting from business processes changes- early identification of change impacts in Web systems. In Stage 2 of this research, we investigated current industrial practices of CIA in Web systems. The investigation results from Stage 2 reveals a set of features that CIA approaches should possess, as discussed in Chapter 4. In Stage 3 of our research, we propose a CIA approach in response to the findings both from Stage 1 and 2. The proposed approach supports for early identification of change impacts in Web systems and is referred to as the process model of CIA (PMCIA) in this thesis. In essence, the main contribution of this research study is to propose an extension of current CIA approaches specifically to support impact identification on architecture design resulting from business processes changes that in turn addresses the resulting focus of Web systems characteristics.

An analysis of investigation results from Stage 1, firstly direct us to the identification of impacts on architecture design resulting from business process changes (we called it as early identification of impacts in Web systems). Secondly, results from Stage 1 also encourage us to employ design decision information during CIA. PMCIA is developed in response to serve the direction provided by the investigation of results from Stage 1. Furthermore, an analysis of investigation results from Stage 2 directs us towards a set of features that the CIA approach should possess. These features mainly include step-by-step, structured and a rigorous approach, re-use of design knowledge and early identification of change impacts in Web systems. Subsequently, PMCIA is developed in response to the direction provided from the Stage 2 investigation results.

This chapter describes how PMCIA supports early identification of change impacts in Web systems, steps of PMCIA, activities performed during each step, guidelines to execute those activities, execution flow, and inputs/outputs for PMCIA. To depict the data input requirements of PMCIA, this chapter also describes a supporting data model. Furthermore, in this chapter we also provide a mapping between PMCIA and the features identified from the investigation of Stage 2. Essentially, this mapping supports to understand how PMCIA is developed in response to these features.

This chapter begins by describing, in Section 5.2 extensions of existing CIA approaches. In Section 5.3, we describe the use of design rules and constraints in PMCIA. Further, in Section 5.4 and 5.5, we describe the details of data model and PMCIA respectively. We then discuss how PMCIA is developed in response to CIA features identified from Stage 2 in Section 5.6. Additionally, in Section 5.7 we discuss how PMCIA is developed in response to the proposed extension (described in Section 5.2). We finish by providing a summary of this chapter in Section 5.8. Proof of demonstration of PMCIA through an exemplar is presented in Appendix E.

5.2 Extensions of Existing CIA Approaches

The findings from the state of the art (Mehboob et al., 2009) indicates the paucity of existing CIA approaches to support early identification of impacts in Web systems. Additionally, the state of the practice is generally not encouraging while adopting those existing CIA approaches in Web systems and suggest a list of features that a CIA approach should possess (Mehboob and Zowghi, 2009). Importantly, these features include the structured and consistent execution of the analysis task, information re-use and the effectiveness of impacts identification. These features have also arguably been reported as fundamental aspects of a process approach (Bohner and Arnold, 1996, Jonsson, 2007). Both findings from the state of the art and the state of the practices indicate a possible extension of existing CIA approaches to suitably adopt in Web systems. We have selected Bohner's model of CIA (1996) for a possible extension and proposed a process model of CIA. There are a few reasons for selecting Bohner's model of CIA: (i) it provides a generic concept of CIA in software development (ii) it advocates the abstraction of possible steps and their interaction (iii) it provides the necessary foundation for the CIA approach (iv) it is the most refereed and seminal research work in the area of CIA. Following are the list of extensions incorporated in PMCIA.

1. Impacts identification on the architecture design resulting from business processes changes - early identification of change impacts in Web systems.
2. Step-by-step and structured approach for a rigorous and consistent execution of analysis task.
3. Systematically and carefully incorporating the notion of design decisions information (information re-use) to support adequate identification of change impacts.

The above listed points suggest that current CIA approaches can be extended specifically by focusing on the identification of impacts on the architecture design resulting from business processes changes, systematically and rigorously executing CIA activity, and employing design decision information (such as design rules and constraints) to facilitate change impact analysis in Web systems. In Chapter 2, Section 2.8, we have already covered in detail, design decisions and the importance of design rules and constraints in relation to CIA. In the next section, we briefly cover the use of design rules and design constraints in PMCIA.

5.3 Using Design Rules and Constraints for PMCIA

For change impact analysis it is often important to trace the relationships between system entities. Tracing these relationships may allow us to identify possible dependencies (and hence change impacts) among system entities. In architecture design, design rules and constraints inherently bind multiple design entities together, and hence implicitly capture relationships between those design entities. This indicates that an analysis of these rules and constraints can lead to the identification of possible dependencies between design entities.

The discussion made in Section 2.8 indicates that design rules and constraints bind a combination of design entities together. In essence, the design entities (that underpin those design rules and constraints) are interacting with each other (Kruchten et al., 2006, Sangal, 2007), however, often there are implicit relationships among those design entities (Mattsson et al., 2008, Tyree and Akerman, 2005). A possible mechanism to link one design entity to another design entity may allow us to identify those implicit relationships, and hence provides support for the identification of possible dependencies among design entities.

Investigation of the design decisions repository informed us that design rules and constraints are commonly documented in a natural language text form. Given the textual description of design rules and constraints, the relationships between design entities are mostly reported as implicit (Tyree and Akerman, 2005) as those relationships are mostly inherited within the textual description of design rules and constraints (Mattsson et al., 2008). In PMCIA, the same textual form of design rules and constraints are used, however, we resolve the issue of implicit relationships among design entities by (forward) linking one design entity to another design entity (see Section 5.5). The purpose of forward linking is to examine the implicit relationships among design entities, to represent explicitly those relationships by directed links, and hence to give a structure to those relationships (in the form of a graphical representation). In order to achieve this purpose, after selecting relationships among design entities, we represent those design entity relationships in a structure form through graphical representation (see Figure 2). In doing so, we perform content analysis on the textual form of design rules/constraints and pick all the relationships among design entities that underpin a design rules/constraints. A graphical representation of design entity relationships, and further its analysis allow us to identify possible dependencies (and hence change impacts) between design entities. Therefore, we believe that the consideration of design rules and constraints can be important for the identification of possible dependencies among design entities.

Consideration of design rules and design constraints during PMCIA (specifically employing them for linking one design entity to another design entity) guides us toward important information constructs including design entity relationships, dependency relationships, design entities, and design decisions themselves. These information constructs are further used as data input during PMCIA (discussed in Section 5.3.1 and 5.3.3). In the following section we will describe a data model that represents necessary information construct to support the execution of PMCIA and serve as a partial data input requirements for the process model.

5.4 Data Model

As described by Simsion & Witt (2005), the purpose of a data model is to provide support for communication with the process model. Keeping this view in mind, we developed a data model, describe its various elements and explain different connections, to better depict the data input requirements of the process model. We have followed a conceptual level of data modelling that consists of general concepts about the information content to be used in the process model (Simsion and Witt, 2005). Subsequently, data model portrays data objects, constraints on these objects, their relationships and serves as an excellent tool for communication with the process users. In the data model, shown in Figure 5.1, we have specifically focused on design decisions and their related information. Given the previously discussed relevancy of design decisions during CIA (Davide et al., 2008a, Davide et al., 2008b), we have taken into account information such as design rules and constraints.

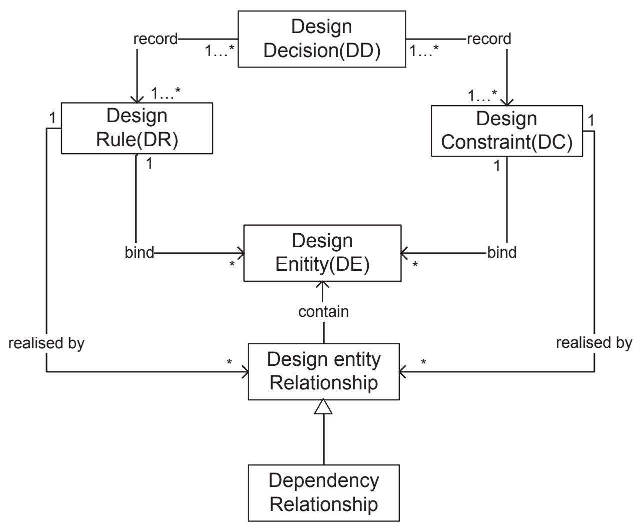


Figure 5.1: Data model to support data input requirements of PMCIA

We have developed the data model by using class diagram notations (Pilone and Pitman, 2005) as shown in Figure 5.1. The main information categories in the data model are design decisions (DD), design entities (DE), design rules (DR) and constraints (DC), deign entity relationships and dependency relationships. This data model is different from other previous works (Babar et al., 2006, Capilla et al., 2006, Jansen and Bosch, 2005) in two ways. Firstly, our proposed data model focuses on design decisions and their relevant information such as design rules and constraints.

Secondly, our data model employs design rules and constraints to realise design entity relationships and to further present possible dependencies among design entities.

The data model has similarities, such as the representation of design decisions and design entities to other related data models (Babar et al., 2006, Capilla et al., 2006, Jansen and Bosch, 2005). However in comparison with other data models, our data model is different because it augments design entity relationships that stem from design rules/constraints. Additionally, our data model also represents the possible dependencies among design entities resulting from design entity relationships. Subsequently, our data model also strengthens the linkage between design entities, design rules/constraints and design entity relationships. To better understand the rationale for our data model, it will be important here to describe the sources, appropriateness and the form taken by each data model elements. To address this aspect, we have developed Table 5.1, where each column describes the sources, appropriateness and the form taken by each data model element.

Table 5.1: Description of sources, appropriateness and the form taken by each element of the data model from Figure 1

Data Model Element	**Sources**	**Appropriateness**	**Form Taken by Data Model Element**
Design decision	(Jansen and Bosch, 2005, Jansen et al., 2007, Kruchten et al., 2006, Tyree and Akerman, 2005)	Design decisions are the earliest design knowledge available from the architecture designing activity and reveal important design information such as design rules and constraints to be employed for change impact analysis.	Design decisions are identified by the design decision-IDs such as (DD1,DD2…DDn) and retrieved by a search query from the design decisions repositories.
Design entity	(Perry and Wolf, 1992, Shaw, 1993)	Design entities are the main objects of	Design entities are the components of design

		interest while identifying the impacts of change on the architecture design.	architecture and mostly represented by Ids such as (DE1, DE2,…DEn).
Design rule	(Kruchten et al., 2006, Mattsson et al., 2008, Sangal, 2007, Tyree and Akerman, 2005, Wan-Kadir and Loucopoulos, 2004)	Design rules impose certain behaviours and influence different design entities underlying that design rule.	A design rule documented simply in a textual form and recorded with design decisions descriptions.
Design constraint	(Albin, 2003, Babar and Gorton, 2007, Chevaliera and Ivory, 2003, Jansen et al., 2008, Kruchten et al., 2006, Mattsson et al., 2008, Sangal, 2007, Shaw, 1993, Tyree and Akerman, 2005)	Design constraints limit certain behaviours and influence different design entities underlying that design constraint.	A design constraint is documented simply in a textual form and recorded with design decisions descriptions.
Design entity relationship	(Jansen et al., 2008, Kruchten et al., 2006, Mattsson et al., 2008, Sangal, 2007)	Design entity relationships are explored in order to make the implicit influences between design entities explicit. And to use these relationships for the purpose of	Design entity relationships are realised by underlying design rules and constraints. These relationships are represented by direct arrows between two design entities.

		dependency analysis.	
Dependency relationship	We have explored dependency relationships among design entities.	Dependency relationships are the dependencies between design entities and they are revealed from design entity relationships.	Dependency relationships is a kind of design entity-relationship and are identified whilst back-tracking design entity relationships. Dependency relationships are represented by direct solid arrows between the two design entities in a dependency graph.

In proposing this data model, comprehensiveness was considered an important aspect of the data model. Loshin (2001) has described the comprehensiveness as how well a data model encompasses enough information to accommodate the current needs of its use. In our approach, the important aspect is how well the data model encompasses enough information to address the data input requirements for the process model. In particular, how well the data model encompasses enough information to reliably support the identification of the dependency relationships (for the dependency graph in Section 5.5). While developing the data model, we ensured the data model and process model are easily comprehensible, meaning that the data model elements are modelled such that they can be understood correctly in a reasonable amount of time (Lange and Chaudron, 2005). Subsequently, we considered the data required during the forward linking of design entities including design decisions, design entities and deign entity relationships. Further, we considered the data required for backward tracing (thus for the dependency graph) including design entities and the dependency relationship, ensuring that the proposed data model is comprehensive so as to reliably trace the dependencies among design entities. In essence, the proposed data model encompasses adequate information to accommodate the data requirements for PMCIA.

Lange and Chaudron (2004) described the completeness of the data model as the system (in our case the process model) that is completely described by the model. Therefore, to focus on completeness, we have looked at the data model as a whole and ascertained that the representation of any part of data (as input requirements of process model) is not missing. Additionally, we have focused on the following two ways to ensure the completeness of data model as described by Reingruber and Gregory (1994).

Conceptual Completeness - Conceptual completeness implies that the data model contains adequate elements for describing the scope of the information domain that the model purports to represent (Reingruber and Gregory, 1994). In our data model, we have represented the data concepts and their relationships to express completely the information domain. Additionally, with each relationship, relationship cardinalities are also shown.

Syntactic Completeness - Syntactic completeness implies that the modelling process has been carried out completely and thoroughly to produce a data model. This refers to the data modelling process rather than the data model itself (Loshin, 2001). We have adopted the class diagram as the modelling technique where the necessary data concepts are captured at appropriate points in the modelling process.

5.4.1 Description of data model

In the previous section, Table 5.1 only provides a description of sources, appropriateness and the form taken by each element of the data model. However, in this section we will describe in detail each element of the data model.

Design Entity (DE) represents the design object that participates during architecture design. They are the basic components of the architecture design (IEEE, 2000, Perry and Wolf, 1992) and these components are mainly modified along with their connection to address a design change.

Design Decision (DD) focuses on a selected design solution describing the details of the decision being made (Babar and Gorton, 2007, Bosch, 2004, Capilla et al., 2006). Mostly design decisions are documented in template form and managed through design decision repositories. An important part of the design decision is design rules and constraints (Babar and Gorton, 2007, Mattsson et al., 2008, Tyree and Akerman, 2005).

Design Constraints are the limitations to what can be achieved. They may be of a technical, infrastructure or other nature. Further, design constraints are prescriptions for design decisions and describe what is not allowed during architecture design. i.e. they prohibit certain behaviours and limit the design to remain sound (Jansen and Bosch, 2005).

Design Rules are the way or the behaviour of one design entity that influences another design entity in different parts of the architecture by encouraging the available design options. Additionally, design rules define partial specifications to which the realisations of one or more design entities have to conform. Design rules define how the components of architecture design solution should be structured, organised and realized (Jansen et al., 2008) and these rules serve as mandatory guidelines (Jansen and Bosch, 2005).

Design Entity Relationship are the relationships among design entities resulting from the specific interaction among design entities that underpin a design rule and constraint. Therefore, these relationships can be considered as connectors linking one design entity to other design entities (Burge and Brown, 2006, Capilla et al., 2007, Jansen et al., 2008). We consider design entity relationships as an important source, further enabling exploration of possible dependencies among design entities.

Dependency Relationship is the dependency that a design entity can have on other design entities. In order to identify the possible dependencies between design entities (see Section 5.5), we have focused on the backward tracing of design entity relationships. Indeed these dependency relationships mainly support us to develop a dependency graph for the purpose of change impact analysis.

5.5 Process Model

Both investigation of the current research literature and industrial practices have highlighted a limited focus on the process of CIA (Jonsson and Lindvall, 2005, Mehboob and Zowghi, 2009). In particular, an investigation of industrial practices of CIA in Web systems indicates that CIA approaches are mostly informal, unstructured and ad hoc (Mehboob and Zowghi, 2009). Other important aspects derived from an industrial perspective are inadequate description of steps, guidelines, inputs and lack of mutually agreed approaches required to carry out the CIA in Web systems. Based on these investigation findings, we have proposed PMCIA (Mehboob et. al., 2011) as a possible extension of Bohner's CIA approaches. There are two important aspects of PMCIA including forward linking and backward tracing. It is important to understand these two concepts before introducing PMCIA in detail.

Forward linking for design entities relationships- Mostly design entity relationships are reported as implicit within design rules and constraints (Mattsson et al., 2008, Tyree and Akerman, 2005). The purpose of forward linking of design entities is to explore those implicit relationships, explicitly represent those relationships by directed links among DEs and hence to give a structure to those relationships (in the form of a representation) as shown in Figure 5.2. In essence, forward linking offers a means to reveal design entity relationships and provides us an additional perspective to understand the specific interaction that a design entity can has with other design entities.

There are three steps followed to develop design entity relationships:

1. Consider a source DE where linking is to begin, say DE1.
2. For the descendants of DE1, link all these descendants DEs by following the relationships revealed from a design rule or constraint. Forward linking of a design entity forms an acyclic structure; therefore, the traversal will terminate when all descendants DEs have been traversed.
3. DEs that are not specified in an underlying design rule/constraint, but are located between DE1 and its descendant DEs are also retained, as shown in Figure 5.2 e.g.DE10 in Figure 5.2. This is because if these unspecified DEs are not traversed, then the chain will be broken and some of the required DEs downstream from DE1 would be overlooked.

The concept of forward linking design entities by following the relationship revealed from the design rules is illustrated in Figure 5.2 (we have just referred to design rules in Figure 2, however, the same concept is applicable in the case of design constraints).

Figure 5.2 provides a conceptual illustration of design entity relationships (by arrowed links), design entities (DEs) and design rules (DRs) that encompass a set of design entities. Essentially, forward linking of design entities support making the design entity relationship explicit, and these are revealed from design rules and constraints. For example, as shown in Figure 2, forward linking of DE5 from DE2 can be represented as DE2→DE10→DE1→DE3→DE4→DE5 or forward linking of DE5 from DE8 can be represented as DE8→DE6→DE5. In both of these examples, DEn are design entities and the open end arrow '→'are the links that represents the design entity relationships.

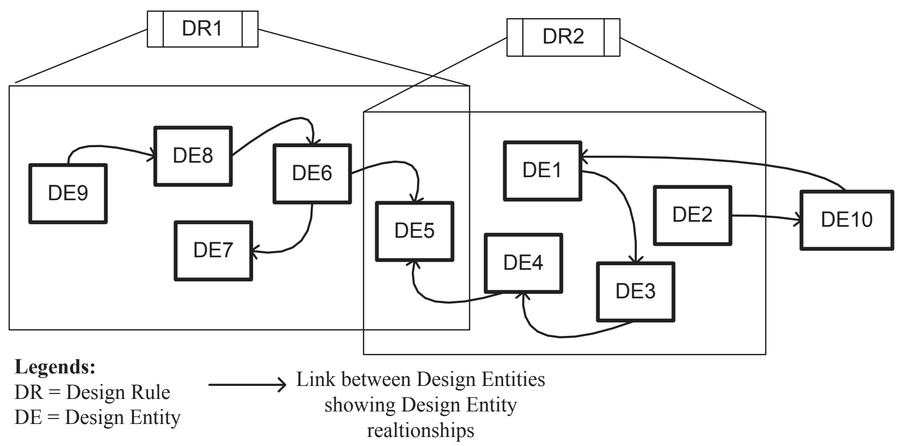


Figure 5.2: Envisioning design rules, design entities and design entity relationships[3]

This is adopting a forward linking approach, and linking all design entities for all design rules/constraints within a design decision and subsequently across all design decisions under the study. In doing so, we make the relationships between design entities explicit and represent these relationship in a structured form of representation.

Backward tracing for dependency relationships **–** Once a design entity relationship was identified and represented in a structured form, we trace backward these design entity relationships for dependency analysis. Backward tracing of these relationships may allow us to identify possible dependencies and (hence change impacts) among design entities.

[3] Here DE10 is not associated with a design rule while forward tracing, but located between DE2 and DE1. DE10 is retained due to the fact that if the unspecified DE (such as DE10) is not traversed, then the chain will be broken and some of the required DEs downstream to and from DE1 will be overlooked

Are web developers/architects mostly interested in studying what dependency relationships there are (from one design entity) that may affect other design entities? Further, are Web developers/architects also interested in knowing what needs to be modified if they make changes to a design entity (say DE1)? In order to address these concerns one can trace backwards from DE1 to other DEs through the design entity relationships (identified from forward linking of design entities). During backward tracing, it is easy to pick one DE, which Web designers/architects are interested in, and then backward trace from it. Consequently, a network of dependencies among design entities can be developed as shown in Figure 5.3.

Graphical representation of dependencies such as the dependency graph is reported as a mature strategy that is available for CIA (Bohner and Arnold, 1996). Therefore, we represented the network of dependencies among design entities through a dependency graph. The design entity relationships do not create a cyclic structure; therefore, backward tracing of design entity relationships also results in a directed acyclic dependency graph. A directed acyclic graph is a directed graph if it contains no cycle or does not have any self-loop (Chen, 1997). For backward tracing, we follow the backward direction of the design entity relationships (represented by edges) that leads us from a vertex (DE_n) to the predecessor (DE_{n-1}) in a directed acyclic graph (Savage, 1999). In this way all design entities that a DE depends on, directly and indirectly, are traversed and retrieved. There are two steps followed to develop a dependency graph.

1. Consider a source DE where tracing is to begin, say DE1
2. For all the predecessors of source DE1, traverse the DEs recursively by following the backward direction of the design entity relationships. Since the design entity relationships structure is not cyclic, the traversal will terminate when the entire predecessor DEs have been traversed.

By following the above two steps a dependency graph can be developed that mainly supports the understanding of the dependencies among design entities. As shown in Figure 5.3, a dependency graph is a directed graph, represented by tuple (DE, DEdge) (Rosen, 2006), where DE is a set of vertexes representing design entities, and DEdge is a set of dependency relationships (between two vertexes) represented by directed edges ‘e’ $\in$ DEdge , such that

1. All dependency relationships must be associated by DEdge with two DEs.
2. No subset of edges in DEdge form a directed cycle[4]

According to Clause 1, a dependency relationship is connected to exactly two DE vertexes through an element of DEdge, one is a source design entity DE_s and the other is a destination design entity DE_d. Clause 2 specifies that the insertion of an edge is not allowed if it results in a directed cycle. Essentially, this means that the edges maintain the integrity whereby dependency relationships avoid creating a directed cycle, directly or indirectly. We have given 'e' the value a dependency name during the instantiation of PMCIA (see Figure 11 in Appendix E).

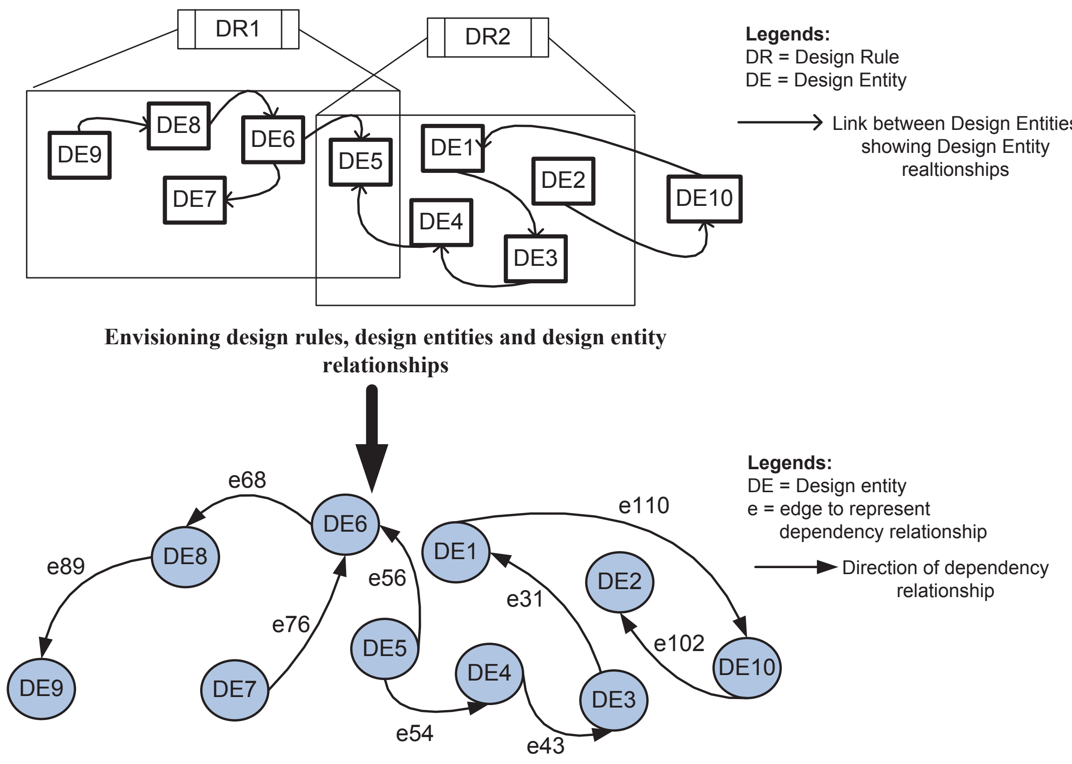


Figure 5.3: Dependency graph

Once the dependency graph has been constructed, the set of estimated impacts on architecture design are determined by taking into account every design entity that depends on other design entities directly and indirectly. For example, as shown in

[4] In graph theory, a directed graph may contain directed cycles, a one-way loop of edges. Mostly, such cycles are undesirable to avoid the deadlock (Rosen, 2006).

Figure 5.3, the impacted design entity such as DE6 can be determined with the direct dependency (represented by e56) from DE5 to DE6 whereas DE2 can only be determined by using a chains of dependency, from DE3 to DE2 as $.DE3 \xrightarrow{e31} DE1 \xrightarrow{e1\,10} DE10 \xrightarrow{e10\,2} DE2$.

The backward tracing serves two purposes. Firstly, it accounts to a corresponding design entity relationships that are there due to design rules and constraints. Secondly, it helps us to understand the possible dependencies between design entities, directly and indirectly-through a chain of dependency relationships.

Both forward linking and backward tracing are two important aspects of PMCIA as described in previous sections. Our proposed PMCIA is comprised of three steps, activities in each step, flow from one step to another, and a set of inputs/outputs. The high level model of PMCIA proposed in this thesis is depicted in Figure 5.5 and the detail of step#1 and step#2 is illustrated in Figure 5.6 and 5.7 respectively. In the following sub-section, we describe in detail the inputs, steps, activities in each step, flow from one step to another, and the output of the proposed PMCIA.

5.5.1 Initial inputs

Given the discussion in Section 5.3, overall, we consider design decisions as an important input for PMCIA along with other supportive inputs such as change request/information, previous experience, business processes specifications, architecture requirements and design representation as discussed below.

Change requests/information - change requests are the initial source of change information for PMCIA. The change request can be initiated for a number of different types of changes such as business processes changes.

Previous experience - previous experiences are utilised to better predict the change extent, and for a deeper understanding of impacts on the architecture design.

Architecture requirements and design representation - Architecture requirements specification is the document explaining architecturally significant requirements and

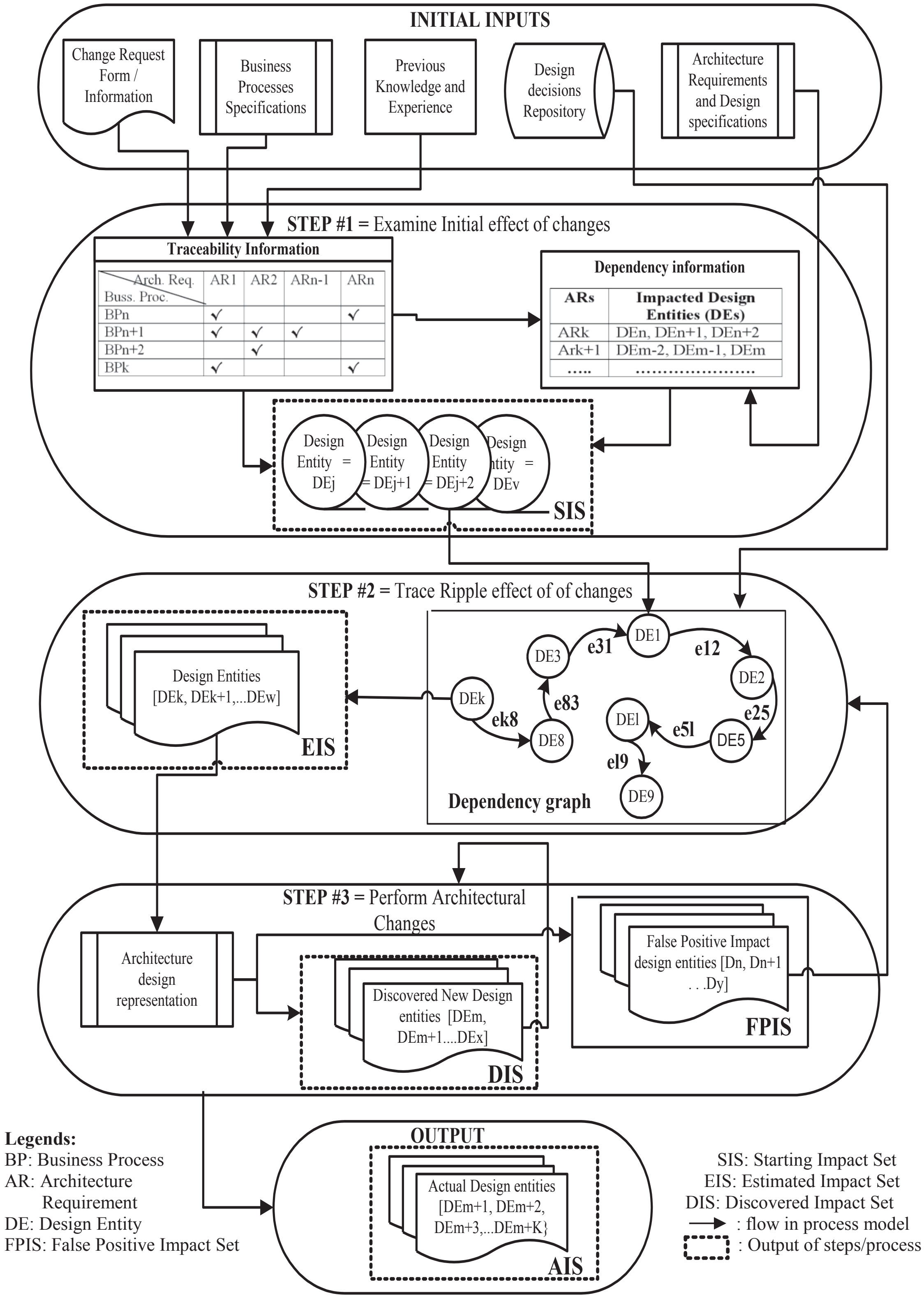


Figure 5.5: Process model for change impact analysis, illustrating three steps and the activities in each step.

provides a set of quantitative statements outlining what a system implementation must do in order to comply with the architecture (Group, 2010). Whereas, architecture design representations is used to capture those architecture requirements (Mike J. Van Wie et al., 2003). Architecture design representations come in many different forms e.g. architecture sketches, view-based architecture models, and textual descriptions of system components and so on. However, proposed PMCIA is limited to architecture design representations that include architecture design languages, visual references and the interface of architecture components namely WebML (Ceri et al., 2000b), OOHDM (Jacyntho et al., 2002) and WIED (Tongrungrojana and Lowe, 2004b).

5.5.2 Step#1: Examine initial effects of changes

The initial effect of business processes changes on the architecture design is determined in step#1. In step#1, there are four sources of input i.e. change requests/information, business process specifications, architecture requirements and design specifications; and previous experiences are used in step#1 of PMCIA. In addition to these sources of information, we have further utilised two types of information (i) traceability information (the traceability matrix depicting the relationships between Business Processes (BPs) and Architecture Requirements (ARs)) and (ii) dependency information (dependency between ARs and impacted design entities). Figure 5.6 describes in detail the step#1 of PMCIA, where mainly traceability information, impacted architecture area and dependency information are identified. Given the purpose of step#1 is to identify initial impacted design entities at architecture design, we propose the following five activities during step#1.

1. Determine the scope of business processes changes (① in Figure 5.6)
2. Trace from business processes to architecture requirements (② in Figure 5.6)
3. Determine the scope of architecture requirements changes (③ in Figure 5.6)
4. Identify effected architecture area, (④ in Figure 5.6)
5. Identify impacted design entities, (⑤ in Figure 5.6

Determine the scope of business processes changes - The scope of business processes can be determined by locating all those proposed business processes changes in the textual description of business processes specifications.

***Traceability from the business process to architecture requirements* -** PMCIA is leveraged by existing research on the topic of the traceability matrix. For this research, we have referred to the traceability matrix proposed by Palmer (1997) and Pohl et al. (2001). While developing the traceability matrix, we have identified the relationships between BPs and ARs by examining change request/information, business processes specifications, architecture requirements specifications and by employing previous experiences of change impact analysis. The identified relationships are specified in the traceability matrix, where each row corresponds to a business process, and each column corresponds to an architecture requirement. These relationships are expressed by putting a mark where the row of the business process and the column of the architecture requirement intersect (② in Figure 5.6). In essence, the traceability matrix provides us with the necessary information to trace those architecture requirements that are affected by the proposed business processes changes.

It is important to note here (as described in Chapter 4), that we have observed from the investigation of industrial practices that mostly the traceability matrix is not developed or maintained to provide sufficient support during CIA (Mehboob and Zowghi, 2009). Therefore, we proposed the traceability from business processes to architecture requirements as an activity during step #1 of PMCIA, (② in Figure 5.6). If the same traceability matrix as we have mentioned in Figure 5.6 is already being developed and maintained then Web developers/architects can skip the traceability matrix activity (② in Figure 5.6) and resume step#1 - from determining the scope of architecture requirements change (③ in Figure 5.6).

***Determine the scope of architecture requirement changes* -** The scope of architecture requirements changes is comprised of all those architecture requirements that are affected by business processes changes. The scope of architecture requirements changes (③ in Figure 5.6) can be determined by taking each business processes change (from scope of business process changes) and then tracing that business process to the level of architecture requirements using the tractability matrix. All these traced architecture requirements will be included in the scope of architecture requirements changes.

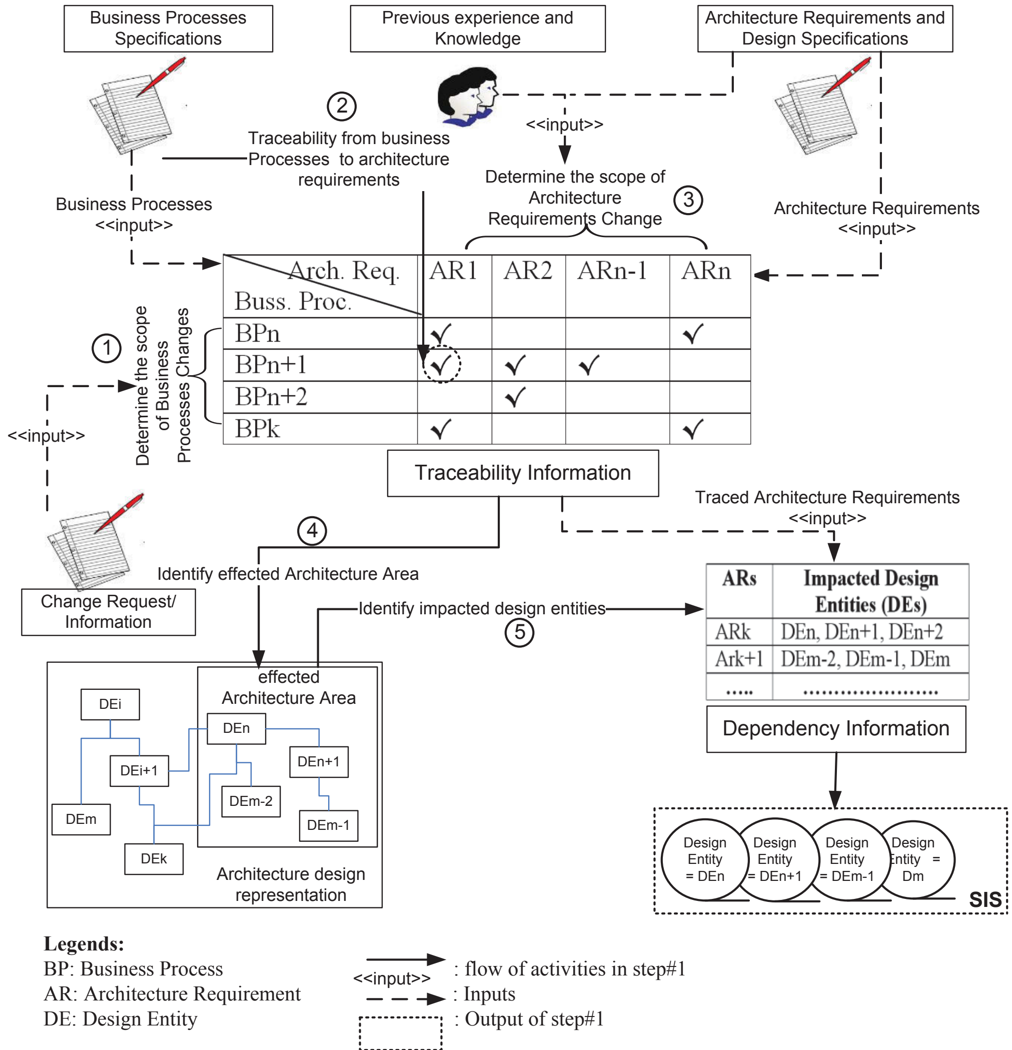


Figure 5.6: Detail of step#1 in the PMCIA as shown in Figure 5.5

Identify effected architecture design area **-** All traced architecture requirements (from ③ in Figure 5.6) are used to determine the effected architecture design area (④ in Figure 5.6), and where the changes need to be implemented. An effected architecture design area characterises a partial representation of the architecture design where those traced architecture requirements are implemented. Taken into consideration is each traced architecture requirement and its implementation at the level of architecture design. We can identify an architecture area(s) that gets affected by the modification of those architecture requirements. We refer to architecture area(s) as an effected architecture design area(s) in PMCIA.

Identify impacted design entities - After the identification of the effected architecture area, the underpinning design entities (⑤ in Figure 5.6) are determined. Architecture requirements depend on a set of design entities for implementation at the architecture design level. Therefore, for each traced architecture requirement, we look for all those design entities that implement them. Subsequently, the effected architecture area (④ in Figure 5.6) confines a set of DEs on which the traced architecture requirement depends on. All those DEs that lie in that effected architecture area are considered as a starting impact set (SIS).

Starting Impact Set (SIS) - Both the identification of the traceability information (between BPs and ARs) and the dependency information (between ARs and DEs) facilitate to identify the starting impact set i.e. the set of design entities that are thought to be initially affected by changing business processes. These effected design entities are considered as initial impacts (of business processes changes) on the architecture design.

5.5.3 Step#2: Examine the ripple effects of change

The ripple effects of changes on architecture design are identified in step#2. Main sources of information used in step#2 of PMCIA are design decisions and starting impact sets (SIS). Figure 5.7 describes in detail step#2 of PMCIA. Mainly, design rules/constraints are used for the identification of possible dependencies among design entities, and further those dependencies discover the ripple effects of changes at architecture design. Followings are four main activities in step#2.

1. find design decisions,
2. explore design rules/constraints,
3. forward linking for design entities relationships and
4. backward tracing for dependency relationships.

Find design decisions - For each design entity from SIS, related design decisions can be retrieved from the design decisions repository by a search query, as shown in ① of Figure 5.7. Commonly, design decisions reflect information such as decision identifiers, related design entities, design rules, design constraints, alternative solutions, and trade-off, etc. (Babar and Gorton, 2007, Capilla and Nava, 2008, Capilla et al., 2007, Jansen et al., 2007). From each retrieved design decisions (as shown ① of Figure 5.7), we have considered design rules, design constraints along with design entities and decision identifiers. The relevance of design rules and constraints in relation to CIA has already been discussed in Section 2.8.

Explore design rules/constraints- Mostly, design rules and constraints are documented in natural text from within the description of design decisions (either in a section of template or as an entry in the design decision repository/tool record). From that textual description of design rules and constraints, all design entities are selected as shown in Figure 5.7a as underlined text. In this way all design rules or constraints are explored and design entities are selected from the textual description.

> For example, to describe the context of navigation for Web systems architecture, a design rule can be stated as "Each navigational object is *identified by* a set-based object, *providing accesses* to content through navigational links and *defined by* an InContext object" (Garrido et al., 1997).

Figure 5.7a: An example of a design rule adopted from (Garrido et al., 1997).

Forward linking for design entities relationships- Given that design rules and constraints inherently bind multiple design entities together, and hence they implicitly capture relationships between design entities. After the identification of design entities, the textual description of design rules and constraints are further examined. From the given example of the design rule in Figure 5.7a, design entities such as Set-based navigation, InContext, and InContextSequential are inherently binded together, and there are implicit relationships among those design entities (shown in italics in Figure 5.7a). These implicit relationships are retrieved by forward linking design entities (③ in Figure 5.7), thus making design entity relationships explicit and representing them with direct links (see design entity relationship in Figure 5.7, ③). Forward linking of design entities supports to envision design rules/constraints, underlying design entities and to represent design entity relationships in a structured form.

Backward tracing for dependency relationships - By backward tracing of design entity relationships, the possible dependencies among design entities are determined and

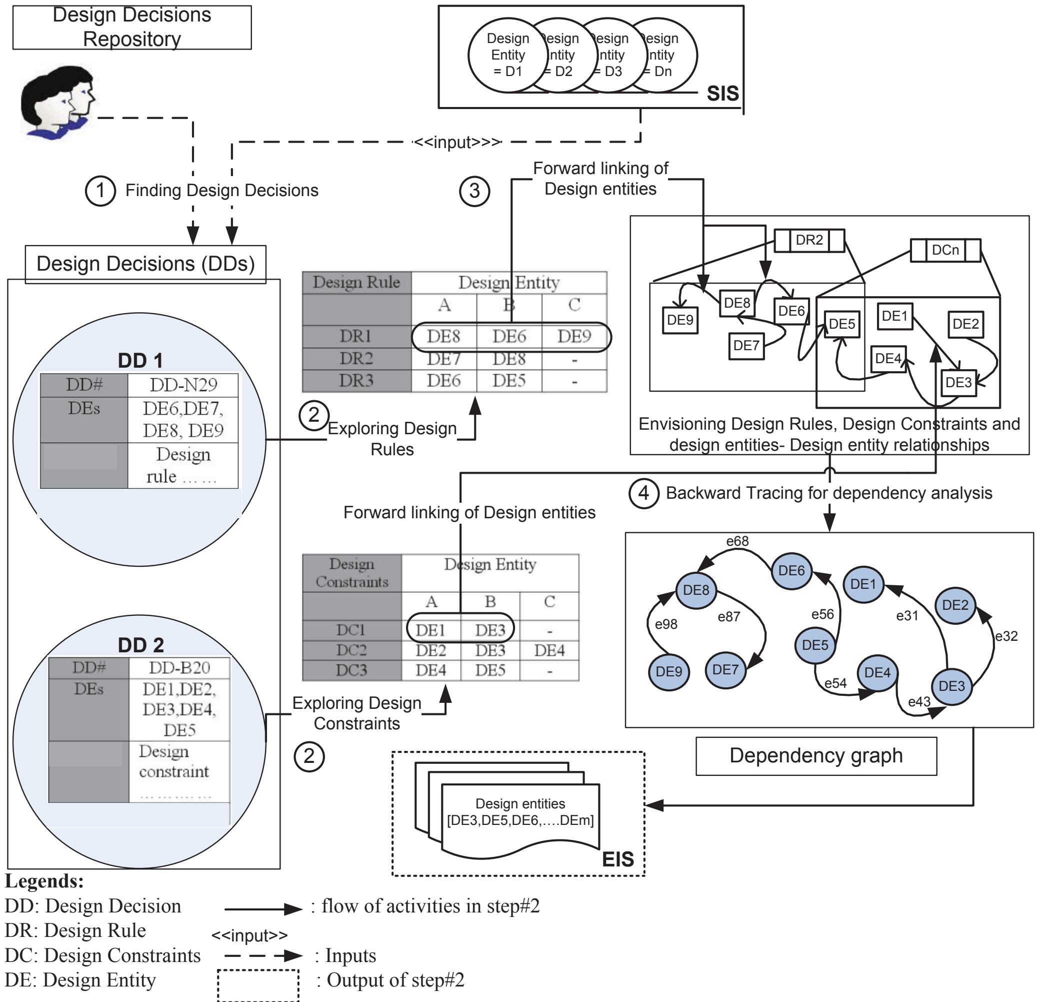


Figure 5.7: Detail of step#2 in the process model as shown in Figure 5.5

represented by a dependency graph (④ in Figure 5.7). Each node in the graph represents a design entity, and each edge in the graph represents a dependency relationship between two design entities. Once the dependency graph has been developed, the ripple effects of changes (i.e. set of estimated impact design entities) can be determined by traversing all dependency relationships in the graph.

Estimated Impact Set (EIS) - This is the set of design entities that are estimated to be affected by applying a change assessment approach such as the dependency graph, inferencing, slicing or heuristics methods as reported in the literature (Bohner and Arnold, 1996). However, in PMCIA we have focused on the dependency graph approach as described before in detail. EIS includes the SIS and can therefore be seen as an expansion of the SIS. The dependency relationships in the dependency graph are used to identify both direct design entities that are connected by a dependency relationship and indirect design entities that are connected by a chain of dependency relationships. The identification of design entities (as elements of EIS) is repeated by traversing dependency relationships until all design entities (that may be affected directly and indirectly) are discovered. Design entities in EIS are those design entities that are estimated to be affected while tracing the potential impacts of modifying design entities from the starting impact set (SIS).

5.5.4 Step#3: Perform architectural changes

In PMCIA, the actual emphasis is on the estimation of impacts, since what was actually changed is not fully known until after the software change is complete. Therefore, the purpose of step#3 is to validate EIS and may further discover other impacted design entities, if any. In step#3 of PMCIA, the main source of information used is Estimated Impact Set (EIS) and architecture design representation.

For each design entity identified (as EIS) in step#2, adequate changes are implemented in the architecture design. While a change is being implemented, there may likely to be other impacts categorised as (i) Discovered Impact Set (DIS)- represents an under-estimation of impacts, and (ii) False Positive Impact Set (FPIS)- represents an over-estimation of impacts. DIS can be determined after modifying each design entities (from EIS) at the architecture design and the other design entities can be examined in the architecture design that possibility were affected by making those changes. Similarly, FPIS can be determined by examining those design entities (from EIS) that do not require any modification. It means that FPIS are those design entities which are the over-estimation from the set of EIS. The objective of most CIA approaches is to have the estimated impact set produced from tracing potential impacts as close to the Actual Impact Set as possible by identifying true impacts while eliminating false-positives. Similarly, in PMCIA, by identifying adequate dependency relationships among design entities that are revealed from design rules and design constraint, we adequately estimated impacted design entities (EIS) and therefore the possibility of DIS and FPIS being a large set could be minimised.

As shown in Figure 5.5 the output of proposed PMCIA is Actual Impact Set (AIS). This impact set represents the set of design entities in architecture design that are actually modified as a result of implemented business processes changes. The EIS plus additions of DIS and deletions of FPIS represents the AIS (Bohner and Arnold, 1996). Therefore, the actual architecture impact set can be represented as, AIS = (EIS$\cup$ DIS) \ FPIS.

5.6 Mapping between Identified CIA features and PMCIA

In this section we describe a mapping between reported features that a CIA approach should possess (Mehboob and Zowghi, 2009) and the PMCIA that we proposed in this chapter. This mapping demonstrates that PMCIA is developed in response to the features (of a CIA approach) that are deemed essential by Web designers/architects. In the following table, the first column describes the reported features of a CIA approach and the second column describes how PMCIA is developed in response to those features.

Table 5.2: Mapping between CIA features and the process model of CIA

Identified CIA features	PMCIA
FA-1 link business processes and architecture design (information architecture IA) entities to keep the traceability consistent between them.	In step#1, the tractability from business process to architecture requirements and further tracing from architecture requirements to impacted IA design entities provide a complete linkage from business process to IA design entities.
FA-2 Impact analysis at architecture design (IA) facilitates to: • FA-21 Identify how changes made at one abstraction of architecture design affects design entities on the other abstraction of architecture design (structure, behaviour, location etc.)	• Investigate implicit influences (through design entity relationships) that a design entity may have on other design entities underlying a design rules/constraints, and further how these design entity relationships support to identify dependencies among design entities for the

• FA-22 Identify how changes made in the architecture design affects another part of the architecture design within a single abstraction (structure to structure, behaviour to behaviour etc.) • FA-23 Identify which design entities are necessary to modify after making a change	purpose of CIA. • Investigate implicit influences (through design entity relationships) that a design entity may have on other design entities underlying a design rules/constraints and further how these design entity relationships support to identify dependencies among design entities for the purpose of CIA. • The iterative aspect of change impacts identification has been covered in Step#3 so as to determine what design entities need to be modified after making a change.
FA-3 Process development support for: • FA-31 Activity descriptions (steps) • FA-32 Each activity (from FA-31) should be better guided and instructed (guidelines)	• Three steps and related activities of the process model of CIA are described. • Activity guideline and how to perform each activity is provided in each step.
FA-4 Process working support for: • FA-41 Mechanism to extract information related to the dependencies among design entities of architecture (IA) • FA-42 Visualization of inter-level (e.g. structure to behaviour) and intra-level (e.g. structure to structure) dependencies among	• Step#2 covers the details of how to discover design decisions and design rules/constraints related to design entities considered for modification. • Dependency graph and dependency relationship with their dependency description serve the purpose.

design entities of architecture	
FA-5 Process management support for: • FA-51 A consistent and systematic way to identify impacts on architecture design • FA-52 Support feedback mechanism used to evaluate the results of impact analysis (impact set) after retesting affected software	The proposal for a process model along with steps, activities and guidelines provide the necessary basis for a consistent and systematic way to identify architecture design. Iterative and discovering nature of CIA is covered in Step#3 so as to focus on feedback of change impact set after making change and retesting. A feedback loop from step#3: perform architectural changes to step#2: trace potential impacts will help evaluate the results of impact analysis (impact set).
FA-6 Avoid overestimation (unnecessary change impacts) and underestimation (overlooking change impacts)	A structured representation of design entity relationships and dependency relationships provide insight and detail of influences that a design entity can have on another design entity, and thus avoid overestimation and underestimation. Starting the impact set, estimated and actual impact set, further refine the resulting impact set so as to avoid the possibility of both overestimation and underestimation.
FA-7 Employ architecture design decisions repository/information	Design decisions and their underling information such as design rules and design constraints from design decisions repository have been

	employed.
FA-8 Analyse impacts adequately on architecture design before detailed design actually begins to reduce un-necessary rework	By employing design rules and design constraints, a detailed insight/ understanding of dependencies among design entities is gained and subsequently, this understanding of dependencies supports to adequately analyses impact on the architecture design before detailed design actually begins
FA-9 Identify change impacts on architecture design (IA) resulting from business processes changes	The process model of CIA employs both traceability from business processes to architecture requirements, and dependency analysis from arcinterview's hitecture requirements to design entities within an impacted architecture area. Thus, the process model supports to identify change impacts on architecture design (IA) resulting from business processes changes

The mapping as shown in Table 5.2 provides a sufficient basis to claim a successful incorporation of the essential features of CIA. These features include the early identification of impacts (FA-2) and the execution of analysis tasks (FA-3, FA-4, FA-5). Indeed, impacts identification and the execution of analysis tasks are arguably the fundamental aspects of the CIA process (Bohner and Arnold, 1996, Jonsson, 2007) and they are adequately addressed by PMCIA.

5.7 Discussions

We have described PMCIA in detail in Section 5.5. Furthermore in Section 5.6, we have presented the possible mapping between required features for CIA approaches and PMCIA (developed in response to those features). Given the discussion made both in

Section 5.5 and 5.6, we argue that PMCIA can be used to support early identification of change impacts in Web systems and in turn PMCIA also addresses the related characteristics of Web systems. Further, PMCIA provides a single approach to combine traceability and dependency analysis, and employs design decision information, and presents a step-by-step and structured approach to support adequate identification of change impacts. In the following paragraphs, the distinguishable aspects of PMCIA are described in detail.

Firstly, in Section 2.3, we described that a potential change in business processes may typically lead to fundamental changes in the supporting architecture design and it is mainly so due to both (i) the business processes being intricately interwoven with Web systems and (ii) there being a tight-connection between business processes and the architecture design of Web systems. While investigating these characteristics of Web systems, it was observed that the resulting focus to address two characteristics is adequate to support identification of impacts on architecture design resulting from business processes changes. *While investigating does PMCIA address those characteristics?* It was observed that firstly, PMCIA specifically focuses on the linkages between business processes and architecture design entities, and thus essentially supports to identify impacts on architecture design resulting from business processes changes. Essentially, PMCIA may specifically focus on the traceability from business processes to architecture design entities, mainly to identify which design entities become affected by business processes changes (breadth of change propagation). Secondly, PMCIA also pays specific focus to the possible dependencies among architecture design entities and supports an adequate identification of impacts on architecture design (before the implementation actually begins). A specific focus on analysing dependencies among design entities supports to identify what design entities become affected by the modification of other design entities in architecture (depth of change propagation). It was identified both from an illustrated example (in this chapter) and validation of PMCIA (see Chapter 7) that employing traceability and dependency analysis in a single approach facilitates identifying impacts on architecture design resulting from business processes changes- early identification of change impacts in Web systems. Overall, the findings indicate that PMCIA addresses the two characteristics of Web systems by taking into account the identification of impacts on architecture design resulting from business processes changes.

Secondly, most of the architecture levels CIA approaches support impact identification only for those changes that are made to architecture design (see Section 2.7). These CIA approaches do not support impacts identification across multiple system artefacts such as from business processes to architecture design. In Section 2.5, we have described that there is a distinct nature of linkage from business processes to architecture design in Web systems. While adopting the existing architecture level of CIA approaches in Web systems, there is a tendency to overlook those impacts (on architecture design) that are resulting from business processes changes. PMCIA supports impacts identification across multiple system artefacts and do not limit impact identification to only those changes that are made to architecture design. In particular, PMCIA takes into account the identification of impacts on architecture design resulting from business processes changes, and thus supports early identification of change impacts. While focusing on the identification of impacts on architecture design resulting from business processes changes, essentially, PMCIA also supports to addresses the characteristics of Web systems- in Section 2.5, we already argued that those characteristics guide us toward early identification of change impacts in Web systems. This aspect of PMCIA is quite aligned with the first suggested extension of CIA approaches as described in Section 5.2.

Thirdly, as it has been discussed in Section 2.7 that CIA approaches can be classified into two possible groups based on (i) traceability analysis and (ii) dependency analysis. Approaches that are based on traceability analysis support to identify change impacts from one system artefact to another system artefact, covering the breadth of change impact analysis and opt to support an abstract level of impacts identification on system artefacts. Another group of CIA approaches focus on dependency analysis and supports to identify impacts within a system artefact i.e. from one component of system artefact to another component of system artefact and covering the depth of change impact analysis. PMCIA as a single approach takes into account traceability analysis to address the connection between business processes and architecture design; and dependency analysis to address the dependencies between design entities of architecture. Consideration of both traceability and dependency analysis is a distinct characteristic of PMCIA and leverage to support an adequate identification of change impacts on architecture design of Web systems.

Fourthly, it has been reported in Section 2.7, that most of the CIA research has been done on technical aspects, assuming the availability of detailed and supportive infrastructure (such as supporting tools, documents and activities) to execute change impact analysis. Additionally, most of the CIA approaches do not cover process aspects such as details of steps to follow, activities in each step, flow from one step to another and inputs/outputs for the CIA approach. For the proposed PMCIA, the steps, activities to execute those steps, flow from one step to another and inputs/output have been described in sufficient detail. Indeed, these process aspects of PMCIA support a consistent, rigorous and step-by-step approach. Subsequently, PMCIA results in a more objective and well-founded analysis for change impacts, and also tends to reduce the risk that something is missed during the analysis. The process aspects of our proposed CIA approach is another distinct characteristic and quite aligned with the second suggested extension of CIA approaches as described in Section 5.2.

Finally to support early identification of change impact in Web systems, PMCIA employs design decision information such as design rules and constraints. In Section 2.8, we argued that a lack of explicit support to utilise design decisions information such as design rules and constraints render the existing CIA approaches less than useful in Web system context because early design knowledge such as design decisions and their underlying information have not been well-characterised and employed yet. Design rules and constraints inherently bind multiple design entities together, and hence implicitly capture relationships between these entities. In PMCIA, these rules and constraints are analysed and lead to the identification of possible dependencies (hence change impacts) among design entities. In essence, the use of design rules and constraints to support CIA is arguably a distinct feature of PMCIA and quite aligned with the third suggested extension of the CIA approach as described in Section 5.2.

5.8 Chapter Summary

In this chapter we have presented PMCIA, a process model of CIA to support the identification of impacts on architecture design resulting from business processes changes- early identification of impacts in Web systems. While developing PMCIA, we have proposed three possible extensions to Bohner's CIA approach as described in Bohner & Arnold (1996). As we have discussed and illustrated by example, PMCIA supports impacts identification on architecture design resulting from business processes changes. In particular, this approach focuses the necessary components for systematically and rigorously executing the CIA activity to support early identification of change impacts in Web systems. This is also an important step in providing a possible solution to addresses the related characteristics of Web systems. In Section 2.3 and 2.5, we have already argued that the resulting emphasis of Web systems characteristics is to provide support for early identification of change impacts in Web systems.

We have also presented a data model to support the data input requirements of the process model. Further, as an important source of information we have used design decisions information such as design rules and constraints during impact analysis. There are mainly two reasons for using design rules and constraints. Firstly, design rules and constraints facilitate discovery of the possible dependencies among design entities. Secondly, if this information is used to identify impacts on the architecture design resulting from business process changes, then Web developers/architects can identify impacts early (at the architecture design stage) and will not have to delay the impact identification for later stages of Web systems development. In fact, as far as we know, this is the first attempt to support early identification of change impacts in Web systems while employing design decisions information.

In Appendix E, we have also demonstrated the instantiation of PMCIA, as a proof by demonstration, with an exemplar Web system so as to provide its initial utility. Results from proof by demonstration are encouraging as it provides the ability to understand the dependencies among design entities of architecture design and subsequently supports identification of impacts on architecture design resulting from business processes changes. In the next chapter we present a validation of PMCIA, aiming to find out whether PMCIA support for early identification of changes impacts on Web systems. The case study design and validation of PMCIA are described in Chapter 6 and Chapter 7 respectively.

Chapter 6: The Case Study Methodology

6.1 Introduction

This chapter describes the research methodology, data collection and analysis methods used to validate the process model of CIA. Software engineering is a field of applied research. Mostly, software engineering research involves investigating how people aim to develop systems and work in teams in organisations. It is commonly argued that a clear understanding of the development process and its underlying activities is necessary to successfully produce systems, and to manage validation of those processes. In order both to build an understanding of process and to validate process, researchers need an appropriate research methodology (Runeson and Host, 2009) – one such example is the use of cases studies. A case study as a research methodology is considered feasible when individuals, groups, organisational and social phenomena are being investigated (Runeson and Host, 2009, Yin, 2003), hence it is often found suitable for software engineering research.

We have focused on the process that describes how case study research is designed and implemented. Subsequently, in this chapter we present a case study as a research methodology, case study design, data collection and data analysis methods. The rationale, strengths and weaknesses of a case study as a research methodology are discussed. Considering an industrial case for the case study research, various data collections and analysis methods are employed. Further, during case study a triangulation of data and triangulation of methods are selected. This chapter begins with describing a case study as a research methodology in Section 6.2, case study protocol in Section 6.3, and case study design in Section 6.4. Further, we describe data collection in Section 6.5, collection evidence-applying case study design in Section 6.6, data analysis methods in Section 6.7 and case study reporting in Section 6.8. We discuss research reliability, validity and generalisability in Section 6.9. We finished this chapter with a chapter summary in Section 6.10.

6.2 Case study - A Research Methodology

We refer to the general definition of the term case study according to Robson (2002), and Yin (2003). These three definitions agree that a case study is an empirical method aimed at investigating contemporary phenomena in their contexts. There are a large variety of case study types (Cunningham, 1997) and classifications are based on different purposes of the research methodology such as exploratory, descriptive, explanatory and improving (Robson, 2002). Additionally, Klein and Myers define three types of case studies depending on the perspective, i.e. positivist, critical and interpretative (Klein and Myers, 1999). Runeson and Host (2009) reported that software engineering case studies tend to lean toward a positive perspective and especially toward explanatory type studies, where a researcher searches for formal propositions, measures variables and draws conclusions from a sample (*positivist)* as well as seeking an explanation of the situation or a problem(e*xplanatory)*. In our research, a case study is explanatory in nature, seeking an explanation of the proposed solution of the process model of CIA in Web systems, and producing a descriptive analysis in a selected environment. Furthermore, Leedy asserts that the case study methodology is suitable for studying phenomenon in a complex environment (Leedy and Ormrod, 2004). In this research, the case study is designed as a means of studying the process model of CIA and thus its validation within the context of a Web systems project, which is considered a complex environment.

Kitchenham has described various advantages of case studies (Kitchenham, 1996, Kitchenham et al., 1995) as a research method for software engineering. These advantages include 1) the case study can be incorporated into the normal development activities, 2) it shows the real effects of the approach in real situations, and 3) it enables validation of the work by actual practitioners. In this research, the case study is designed to be incorporated into the normal development activities of a Web development organisation to show the real effects of the process model of CIA in a real-life Web systems project. Additionally, the case study design will validate the process model of CIA in terms of early identification of change impacts in Web system projects. However, a few disadvantages of case studies were also reported by Kitchenham (1996) including 1) that with little or no replication a case study may give inaccurate results, 2) there is no guarantee that similar results will be found on other projects, and 3) there are few agreed standards/procedures for undertaking a case study in software engineering. Additionally, in the past some have argued that case studies are extremely limited in providing conclusions that can be generalised, given that they are based on a single set of circumstances. However, over time this has regularly been rejected (Flyvbjerg, 2007, Runeson and Host, 2009), stating instead that case studies are acceptable and can provide very meaningful information provided they are designed and performed with appropriate degrees of rigor and structure (Hamel et al., 1993). Furthermore, case studies have also been criticized for being of less value and being biased by researchers. This critique can be met by applying proper research methodology practices as well as reconsidering that knowledge is more important than statistical significance (Flyvbjerg, 2007, Lee, 1989). Another issue is the ethical consideration in association with the researcher's personal integrity, sensitivity, or personal biases while conducting the case study (Singer and Vinson, 2002). To address these issues, generally, detailed consideration of ethical issues of validity and reliability are covered in relation to the case study research.

Case studies represent an appropriate choice for our research goal as they enable us to validate our proposed solution in a real-world project and within a complex organisational environment, which is vital given our research is partially based on addressing aspects stemmed from practices. According to the selection criteria for case study as reported by Kitchenham (1996) and Bratthall & Jorgensen (2002), a case study is suitable in our research due to following four reasons.

1. the context in which the study is performed is not under the control of the researcher, i.e., the confounding factors that may impact the result are not entirely known or cannot be controlled completely
2. laboratory cases do not have the Web systems complexity to demonstrate the real benefits of our approach for change impact analysis.
3. the benefits are observable on a single project,
4. the timescale for the evaluation is commensurate with the elapsed time of a normal size project.

Furthermore, a number of similar examples are reported in the research literature where a case study as a research methodology has been successfully used for validation of Web engineering approaches (Garrido et al., 2009), for Web application design processes, (Vlaanderen et al., 2009) and for model-driven processes in Web engineering (Foster et al., 2005).

For planning and conducting a case study, we have referred to Yin's work (2003) as it has been widely regarded as seminal, and possibly the most cited reference for general case study research. We also have referred to other specific works on case studies including Kitchenham's work as a case study research method for software engineering (Kitchenham et al., 1995), Perry's work as case study practices for software engineers (Perry et al., 2004), Host's work as a checklists for software engineering case study research (Host and Runeson, 2007) and most recently, Runeson's work on guideline for conducting and reporting case study research in software engineering (Runeson and Host, 2009). Accumulatively, all these research works on case studies describe that a typical case study methodology should address the following components.

1. Case study protocol
2. Case study design
3. Data collection
4. Evidence Collection
5. Data Analysis
6. Reporting

In the following section, we describe in detail each component of the case study research methodology in relation to our work.

6.3 Case Study Protocol

It has been reported that adequate planning for a case study is critical for its success (Runeson and Host, 2009). Further, in order to counter some of the limitations of a case study (as discussed in the previous section) and to ensure a rigor of case study as a research methodology, researchers are encouraged to develop a plan called case study protocol (Runeson and Host, 2009). There are several issues that need to be addressed during case study protocol including selection of data collection and analysis methods, selection of the organisation or department, selection of interview participants, frequency of interviews and selection of related documents for review, etc. (Runeson and Host, 2009). It means that the case study protocol typically serves as a container for the decisions taken for a case study as well as procedures for carrying it through. To specifically focus on case study protocol, Pervan and Maimbo (2005) have described an outline of case study protocol that is quite similar to the one recommended by Yin (2003) and it mainly includes:

1. An overview of the case study (research objectives and case study approach being investigated);
2. Field procedures (credentials and access to sites and sources of information);
3. Case study research questions (the questions that the researcher must keep in mind during data collection);
4. A guide for the case study report (outline and publication of the report).

In this case study, we have focused on the outlines mentioned above and described our case study protocol in the following paragraph.

In this case study, the first element of the case study protocol is related to an overview of the project selected, purpose of the case study and issues being investigated. Subsequently, our case study protocol serves as a guide when conducting the case study while keeping the researcher's focus on the research goals and objectives (Runeson and Host, 2009).

Secondly for our case study, case study protocol prevents the researcher from the situation where he or she may miss collecting data that is planned to be collected by using field procedures (Runeson and Host, 2009). It is reported that case study protocol should be properly designed so the researcher does not control the data collection environment (Yin, 2003). Therefore, we have focused on field procedures including case study design, data collection and analysis methods developed for a case study and these are described in detail in Section 6.

Thirdly, the research questions for this case study are stated in Section 4. Indeed, research questions determine the types of data to be collected, their possible sources, and the research methods to be used in this study. Further, research questions as a part of formulating case study protocol made the research concrete in the planning phase, which helped the researcher to decide what questions to ask and what data sources to use (Runeson and Host, 2009).

The final element of this case study protocol is the guide for the case study report. By their nature, case study findings often portray a complex problem; therefore, we have considered it is important to plan the report for the case study. In this study, the report will be written in the form of research papers at major milestones achieved and communicated through research conferences and eventually in this thesis.

6.4 Case Study Design

A case study design, both as a validation approach and for the generalisation made from it support to build a basis for valid inferences from events and evidences collected during the case study. Keeping in view the importance of case study design, we have designed a case study to validate the process model of CIA that subsequently provides a better understanding of the methodology, and it will provide guidance both to researchers from academia and real practitioners from industry.

6.4.1 Components of case study design

Given that our case study is an ‘explanatory’ type of case study; we have described how and why the process model of CIA can achieve the research objectives referred to in the cases study protocol. Further, we describe how the following five components of case study design as reported by Yin (2003) are addressed by this case study.

1. Study questions, what to know?
2. Study propositions, what to achieve?
3. Unit(s) of analysis, what is studied?
4. Linking data to the propositions, and
5. Criteria for interpreting the findings

Taking into consideration the five components in our case study design, we determined that there are inter-relationships among five components and for this case study design it can be explained as: The study questions can be mapped to a study proposition. The process model of CIA can then generate results by applying it to units of analysis. Then these results are linked to study proposition as evidence through the analysis of results and the interpretation of findings.

The research objectives of our case study research is to provide a validation of evidence for the process model of CIA, in terms of addressing the specific characteristics of Web systems (*Research Question 11*) and supporting early identification of change impacts in Web systems (*Research Question 12*). In order to validate how and why the process model of CIA can achieve its research objectives, the following five important components are described as follows:

Study Questions: Study questions need to be documented and clarified precisely to investigate the phenomenon under study. For our research study, the main study question is "Can a change impact analysis approach that addresses characteristics of Web systems be used to support early identification of change impacts during the Web systems development? This research study question is further deconstructed into three sub questions to be addressed during case study implementation.

RQ11: Does the process model of CIA address the specific characteristics of Web systems?

RQ12: Does the process model of CIA support Web designers/architects in the early identification of change impacts?

The process model of CIA is characterised so as to support identification of change impacts on architecture design resulting from business processes changes. i.e. early identification of change impacts in Web systems. An exemplar as a proof-of-concept has been described in Chapter 5. Indeed, the exemplar has provided initial evidence for the validation of the process model of CIA. Further, using a case study as a research design method, the process model has been validated by comparing the results of two different releases of a selected system in a Web development organisation. For the current release of the system, impacts of business processes change on architecture design were studies that were originally identified by using current CIA approaches in the organisation. In the next release, Web designers/architects were asked to use the process model of CIA for proposed change in their Web system. Further, Web designers/architects were interviewed specifically to determine whether the process model of CIA supported them to identify impact on architecture design resulting from business process changes (i.e. early identification of change impacts in Web systems)

Study Proposition(s) – The study proposition is derived from the study questions, and focuses on assertions that should be examined to answer research study questions. For our case study design the *study proposition* can be stated as, "The process model of CIA support to identify impacts on architecture design resulting from business processes changes and thus addresses the underlying characteristics of Web systems".

Units of Analysis - Units of analysis are the selected resources to be examined in the case study. In our case study design, the process model of CIA was applied to the (unit) "project release". The scope we described for project release includes change request form/information, business process specification, architectural requirement specifications and architecture design documents. Web designers/architects also take into consideration the design decision repository and underlying design decision information.

Linking Data to Propositions - According to Yin (2003), linking data to the proposition tightens the data analysis steps for a case study design. This component of case study design could be related both to the data analysis activity and reporting of case study results. When the process model of CIA is applied to the units of analysis, subsequently, linking data to proposition plays a vital role in connecting the case study results further to research questions, study proposition and finally to research objectives.

Criteria for interpreting the findings - Interpreting the findings leads to the meaning of the case study analysis. In this case study the researcher interprets the meaning of the case study, either the process model of CIA support for early identification of change impact in Web systems, by making comparison of the two releases of a Web systems project.

We considered these five components very important in case study design. Indeed, these components provide adequate guidance for case study execution and serve as the fundamental basis in validating case study methodology. Besides these five components of case study design, site selection is also considered very important for the case study research. In the following section we will describe the site selection for our case study.

6.4.2 Case study site selection

Site selection in case study design is one of the key aspects while conducting the research. Keeping in view the study questions, we have defined the following criteria for case study site selection.

1. The case study site should be an organisation that has engaged in Web systems development.
2. The organisation should be committed to adopting new process for the purpose of improving their Web development practices.
3. The size of the company should be medium to large, and should employ current web systems development technologies.

While choosing a Web development organisation as a case study site, the above selection criteria were considered essential due to following reasons:

Firstly, this study covers change impact analysis in Web systems and our research focus is related to the identification of impacts on architecture design resulting from the changes made at business processes. Thus, an organisation was required to engage in Web systems development and cater for changes in Web systems projects.

Secondly, this study also investigates the validation of the process model of CIA while supporting early identification of change impacts in Web systems. Early identification of change impacts in Web systems involves a systematic and rigorous approach, where CIA approaches need to be adopted and change impact analysis should be controlled and managed. Therefore, an organisation committed to adopting new processes was desirable as a case study site. Keeping in view both the special emphasis of change impact analysis during change management process (CMMI, 2006) and current research focus towards the process aspect of CIA (Jonsson, 2007), it is believed that such an organisation was flexible enough to adopt the process model of CIA in their normal system development practices.

Thirdly, the medium to large size of the organisation was essential since the size of the company and the complexity of the Web technologies in practice could determine the way an organisation can adopt a new process for improving its development practices.

Additionally, in case study design, it was also important to select a case study site that will enable the researcher to study and observe the instantiation of the process model of CIA and examine whether the process model of CIA supports early identification of change impacts in Web system projects.

Case studies can be either a single-case study or multiple-case studies (Yin, 2003). The rationale to adopt a single-case study or multiple-case studies depends on the context and research objectives. Runeson and Host (2009) described that if the context is a specific domain or company, and the two releases of a single project are two units of analysis it is considered as a single-case study. On the contrary, if the context is software companies in general and the research goal is to study a particular practice then the two projects may be considered as two units of analysis in multiple-case studies (Runeson and Host, 2009). In this case study, an organisation was selected as a single-case study and the two releases of a single Web systems project are two units of analysis. However, the case and the units of analysis were selected intentionally as the purpose of selecting a case study was projected to be "typical" or representative of Web systems projects (Benbasat et al., 1987).

In this study, the Web Development Solutions (WDS[5]) organisation was chosen as the case study site. WDS is a multi-national healthcare software solutions provider that primarily caters to clienteles in the UAE and USA. WDS provides services to facilitate the healthcare industry with its practical, simple and effective Web-based healthcare solutions. WSD is an organisation that develops Web based systems and the organisation is committed towards process improvement programs including maintaining ISO 9001:2000 and scaling up its maturity from CMMI level 2 to CMMI level 3. The characteristics of WDS satisfied the selection criteria as discussed before. The detailed description of the organisational context will be presented in Section 7.4.

This study was a single-case design, adopted to validate the process model of CIA with a Web system project in a WDS organisation. WDS represents a particular Web development organisation and can inform us of the validation of PMCIA in a Web systems development setting. During the case study execution, the researcher had full

[5] The name is fictitious to preserve anonymity

access to the WDS site, i.e. document inspection, access to project and organisational repositories, opportunities to talk to Web developers/architects, project managers and the QA manager, participation in project weekly meetings as well as software process engineering group (SPEG) meetings. The four phases of case study design and the application of the case study are described in the next section.

6.5 Applying Case Study Design

As described earlier, this study adopts an explanatory case study methodology to accomplish the research questions RQ11, RQ12. This case study was set up in a WDS organisation (described in Chapter 7) where two releases of the Web system project were analysed, *Release_A* and *Release_B*. In the beginning of the case study, the researcher had full access to most of the WDS's organisational data and projects under development.

The case study was carried out to the end of the *Release_A* lifecycle. During that period, the researcher had developed a good relationship with the WDS personnel and project team, and collected substantial evidence from *Release_A* in relation to changes in business processes and how their impacts were identified at architecture design. The WDS's management further allowed the researcher to continue the case study investigation on *Release_B*. This was a unique opportunity to study change impacts on architecture design in two project releases with full access to most of their data.

This explanatory single-case study was designed to explain and to validate the process model of CIA with a typical representation of a Web systems project in an industrial setting. In a period of over sixty days, the research activities of this case study fell into four phases where some of the phases were interrelated. The overall view of the case study research is illustrated in Figure 6.1. This figure describes each phase of the case study research including research activities, sources of evidences, data collection and analysis methods used. In the following section we describe each phase of our case study. Data collection and data analysis methods are used in all four phases of case study design and are described in detail in Section 6.6 and 6.7 respectively.

6.5.1 Phase 1: Organisational context

The organisational context was the first phase of our case study where the initial research activity was carried out at WDS. The purpose of this phase was to explore and gather initial information regarding the case study site, and the characteristics of the Web systems project. Additionally the aim in Phase 1 was to know the key players in the organisation that could be contacted or could provide key information and other sources of information in relation to this case study. The data collection methods used in this phase was meetings, discussions and observation.

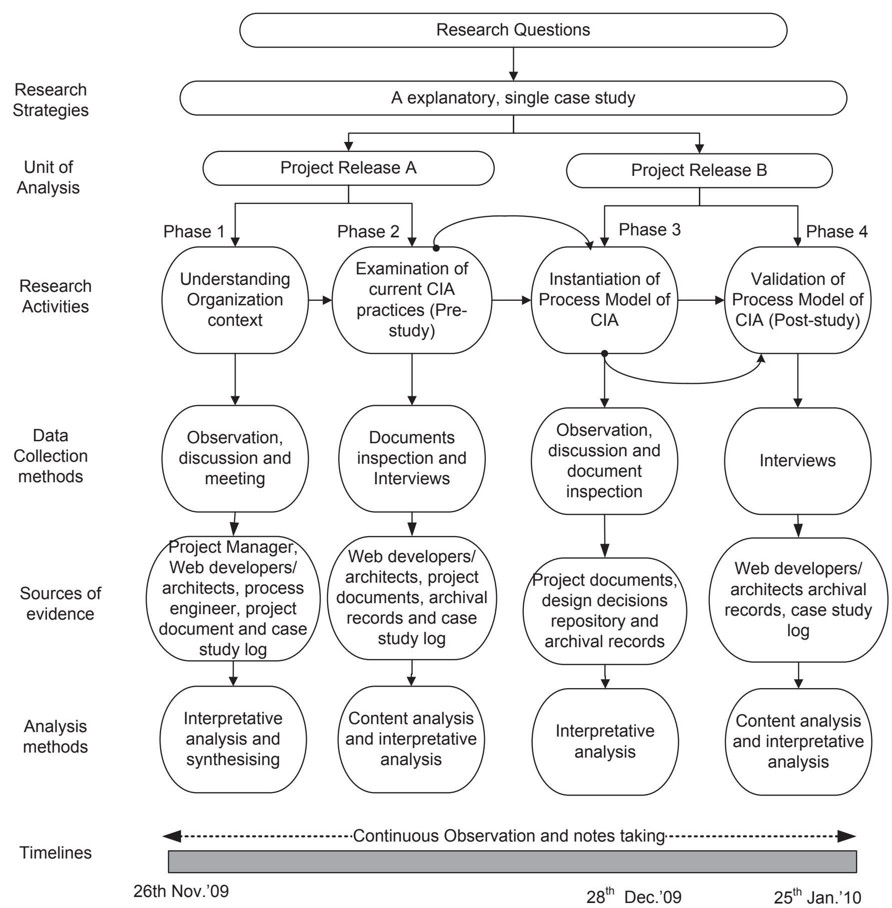


Figure 6.1: The four phases of case study design to validate the process model of CIA

During meetings and discussion, the questions were mainly related to the organisational context, the characteristics of Web system project, features of the Web system project, two project releases, software processes adopted for project development, sets of technology used during system development, size of the project, and organisation wide software development processes. The participants at those meetings and discussions were project managers, Web developers/architects and the process engineer. Sources of data used in this phase included project and process documents, and information gained from meeting participants. The evidence gathered from the meetings, discussions and observation was recorded in the case study log. Synthesization and interpretative analysis were used as data analysis methods in Phase 1. Indeed, information gathered from this phase increased the researcher's knowledge of the case study setting and the characteristics of the Web system project that was being developed.

6.5.2 Phase 2: Examining current CIA approaches/practices (pre-study Analysis)

The purpose of this phase was to study the change impact analysis results during *Release_A* by employing current approaches/practices and getting the insights of the Web developers/architects on adopting current CIA approaches/practices. The examination of current CIA approaches/practices, archival records and document inspections gave insight into the CIA practices already established and followed at WDS. Besides documents inspections, a set of pre-study interview questions were used in this phase to collect data. Content analysis and descriptive analysis were used as data analysis methods in this phase. The findings from this phase were used as a motivation to adopt the process model of CIA in *Release_B*.

6.5.3 Phase 3: Instantiation of process model of CIA

In Phase 3, the process model of CIA is instantiated and introduced at WDS for *Release_B*. Prior to instantiation of the process model and based on observation and document inspections, the researcher along with the process engineer/QA manager took active roles in understanding the way the process model of CIA could be instantiated and adopted into the normal development activities of the WDS's organisation. The main source of evidence used in this phase was project documents and archival records. Observation, discussions and document inspections were used as data collection methods, whereas descriptive analysis was used as a data analysis method.

6.5.4 Phase 4: Validation of process model of CIA

In this phase the main activity was related to examining the proposed change request and study change impact analysis results in *Release_B*. Data collection methods used were similar to the methods used in Phase 2 along with post-study interviews, observation and discussions. In addition to other data sources similar to Phase 2, design decisions repository were included as a source of evidence and the process model of CIA was adopted to identity change impacts in *Release_B*. The data analysed from post-study interview questions was mainly used for the validation of the process model of CIA and to help address the research questions.

6.6 Data Collection

For the relevance and completeness of data, we have taken into consideration three main principles of case study data collection as described by Gillham and Yin (2000, 2003).

- Use multiple sources of data – it is highly recommended to use multiple sources of evidence for a good case study. The use of multiple sources of data is reported to increase the robustness of the result and strengthens the validity and reliability of the study (Neuman, 2000). Therefore, multiple sources of data were used in this case study. Indeed, multiple sources of data allowed the triangulation of evidences and thus improved both the reliability and validity of data and findings. Multiple sources of data included interview participants and interview findings, project documents, design decisions repository and notes taken during meetings.
- Create a case study database – it is important that during the case study relevant data should be documented and organised. Data collected in this study (from interviews, document inspections, observation and meetings) were organised according to project releases in the form of a case study log.
- Maintain a chain of evidence – it was also considered important to link all case study evidences in the form of a chain of evidence. The links should be traceable from the initial research questions, data collected, to conclusion drawn, or from conclusions back to initial research questions. Hence, this case study also adopted the chain of evidences approach. All evidences collected and analysed from four phases of case study were synthesised, integrated, and built up as a chain of evidence to be used in drawing conclusions. Other evidences were also extracted from the case study log.

The primary data collection method used in the case study was interviews. Besides that, document inspection, observation and informal discussions or meetings were also used as data collection methods. For interviews, we collected data from two sources, the first was pre-study interviews and the second was post-study interviews. We have also used multiple sources of data including project documents, design decisions repository, notes from meetings, observation, archival records and interview findings. An overview of the data collection method and multiple sources of data are described in the following sections.

6.6.1 Interviews

Interviews enabled us to inquire to the practitioners directly about their thoughts and subsequently provided the freedom for practitioners to describe in detail their opinions and beliefs related to our research questions (Nachmias and Nachmias, 2007, Patton, 2002). Interviews were used both for pre-study and post-study (Phase 2 and Phase 4) and include (1) open-ended questions and (2) close-ended questions. Open-ended questions were characterised as occasional conversations. The questions and the answer choices had not been predetermined. The purpose of these questions was to ask Web developers/architects directly about their thoughts and opinions and provided them the freedom to describe and reflect in detail their views and beliefs relevant to the objective of the case study. However, close-ended questions were mainly developed to give feedback based on the Web developers/architects experience.

In-depth interviews were used with the aim of getting a deeper experience and perspective from Web developers/architects while adopting both old and new approaches of CIA in their Web system projects. However, keeping in view the validation of the process model of CIA in post-study interviews, most questions were focused on gaining an understanding of how the process model of CIA supports early identification of change impacts in Web systems. Indeed, face-to-face interviews constituted one of the most important and valuable sources of information.

In the case study, these interviews were initiated with a primer i.e. a short section of simple questions with the purpose of making the individual comfortable with the interview situation. A researcher and the interviewed person were alone in a room for thirty to ninety minutes; the researcher took notes as well as recorded the interview after gaining permission. During the priming, full anonymity was guaranteed. The participants were Web developers/architects. The questionnaire used for these interviews were peer reviewed in advance for the purpose of clarity. Most questions were developed so as not to lead the participants. The participants were asked not to discuss the interviews with other project participants in order to avoid learning effects.

6.6.2 Document inspections, direct observations and discussions /meetings

Extensive inspections on project documents and archival records were conducted for both Web system project releases. Additionally, direct observation was also applied during meetings and was a useful technique to gain a much richer and detailed data than other data collection methods such as questionnaires. Further, informal discussions/meetings were also used as data collection methods. On some occasions after the meeting or document inspections, the researcher identified issues or questions that needed to be answered. The researcher then initiated informal discussions or meetings to discuss those issues. These informal discussions or meetings were not recorded; instead, the researcher took extensive notes or wrote the results of the interviews into the case study log immediately after the discussion or meeting was completed. Document inspections were used in the fieldwork conducted in this case study, together with other methods such as interviews, direct observations and discussions/meetings, which provided face-to-face contact with the social actors in order to explore and probe responses. These data collection methods necessitated the collection of a large amount of rich qualitative data from a number of data sources in order to address the complexity of the context studied. In the following section we will describe the multiple sources of data used in our case study.

6.6.3 Project documents

In this case study, the WDS organisation classified the documents as either project related or organisational related. Project related documents were included. e.g. business processes specifications, architecture requirements specifications, change request forms and architecture design specifications. Organisational related documents included the organisation repository, organisation team structures, and different processes and procedures to carry out system development. However, in this case study, mostly project related documents were analysed; and organisational related documents were analysed only if needed as supportive documentation.

6.6.4 Design decisions repository

Design decisions and other related information were recorded in repository by Web developers/architects in response to design modification and the decisions taken to address those design changes. The design decisions repository was maintained through Archium Tool (Jansen et al., 2007) and design decisions and related information were retrieved from this repository.

6.6.5 Meeting observations

Bratthall and Jorgensen (2002) reported that a number of meetings such as project weekly status meetings more important to gain project updates. For this case study, project weekly meetings focused on project planning/tracking/re-planning and modification issues, but to some extent, technical issues were also discussed such as business process changes and approaches to address these changes. The participants at these meetings were mostly the project manager, designer/architect, developers, tester and sometimes, technical experts. Another type of meeting was the software process engineering group (SPEG) meeting. SPEG meetings primarily focused on the process improvement program. The participants at SPEG meetings were mostly managers from organisations units and process engineers/QA manager. The process group was usually involved in tracking, screening, installing, and evaluating new approaches and technology that could improve the system development practices in the organisation. In both of these meetings, the researcher was remained completely passive and served as a non-participatory observer. Non-participatory observation was useful in obtaining first hand understanding of how meaning emerges through talk and collective action, and how understandings and interpretations change (Emerson et al., 1995) during meetings. Non-participatory observation during meetings provided an additional opportunity for collecting data along with other data collection methods.

6.6.6 Archival records

Archival records refer to documents from different development phases and other organisational data (Runeson and Host, 2009). In this case study, archival records included previously collected measurements data, previous change requests and impact analysis results and previous versions of business processes specifications and architecture design specifications and project development schedules.

Additionally, we maintained case study evidences throughout the investigation in a case study log as recommended by Gillham (2000) and Runeson and Host (2009). Indeed, maintaining case study evidences is reported as a crucial factor in the process of describing the context and outcome of a case study (Gillham, 2000). In the case study log we included two main entities as recommended by Gillham (2000). The first was 'evidence' that could be something researcher observes, such as important comments made by the participants in meeting, meeting resolution and a discussion topic heard during a meeting. These evidences were indeed fragmented but they occurred during the case study period. The second is 'personal notes' that were related to insights, ideas, someone the researcher needed to contact or activities to be done. In this case study, we have used a manual kind of case study log as recommended by Gillham (2000) and Runeson and Host (2009). The case study log further facilitated us during writing the case study report.

6.7 Data Analysis

Analysing the amount and variety of data collected is one aspect of case study methodology that has not been well developed in the literature (Yin, 2003). The researcher needs to develop a general data analysis strategy as part of case study design (Yin, 2003). The analytic strategy should indicate what to analyse and for what reason (why), so as to ensure that data collection is appropriate as well as to support the evidence to be analysed. In this case study, both qualitative and quantitative data analysis approaches were used. For quantitative data such as responses from closed-ended question, we tabulated a spread sheet and analysed with basic statistical methods such as frequency analysis. For qualitative data, content analysis was used to determine the presence of certain words or concepts within texts (Krippendorff, 2004). Content analysis is widely used in various research fields, and can be applied to examine written or recorded communication. Therefore, in order to analyse responses from open-ended questions and project documents, content analysis was used as part of the data analysis process.

Additionally, interpretive analysis was used as a data analysis method. Interpretive analysis advocates a relativistic understanding of the phenomenon being studied (Orlikowski and Baroudi, 1991). During interpretive analysis, the researcher synthesized the data by involving both induction and interpretation, and therefore avoided specifying concepts in advance of the synthesis. Additionally while performing interpretative analysis, the researcher perused meaning as subjective whereas knowledge as a social construction (Walsham, 1993), with the intention that the results of analysis would guide the subsequent analysis of data.

Triangulation is a means of combining multiple sources of data, theories, methods, and observers to improve the validity of the research or to obtain greater accuracy (Patton, 2002). For a single case study, like ours, triangulation is also regarded as a means of looking at a single complex phenomenon under investigation in order to better understand the context in its multifaceted form. Patton (2002) distinguished four types of triangulation, that is, the triangulation: *of methods* (methodological triangulation), *of data sources* (data triangulation), *among different analysts/evaluators* (investigators triangulation), *and of perspectives on the same data set* (theory triangulation). In this case study we used triangulation of data sources as well as triangulation of methods.

We focused on the triangulation of data sources due to two reasons (1) to increase confidence in a finding by identifying confirming or contradictory evidences and (2) to make findings that could not be made using a single data source (Bratthall and Jorgensen, 2002). Triangulation of data sources includes findings from pre-study and post-study interviews, project documents, archival records, and observations.

For the purpose of methods triangulation, both quantitative and qualitative data analysis methods were used with the emphasis on qualitative methods. Miles and Huberman (1994) point out that making linkage between qualitative and quantitative data is essential during a case study. For example, on a few occasions, our case study qualitative data helped us to interpret, clarify, and illustrate quantitative findings (Miles and Huberman, 1994).

6.8 Reporting

Reporting during case study is an important part of the case study research methodology. The case study report communicates the findings of the study and these reports are also considered as a main source of information for judging the quality of the study. Reports may have different audiences, such as peer researchers, research sponsors, and industry practitioners (Yin 2003). This may lead to the need of writing different reports for difference audiences. In this case study, we focused on reports where peer researchers are the main audience, and these reports includes journal or conference articles and possibly accompanying technical reports.

6.9 Research Reliability, Validity, and Generalisability

Case study research mostly deals with a complex phenomenon and generally involves multiple sources of data and tends to produce large amounts of data for analysis. Keeping in view these factors, the reliability and validity of a case study as a research method are mostly considered as primary concerns.

In the past researchers have argued that case studies are extremely limited in providing conclusions that can be generalised, given that the case study is based on a single set of circumstances. However over time the argument in relation to a case study has regularly been rejected (Flyvbjerg, 2007, Runeson and Host, 2009), stating instead that case studies are acceptable and can provide very meaningful information provided they are designed and performed with the appropriate degrees of rigor and structure (Hamel et al., 1993). Additionally, case study has also been criticized for being of less value and being biased by researchers (Runeson and Host, 2009). This critique can be addressed by applying proper research methodology practices as well as reconsidering that knowledge is more than statistical significance (Flyvbjerg, 2007, Lee, 1989). Another issue is the ethical consideration in association with the researcher's personal integrity and sensitivity while conducting the case study (Singer and Vinson, 2002).The following sections show how the issues of reliability and validity are handled for this case study.

6.9.1 Reliability

In the case study research, the issue of reliability is associated with the consistency of case study results (Burns, 2000, Yin, 2003). In this case study, we have addressed this issue by adopting case study protocols (Yin, 2003), particularly during data collection and analysis. The researcher was engaged in the data gathering process at the case study site and both different research methods and sources of data were used to ensure the accuracy of the findings. Furthermore, case study design is documented in detail and the sequence of case study phases and activities are clearly described in this study.

6.9.2 Construct validity

Construct validity is concerned with the use of an insufficient operational set of measures and potential researcher subjectivity (Burns, 2000). In order to address the construct validity, this case study adopted two important ways as proposed by Yin (2003), to counteract both the problems of insufficient operational set of measures and potential researcher subjectivity. These two important ways include (i) using multiple sources of evidence (as described in Section 6.7) and (ii) establishing a chain of evidence that links case study evidences together.

6.9.3 Internal validity

Internal validity is concerned with the credibility of the findings and how well the findings match reality. This case study addresses internal validity issues by considering strategies including triangulation, re-checking with participants, and observation. With regard to the Patton's triangulation types (Patton, 2002), triangulations used in this case study are as follows:

1. *Triangulation of methods*, which is related to the use of different methods for data analysis. Both qualitative and quantitative methods were utilised to examine the data gathered during the instantiation of the process model of CIA in an industrial setting.
2. *Triangulation of data sources*, in which different data sources are utilised within the same study. In this case study, we used multiple data sources. i.e. documents, interviews, observations, archival records and a case study log to improve the accuracy of findings.

Additionally, during case study visits, the researcher initiated and maintained an active corroboration on the interpretation of data with the case study participants who provided the data.

6.9.4 External validity (generalisability)

External validity is related to the researcher's concerns of whether the case study findings can be generalised or other researchers can apply the findings to their own research. Keeping in view the external validity of case study, criticism is often made that a single-case study provides little evidence for scientific generalisation. The case study adopted in this research is classified as a single-case study and the investigation was carried out in a particular organisations. Therefore, the results are only generalisable to organisations with similar structures and business domain and, the findings may not necessarily be applicable to other organisations. It means that the case study findings can be claimed to be applicable to organisations with similar demographics. Although it is a single-case study, the methods and procedures used are described in sufficient and adequate detail, and clearly documented; therefore, they can be adopted and repeated by other researchers and practitioners in their organisations.

Additionally, it is believed that the findings from this study cannot be generalised to all populations i.e. to all Web systems development organisation, but these findings contribute significantly to the current practice and theory.

6.9.5 Ethical considerations

Before conducting this research, both a written permission from the University of Technology, Sydney Human Research Ethics Committee (HREC) and a written recommendation from the WDS management were obtained. This case study research was compiled in accordance with HREC guidelines.

Additionally, before commencing data collection, participants were invited to participate in this research. They were informed verbally about the research and the purposes of the investigation were explained. A consent form was used to obtain the participants' written consent. In the consent form it clearly stated that participants are assured of the confidentiality of their responses either from interviews or discussions and of their freedom to withdraw at any stage of the research without giving a reason (see Appendix G and H). It was made sure that participants agreed that the data collected would only be used for research purposes; and the results would be published in this thesis, conferences, and journal articles without revealing the confidential issues of individuals, projects, and specific organisational practices. The researcher signed a non-disclosure agreement with WDS both to reinforce ethical consideration and to comply with organisational requirements as needed.

6.10 Chapter Summary

This chapter has described the design of case study methodology adopted for the validation of the process model of CIA.This chapter has also described research design protocols while addressing the questions of 'why' and 'how' case study research was adopted so as to validate the process model of CIA in an industrial setting. The case study design comprises of four phases, data collection and analysis methods; and the case study reporting was developed in accordance to the research design protocol as well as the guidelines provided from the research literature. The multiple sources of data collection and multiple data analysis methods were used and discussed in detail.

In case study design, the sequence of activities in each phases has been structured and described in detail throughout the case study. Further, the application of case study design in a commercial organisation setting has been described with a careful composition that is associated with the research questions and subsequently the research objectives.

This case study is classified as an explanatory single-case study and underpinning issues related to validity and reliability were considered. Further, approaches to address issues of validity and reliability were also examined. The application of case study activities which cover the investigation of two releases of the Web system project and in depth findings from the validation of the process model of CIA, are described in detail in the Chapter 7.

Chapter 7: Case Study Findings

7.1 Introduction

In the previous chapter, we presented and described PMCIA - a process model of CIA, as a possible solution to adopt during Web systems development. The proposed PMCIA aims to support the identification of impacts on architecture design resulting from business processes changes i.e. early identification of change impacts in Web systems, which we have argued, is especially required in Web systems. In this chapter, to validate PMCIA, a case study was carried out to investigate whether or not PMCIA supports early identification of change impacts in Web systems (Mehboob & Khan, 2013).

The case study was carried out across two releases of a Web system project in an organisation. The case study findings serve as indicative proof of concept for the validation of PMCIA. This chapter provides the necessary information about the case study and results from the validation of PMCIA.

In this chapter, we begin by providing, the objective of the case study in Section 7.2 and the case study settings in Section 7.3. We then present the analyses of the findings and results from the validation of PMCIA through the selected case study (comprises of four phases) in the next four sections, Section 7.4 to 7.7. We finish the chapter with some discussions and the summary in section 7.8.

7.2 Case Study Objectives

The objective of the case study is to address the research questions *RQ11* and *RQ12* with regards to the validation of PMCIA. Specifically, we intended to validate whether PMCIA supports Web developers/architects in early identification of change impacts in their Web systems project. Further, our intention was to look at whether the process model of CIA addresses the characteristics of Web systems.

RQ11: Does the process model of CIA address the specific characteristics of Web systems?

RQ12: Does the process model of CIA support Web designers/architects in early identification of change impacts?

Given the case study findings that support to address these two questions, we could argue that if PMCIA enables the identification of impacts on architecture design resulting from business processes changes then we can claim that, indeed, PMCIA (i) addresses the specific characteristics of Web systems and supports (ii) Web developers/architects in early identification of change impacts. In the next section, we briefly cover the case study settings according to the case study design methodology presented in Chapter 6.

7.3 Case Study Settings

For the purpose of validation, the case study investigation was carried out by comparing the usage of two CIA approaches (existing CIA approaches and proposed PMCIA as a new approach) in two different releases of a Web system project. As described in Chapter 6, the case study has been designed by focusing on four phases as shown in Figure 7.1. In the following, each phase of case study design is described in the subsequent sections of this chapter.

1. Section 7.4 describes the organisational context of the case study site (*Phase_1*)
2. Section 7.5 examines CIA practices from *Project Release_A,* discusses the findings and presents pre-study interview results in relation to existing CIA approaches adopted in the organisation (*Phase_2*)
3. Section 7.6 presents findings from the instantiation of PMCIA during *Project Release_B* (*Phase_3*)
4. Section 7.8 presents post-study interview results in relation to the validation of PMCIA (*Phase_4*)

7.4 Phase_1: Understanding Organisational Context

This phase was the initial phase of the case study design that describes the necessary information about the selected organisation and its context. The case study was carried out in a real project setting and ran from late November 2009 until late January 2010, for eight weeks. The purpose of this phase was to understand the organisational context by investigating Web systems project, change management and other supporting system

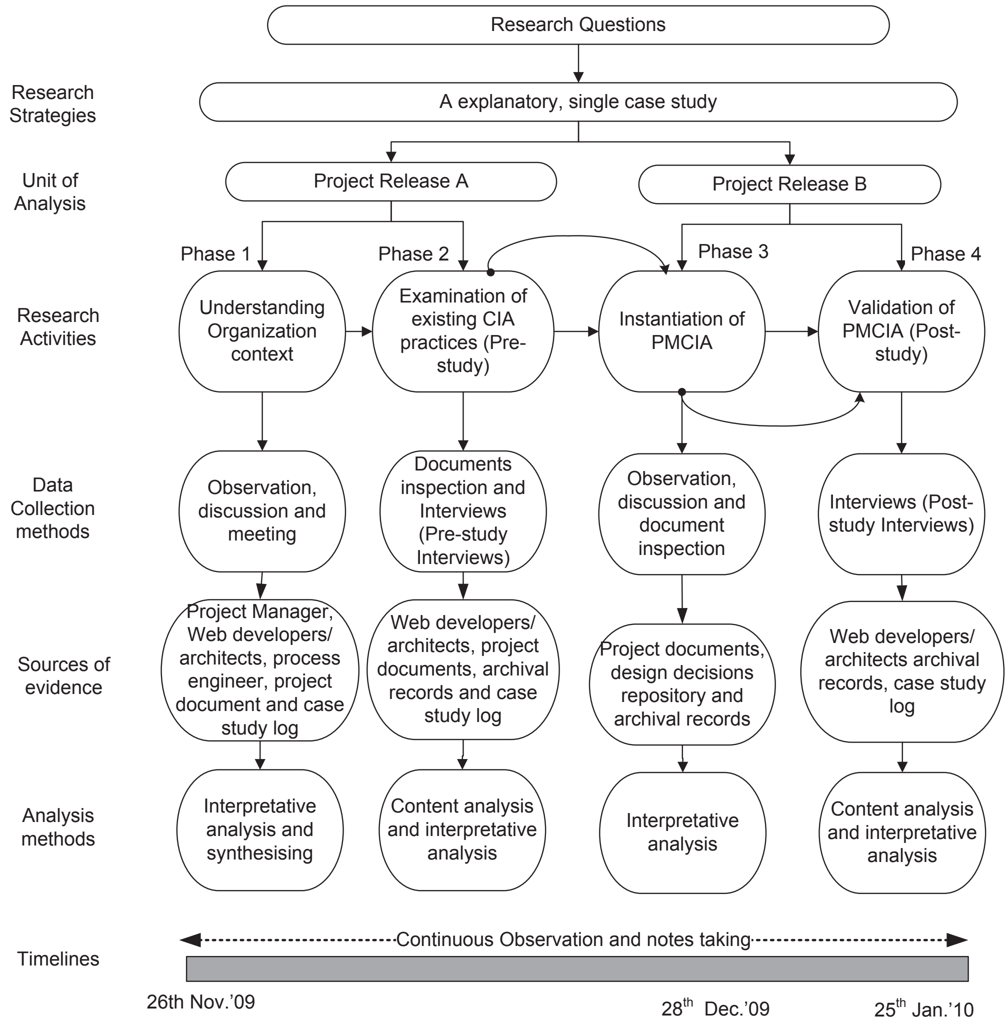


Figure 7.1: The four phases of case study design to validate PMCIA

development activities. An adequate understanding of organisational context allowed us to ensure that the organisation was selected according to the criteria described in Section 6.3.

7.4.1 Phase_1: Data collection and analysis

In Phase 1, a number of data collection methods were used that included discussion, document inspections, non-participatory observations and meetings as described in Section 6.4. To understand the organisational context, document inspections and initial discussions were conducted with senior management (including the project manager and quality assurance manager) in the first few days of the case study. During these discussions and document inspections (both organisational documents and project documents), the researcher took extensive notes and documented them in a case study log. Given that the researcher has seven years of industrial experience (as a quality assurance analyst, project manager and process manager) of executing successful process improvement programs in many ISO 9001:2000 and CMMI level 3 assessed software development organisations, it is conceivable that the researcher has adequate expertise for inspecting both organisational and project related documents. These inspections further led to follow-up questions (if required) that were discussed in subsequent meetings with managers, Web developers/architects and the process engineer.

Further, the researcher attended weekly project meetings where the project team discussed issues and the progress of the project. Moreover, the researcher attended software engineering process group (SEPG) meetings where process engineering and process improvement issues were discussed and process improvement initiatives were planned. In both of these two meetings, the researcher took the role of a non-participatory observer as described in Section 6.6.6. The researcher took notes during all these meetings. In Phase 1, the researcher attended one project meeting and one SEPG meeting. However, the researcher continued attending these two meetings as long as these meetings came up during the course of the case study. The purpose of attending these two meetings was to increase understanding of the organisational context, and to gain an understanding of those organisational and project aspects that were possibly not conveyed or discussed by managers, Web developers/architects and process engineers. The researcher synthesised the collected data and used interpretative analysis as a data analysis method as described in Section 6.6.

7.4.2 Phase_1: Findings

In the following sub-sections we present the necessary information about the organisation's background, the project overview and its releases.

7.4.2.1 Organisation's background

This case study was carried out in Web Development Solutions (WDS)[6], an organisation located in the Asia region. WSD is an international multi-site set-up with

[6] The name is fictitious to preserve anonymity

headquarters in the United State of America (USA) and sub-divisions located in the United Arab Emirates (UAE) and Pakistan. The customers of WDS are scattered across the world. However, most Web system projects have been implemented for USA and UAE customers. WDS conducts business in the healthcare sector and mainly provides clinical management solutions.

The WDS organisation is ISO 9001:2000 certified and SEI CMMI® maturity level 2 assessed. Most of the system development practices such as requirement management, project planning and control, and quality assurance were aligned with the requirements of CMMI®. As part of continuous process improvement programs, the process engineering group started working towards the external assessment of SEI CMMI® maturity level 3 and had taken a number of initiatives towards process improvement programs.

The project development strategy and project requirements are communicated from the Business Development sub-division (where the business development groups and consultants were located) to the System Development sub-division (where system development groups were located). The Business Development sub-division is located in the USA whereas one System Development group is located in the UAE and one System Development group in Pakistan. Customer business requirements are captured by the Business Development group. These customer requirements are then communicated to the system development group via business processes specifications for implementation of a specific project. Direct communications between the system development group and the customers are minimal and these communications are mainly done through the Business Development group.

A change management procedure was in place to support requirement managements. However, the general practice of change impact analysis at architecture design was primarily based on peer reviews. At some instances, traceability links were maintained from business processes to architecture requirements specification or further to source codes through RequistPro® but surprisingly traceability links were not maintained from business processes to architecture design. Instead, once a change in business process was approved then Web developers/architects reviewed the architecture design to address those business processes changes. As a result of review they recommended a set of changes for architecture design (if required). Mostly Web developers/architects used their gut feelings, previous experiences and intuition to recommend impacts of changes on architecture design. Further, these recommended changes were recorded in a design review form and then linked (through Microsoft Visual Source Safe - MS-VSS) with a respective version of architecture design. The decisions made for architecture design changes were stored in a separate document called the Decision Analysis and Resolution[7] (DAR) report. Each DAR report was then linked (through MS-VSS) to the respective change(s) made at architecture design. Additionally, all the information about design change decisions were documented and maintained in the Archium tool (Jansen et al., 2007).

As described in Chapter 2, there are different views of architecture design including functional architecture design, system architecture design, interface design and information architecture design, etc. In this case study, we have focused only on a specific view of architecture i.e. information architecture design due to its utmost importance in Web systems development as described in Section 2.5.2. In the rest of this chapter, we thus will use the term information architecture design and architecture design interchangeably.

7.4.2.2 Web system project overview

The selected project, for this case study, was a web-based clinical management system. In brief, this clinical management system serves out-patient needs, manages patients' health data and other related clinical routine activities. This system allows patients to perform search desired information, requests for appointment, submit queries, and input their medical records. Clinical staff makes informed decisions based on the information provided from the patient and advises them accordingly. Further, this system facilitates the communication between clinical staff and patients while providing most up-to-date data at the point of care. The clinical management system also provides a secure access both for patients and clinical staff to manage health information. Patients can view certain parts of their electronic medical record and thus empowers them to take a more active interest in their health care.

This Web-based clinical management system has the following important features.

[7] Decision analysis and resolution procedure is a part of CMMI® level 3 organisational procedural documentation.

Quality of healthcare

- Quick and easy selection of the appropriate, categorised, and authorised medications with instant prescriptions for doctors.
- Warning messages for medication allergies.
- Computerized lab information.
- Up to date, practically categorised codes for easy selection and integration of billing.

Staff productivity

- Health calendar with reminders to track and manage health-related appointments and activities.
- Secure patient messaging and provider message connectivity solutions.
- Task setting and task assignment with task sending options.

Administration

- Setting customised provider schedules, patient problems and allergies.
- Favourite lists of codes for each provider.
- Customisable templates designed for lab entry.
- Administering and managing resources like Users, Sites and Locations.

Operating efficiency and workflow

- Increased patient flow and peak practice workflow performance.
- Efficiently manage and analyse the clinical process.
- Efficiently manage workflow with easily accessible patient charts and tasks such as orders, prescriptions and documentation.
- Accessibility to one source of centralised, patient information creating a single, comprehensive data source.
- Medical transcription interfacing with outsource or in-house transcription services

When the case study investigation started, two project releases were under development at different stages, according to the project plan. In this case study, we refer to these two releases as *Project Release_A* and *Project Release_B*. The first release, *Project Release_A*, was in the later stages of the development lifecycle (user acceptance testing stage and deployment stage), and the second release, *Project Release_B* has just commenced as the next increment to address proposed business processes changes. Although the first release was already near completion we were still able to examine *Project Release_A* and the related data of this release for the following reasons:

1. Business processes changes and their resulting changes in other system artefacts were documented and recorded in the project repository.
2. A formal change management procedure was established and followed to manage changes in system artefacts.
3. Most of the project team members involved in *Project Release_A* were also working on *Project Release_B*.

Thus sufficient documentation as well as project team members were available and provided us with essential information in relation to *Project Release_A*.

7.4.2.3 Project release_A

Project Release_A focused on business processes changes by implementing the modified functionalities of the system. The aim of *Project Release_A* was to maintain the customer base by keeping Web-based clinical management systems competitive. Subsequently, a few modules were developed during *Project Release_A* to provide the system performance and scalability while supporting both decision making and a high-volume of transaction processing for customers.

7.4.2.4 Project release_B

Project Release_B focused on business processes changes by implementing modified functionalities of the system. The aim of *Project Release_A* was to develop a collaborative environment of the clinical management system in order to gain access with other clinics in a specific region anywhere and anytime. These changes had already been scheduled in the project plan for *Project Release_B*. Besides the strategic direction, the reason for this change was to improve the usability of the system while providing accessibility to a centralised patient information data source. During *Project Release_B, a* few modules were developed to access the information from a collaborative data source (handling data across all the clinics in a specific region) and enable more informed decisions.

7.5 Phase_2: Change Impact Analysis-Project Release_A

This section describes the research activities conducted during Phase 2 as shown in Figure 7.1. The activities in Phase 2 included (i) examining the data in relation to change impacts while adopting existing CIA approaches during *Project Release_A* and (ii) conducting pre-study interviews with Web developers/architects in relation to existing CIA approaches. The aim of examining change impact data was to understand the outcome achieved by employing existing CIA approaches in *Project Release_A*. Whereas the aim of pre-study interviews was to understand the participant's perspectives of existing CIA approaches (while adopting it) during *Project Release_A*. Indeed, the findings from Phase 2 provided valuable information for the case study.

7.5.1 Phase_2: Data collection and analysis methods

In order to execute research activities in Phase 2, a number of data collection and analysis methods were employed. Both, qualitative and quantitative data were collected in Phase 2. The collected data was related to the changes made in business processes, results of impact analysis performed at architecture design to address business processes changes, and the changes made to different system artefacts to address business processes changes.

7.5.1.1 Data collection

The sources of evidence used in this phase included project documents such as change request forms, business processes specifications, architecture design specifications, project archival records, project measurement and analysis data, IBM RequsitePro® tool records and projects weekly meeting minutes. Organisational process documents included change management procedures, measurement and analysis procedures and its records. The details of change requests in *Project Release_A* were stored in a tool called IBM RequsitePro®, and therefore the data stored in this tool's database was also considered an important source of evidence. Other sources of evidence were project team members including process engineers and Web developers/architects during pre-study interviews.

Similar to Phase 1, data collection methods used in this phase were documents and tool inspections, non-participatory observations in meetings and discussions. On some occasions after weekly project meetings, SEPG meetings and document inspections, the researcher identified issues or questions that needed to be answered. The researcher then initiated discussions to address those issues with relevant project team members. These discussions were not recorded; instead, the researcher took extensive notes or wrote the results of the discussion in the case study log immediately after the discussion was completed. The data collected during Phase 2 are as follows:

1. Total number of changes made in business processes,
2. Total number of identified change impacts at architecture design to address business processes changes (using existing CIA approaches),
3. Total number of identified change impacts at various system artefacts (detailed design, code) to address business processes changes (using existing CIA approaches)

Further, pre-study interviews were also used as a data collection method to understand Web developers/architects perspectives in relation to existing CIA approaches. The interview included a set of questions, both open-ended and close-ended, which were reviewed and refined through several iterations before being given to participants. The questionnaire for pre-study interviews is included in Appendix G. The participants of pre-study interviews were Web developers/architects from WSD who had been involved in architecture design reviews, architecture design modification and had experience in analysing change impacts on architecture design. In the following, we describe the steps taken for interviews.

- Step 1: First step for the interview was the selection of participants. Based on the selection criteria specified in Chapter 6, the names of four Web developers (as reviewer) were collected from design review forms and included those who had proposed a few changes for architecture design. Additionally, the names of six Web developers were collected from the revision history of architecture design specifications and included those who had made changes at architecture design based on the design reviewer's comments. Finally, nine Web developers and one project manager were selected as interview participants based both on their availability and recommendation of the development manager. Most of participants remained the same both for pre-study and post-study interviews.

- Step 2: An invitation e-mail along with the interview schedule and interview questionnaire was sent to all participants, both for pre-study and post-study interviews. A consent form was also signed by the participants before conducting the interviews.
- Step 3: As per the schedule, face-to-face interviews were conducted and audio recorded along with notes taken by the researcher.
- Step 4: At the end of the case study, recorded interviews were transcribed. Further, content analysis and basic statistical analysis were performed by the researcher on the transcribed interview data.

As per pre-study interview details provided in Chapter 6, pre-study interviews were conducted on site in the third week of case study. The researcher conducted all interviews face-to-face at the organisation premises and probes were used to let participants expand and clarify their responses. Additionally, at the beginning of the interview, descriptions about early identification of change impacts, information architecture and other technical aspects were explained to participants. All interviews were audio recorded by the researcher himself. All audio recorded interviews were listened again and again and every possible effort was made to convert correctly each recorded interview into interview transcripts. For the purpose of confidentiality, all the interview transcripts (including questionnaire responses) were given a numerical identification code, based on the sequence they were received.

7.5.1.2 Data analysis

Data collected during Phase 2 was analysed both qualitatively and quantitatively. For qualitative data, the content analysis mainly used both project documents and pre-study interview transcripts as most of the data analysed and collected were in the form of textual description. Additionally, interpretive analysis was also used as a data analysis method.

For quantitative data, basic statistical analysis techniques such as frequency analysis were used while analysing both data related to changes made in *Project Release_A* and data collected from close ended interview questions.

7.5.2 Phase_2: Case study findings from project release_A

In Phase 2 of the case study, it was intended to gather valuable information about business processes changes and the results of change impact analysis during *Project Release_A*. The findings from Phase 2 motivated us for instantiation of PMCIA in *Project Release_B* and also served as inputs for Phase 3, as shown in Figure 7.1. In the following sections we present important findings related to business processes changes and impact of changes identified at architecture design and at later stages of system development (architecture detailed design, code) which address those business processes changes.

7.5.2.1 Business processes changes

For *Project Release_A*, a total of eight change requests were collected during the first four weeks. Four out of eight change requests were related to changes in business processes. The rest of the change requests were related to performance enhancement, interface issues and user acceptance testing (UAT) issues. All these changes were scheduled in the project plan to be implemented during *Project Release_A*. Percentages of each type of changes out of total change requests during *Project Release_A* are illustrated in Figure 7.2. Since we are interested in business processes changes and their possible impacts on other system artefacts, we analysed only business processes changes in this case study.

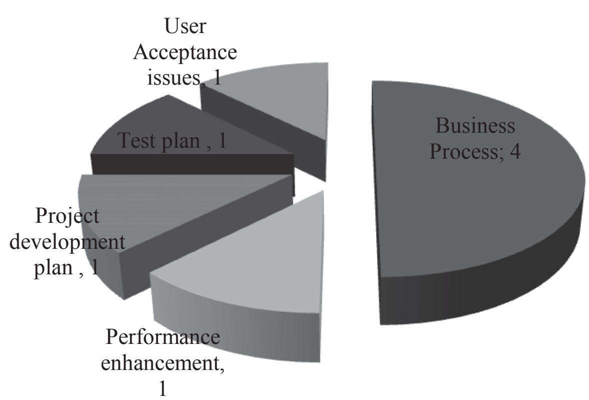


Figure 7.2: Percentage of different types of changes during Project Release_A

7.5.2.2 Identified change impacts at architecture design

For this case study, changes impacts in architecture design were retrieved from change history data of architecture design specifications. This change history data comprises of the complete history of what changes were made, where changes were made and by whom changes were suggested and made. Further, this change history data was stored using a version control tool called Microsoft Visual SourceSafe 6.0® (MS-VSS). General practice to identify change impact at architecture design was described in Section 7.4.2.1. Web developers primarily rely on previous experience, gut feeling, expert judgement and a lot of intuition while analysing impacts on architecture design resulting from business processes changes. The outcomes of adopted CIA approaches or practice are described in the next paragraph.

As described in the previous section there were eight changes proposed at the start of *Project Release_A*. In order to address those changes there were other changes made in various system artefacts as a result of impact analysis. At the end of *Project Release_A*, there were twenty change impacts identified in total to address the business processes changes. Out of a total of twenty change impacts for *Project Release_A,* only two were changes impacts identified at architecture design (to address business processes changes). Whereas six change impacts were identified to detailed design and twelve change impacts were identified to source code to address those eight changes made initially in business processes. It means that most of the change impacts were overlooked at architecture design and indentified very late - only at detailed design and source code. Therefore, two change impacts at architecture design were considered as direct impacts of business processes changes. Whereas a number of impacts as a result

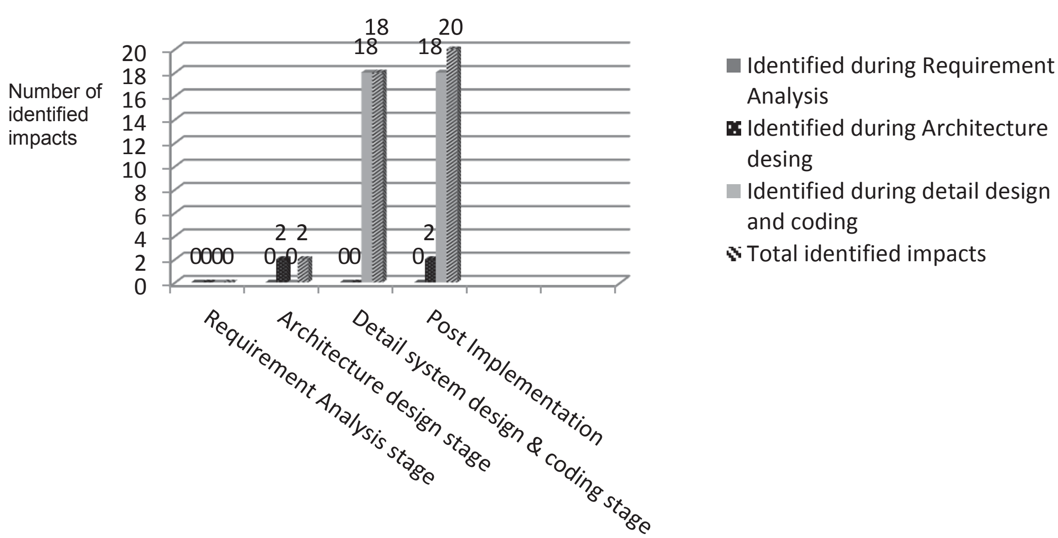


Figure 7.3: Identified impacts at architecture design and later stages of system development to address business processes changes during *Project Release_A*

of business processes changes were overlooked and eighteen out of twenty changes impacts were identified very late once detailed designing and coding had already started. The details of changes impacts identified at architecture design and after architecture design stage (i.e. during detailed designing and coding stages) are provided in Figure 7.3.

7.5.3 Phase_2: Pre-study interview findings and analysis

This section presents the findings from pre-study interviews. Pre-study interviews were conducted to gain insight of Web developers/architects's perspective on existing CIA approaches/practices that were adopted in *Project Release_A*.

7.5.3.1 Participants overview

The total number of selected participants for pre-study interviews was ten; however nine of them remained associated both with *Project Release_A* and *Project Release_B*. The participants' experience in Web systems development varies between four years and six years with a median of five years. A median score for working as Web developers/architects is three years and a median score for working with an organisation (current or previous) is four years. An average number of co-workers on the current Web systems project is fourteen people. All of the respondents have an IT related tertiary qualification. These demographics give us confidence that we gathered data from practitioners who were experienced in Web systems development and especially had experience of change impact analysis on architecture design. The roles of participants in the project at the time of conducting this interview were:

- Nine Web developers/architects
- One Project Manager

7.5.3.2 CIA practices adopted in project release_A

A few questions were focused on understanding the CIA approaches and ways of performing change impact analysis. Participants were asked to rate their understanding of CIA approaches for architecture design. The interview data shows that two participants have '*good*', seven have '*fair*' and one had a '*very good*' understanding of the CIA approach. Furthermore, participants were asked to rank the importance of CIA for architecture design, three participants ranked it as '*important'* and seven ranked it as ‘*very important’*. This was an indication that participants have a good understanding of CIA and considered CIA as an important activity for architecture design.

Further, when participants were asked about ‘how change impacts were analysed’, most participants responded with an informal approach of performing CIA for architecture design. Figure 7.4 indicates that mostly five participants said (a) expert judgement, and considered it highly important whereas three said (b) previous experience and only two said (c) historical data as a possible way to analyse change impacts on architecture design. This grouping of percentile data indicates that most participants perform change impact analysis based on their own judgement, experience and gut feeling instead of any formal approach for change impact analysis.

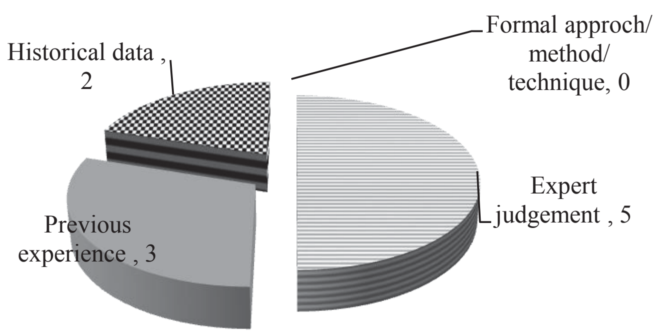


Figure 7.4: Possible ways of performing change impact analysis during *Project Release_A*

Additionally, the findings of the researcher’s observations and inspections described in 7.5.2.2 confirmed the data presented in Figure 7.4. It suggests that existing CIA approaches are not properly formulated and need significant structure and organisation.

Secondly, a set of questions focused on the importance of traceability and dependency analysis, early identification of change impacts and about the approach of CIA. When participants were asked to prioritise on a scale of (1=Very important, 2=Important, 3=Some what important, 4=Not important, 5=Not sure), participants ranked (a) traceability from business processes to information architecture design entities as: eight participants said = very important, two participants said = important. Similarly, participants ranked (b) identification of changes on other parts of architecture design-ripple effects due to changes made at architecture design as: seven participants said = very important, three participants said = important. However, all participants ranked (c) early identification of change impacts as '*very important'*. When participants were asked about (d) structured, consistent and organised way of performing impact analysis, eight participants ranked it as "*important*" and two participants ranked as "*somewhat important*" based on their experience working in process-oriented organisations. Figure 7.5 shows the data in relation to four components of the CIA approach and indicates a very positive feedback towards them.

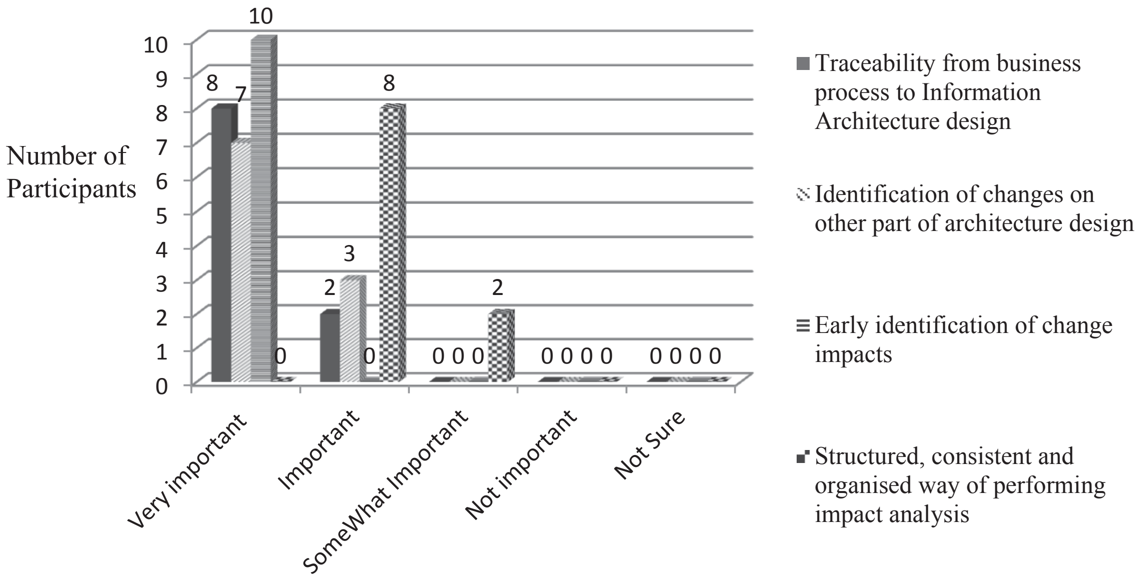


Figure 7.5: Importance of CIA related items

Further to show their agreement and disagreement on the scale of (1=strongly agree, 2=disagree, 3=neutral, 4=disagree, 5=strongly disagree), participants were asked whether existing CIA approaches support *to link business processes and information architecture design entities to maintain traceability between them*. Mostly, participants showed their disagreement (one participant = disagree and nine participants = strongly disagree). This indicates that the traceability from business processes to information architecture design entities was important but the existing CIA approaches did not provide such support.

A third set of questions were related to the availability of information to be used for the purpose of CIA. Mostly participants responded to (a) is sufficient information available to identify the ripple effect at architecture design- as strongly disagree (two participant = agree, one participant = neutral and seven participants = strongly disagree). Participants also responded to (b) is sufficient information available to trace down change impacts to the level of impacted IA area- as strongly disagree (one participant = disagree and nine participants = strongly disagree). However, all participants ranked (c) support to employ design decisions information to investigate the possible dependencies among architecture design entities as '*strongly disagree'*. Figure 7.6 shows the data on the availability of information for the purpose of CIA at architecture design.

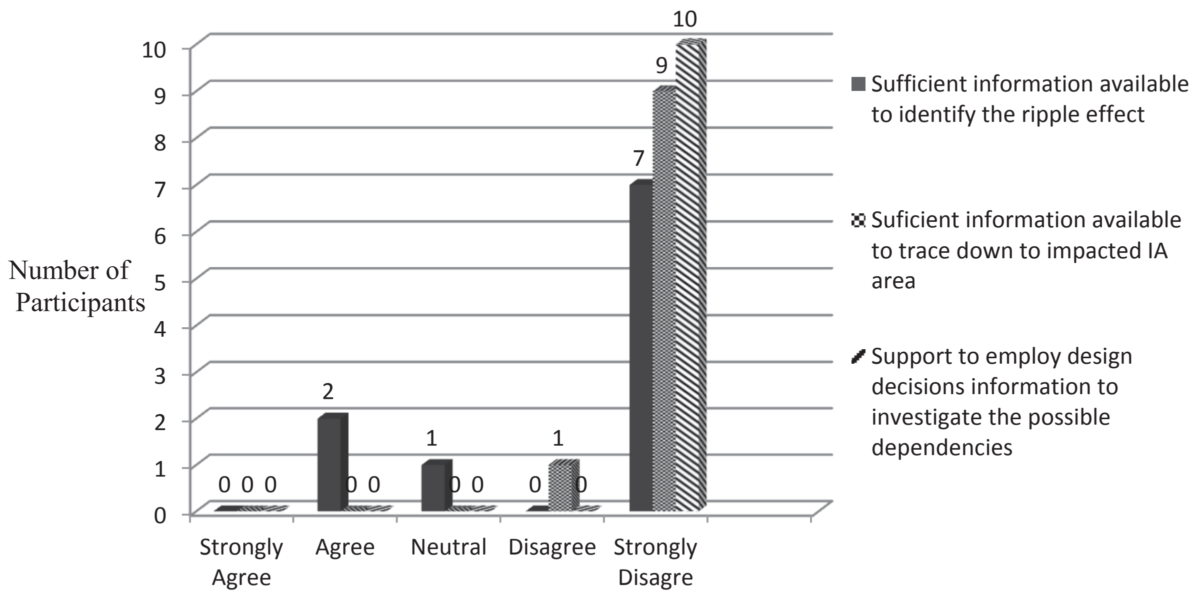


Figure 7.6: Participants agreements on supported aspects of existing CIA approaches

Unavailability of information indicates that existing CIA approaches neither facilitate the tracing down of the impacted IA area nor identify the ripple effect on architecture design. The existing CIA approaches did not extend any support neither for dependency analysis nor employ architecture design information. These findings also confirm the researcher's observation about unavailability of dependency information for CIA (Section 7.5.2.2).

When participants were asked what sources of information are available and being used in relation to existing CIA approaches, eight participants responded to 'traceability information & project documentation', two responded to 'project documentation' and none responded to 'dependency information'. The data indicates that during CIA, participants primarily used project documents and available traceability information while tracing the change impacts from one system artefact[8] to other system artefacts.

[8] The term system artefact has been described in Chapter 2 and a number of examples were given.

However, the use of traceability information limited the identification of change impacts at the level of system artefacts. i.e. breadth of change propagation. Hence, detailed analysis of change impacts was not possible due to the unavailability of required information.

Fourthly, few questions were asked about the process of performing CIA. Most of the participants showed their disagreement for "current CIA approaches being easy to understand" (three participants = neutral and seven participants = disagree). Participants' disagreement was largely true as the details and documentation about CIA approaches were not adequately specified nor clearly documented in organisational procedures. Similarly, most of the participants responded neutrally to "existing CIA approaches are easy to use" (three participants = agree and seven participants = neutral). However, all of the participants responded as '*strongly disagree*' when asked "Does existing CIA approaches provide steps to follow?" Most of the participants showed their disagreement when asked "Does current CIA approaches provide activity description?" (two participants = agree and eight participants = strongly disagree). Additionally, all of the participants showed their disagreement when asked "Does existing CIA approaches provide a feedback mechanism to evaluate the result of impact set after verifying affected architecture design?"

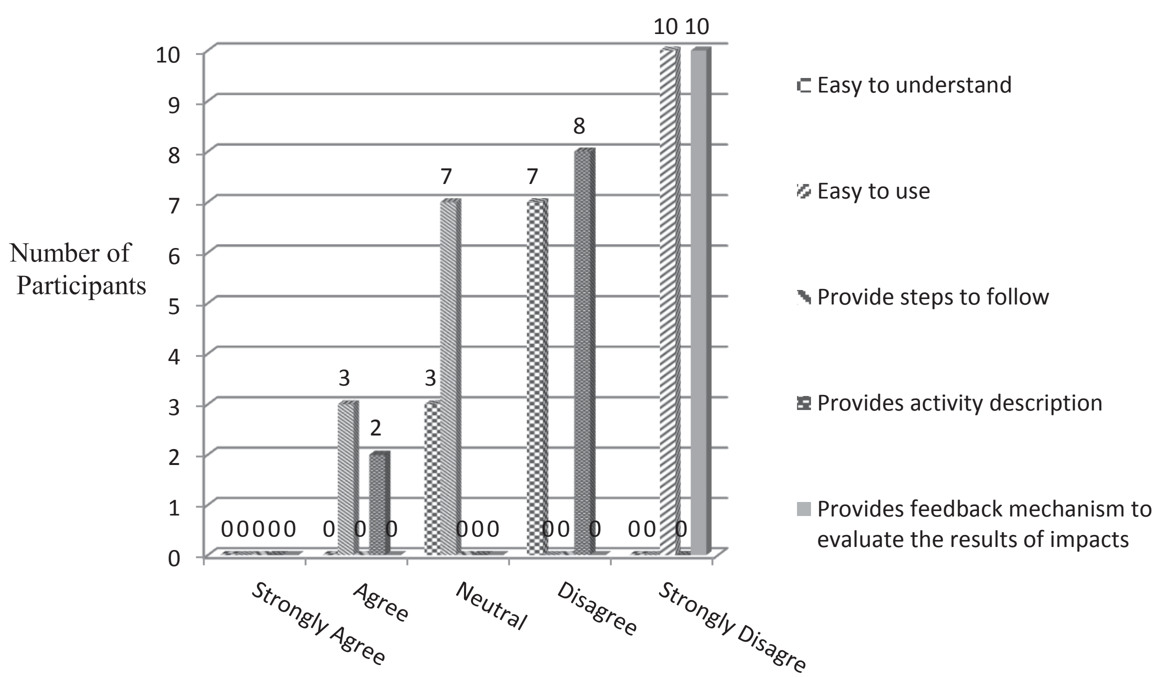


Figure 7.7: Participants perspectives on the process aspects of current CIA approaches

Further when participants were asked about the process aspects of existing CIA approaches, most of the participants showed their disagreement in relation to what existing CIA approaches should have. More importantly, process aspects such as 'the steps', 'activity descriptions' and 'flow from one step to other step' were regarded as missing from existing CIA approaches. Similarly, data related to other process aspects such as ease of use and ease of understandability were also not very encouraging. Figure 7.7 illustrates the data related to process aspects of existing CIA approaches.

Fifthly, a set of questions were asked in relation to the effectiveness of existing CIA approaches. As discussed in Section 4.6.3 the effectiveness of CIA refers not only to identifying all the relevant impacts, but also to avoid the problem of impacts overestimation or underestimation. Participants were asked about the effectiveness of existing CIA approaches in question (a) Are you confident in identification of impacts on IA design adequately by employing available information and existing CIA approaches? Most of the participant's responses were not very positive about the effectiveness of existing CIA approaches. For example, one of the participants responded as:

"...we cannot investigate change impacts deeply at architecture design (such as Information architecture) because there is no architecture design information available and cannot be used...and there is more possibility that we missed many change impacts that can be identified"

The researcher findings as described in 7.5.2.2 related to the availability of information also confirmed the participants' responses as there were no dependency information available to investigate the depth of change propagation at IA design. Another participant said:

"*...existing CIA approach is not much effective as only few impacts can be identified using available traceability information...however many ripple effects of a change get overlooked..."*

Likewise another participant responded:

"...we can only identify which system artefacts get impacted but the coverage of identified impacts at component of that artefacts is remain incomplete...we do not have dependency information and we cannot trace what design entities depend on what others and what design entities get affected due to the changes made in IA design..."

Participants' responses indicate that existing CIA approaches only employed traceability information and thus are limited to the breadth of change propagation. There was no dependency information available to be employed during change impact analysis. As a result, it was not possible to perform in-depth impact analysis on architecture design. Overall participants' responses indicate that existing CIA approaches were not considered an effective approach.

Further, participants were asked the question (b) How would you rate the ease of analysing change impacts based on the available information in current CIA approaches? Eight participants responded '*poor*' and two participants responded '*very poor*'. The finding in relation to the effectiveness of existing CIA approaches implies that an approach that focused both on traceability and dependency analysis is required to support an effective identification of change impacts.

Sixthly, an interview question focused on identification of change impacts on architecture design resulting from business processes changes- i.e. early identification of change impacts in Web systems. Participants were asked (a) Does any approach/technique/method (or any form of it) help you for early identification of change impacts in Web systems? Most participants responded that existing CIA approaches do not support early identification of change impacts. One of the participants stated:

"*...most of time we identify impacts of business process changes during coding phase and it was too late to work out on the impacts of those changes that were originally made to business processes. If the impacts of business process changes can be identified earlier like architecture design then it will enables us to work on the consequences of business processes change at early stage before coding or before detailed design…*

Similarly another participant, who was project manager stated:
"...during detailed designing and coding, high number of change impacts were identified that were originally stemmed from business processes changes...but these impacts were overlooked during architecture design activity and identified late…it effected cost/effort estimation and further on release planning..."

These discussions reveal that existing CIA approaches do not support early of identification of impacts in Web systems.

Finally, participants were asked about the characteristics of Web systems. When, participants were asked (a) To what extent, do existing CIA approaches address the characteristics of Web systems? Most participants responded that existing CIA approaches do not address the two characteristics of Web systems. One of the participants stated:

"... In Web systems development it is common that a change in business processes can effect on information architecture design. Additionally, the connection between business processes and architecture design is mostly complex and so it is hard to identify the impacts at architecture design. Result is that Web developers/architects have to track several combinations of design entities to find out the impacts of a change at architecture design.

Here it is important to note that Web systems characteristics need to be understood adequately and their consequential focus on identifying impacts on architecture design resulting from business processes changes need to be addressed.

Likewise another participant stated:

"...If the Web systems characteristic are not taken into account when identifying change impacts then a problem can come up where most of change impacts are not identified or correct impacts are not identified before detailed designing and coding stages but after detailed designing and during coding. If Web systems characteristics are understood well then a better way can be find to solve this problem."

The above discussion *reveals* that a less-focused consideration of Web systems characteristics during CIA potentially leads to the problem where detailed design and implementation actually begins before change impacts are adequately identified. It is observed from the interview data that it is important to take into consideration the characteristics of Web systems during change impact analysis.

7.5.4 Phase_2: Summary of change impact analysis during project release_A

During Phase 2, investigation of system changes and existing CIA approaches was carried out during *Project Release_A*. While investigating system changes during *Project Release_A*, main documentary evidences include business processes changes, architecture design changes, project archival records and data related to impact analysis results and these were all studied. While investigating existing CIA approaches during *Project Release_A*, pre-study interviews were conducted to understand the perspectives of Web developers/architect in relation to existing CIA approaches. Findings in relation to existing CIA approaches identified from the researcher's observation and inspection were found to be consistent with the findings derived from pre-study interviews. These findings include (i) there is no structured and formal approach of CIA, instead, change impact analysis was mostly performed based on gut feeling and previous experience of Web developers/architects (ii) un-availability of required information (such as dependency information) should be used for the purpose of CIA.

Findings such as early identification of impacts, traceability from business processes to architecture design entities, dependency analysis at architecture design and a structured, organised and consistent approach of performing CIA were regarded as very important and found missing during *Project Release_A*. Additionally, an adequate consideration of Web systems characteristics was also considered important during CIA and found missing from existing CIA approaches.

The findings from Phase_2 reveal that the identification of impacts on architecture design resulting from business processes changes was performed primarily based on previous experience, consultation with other developers/architects and outcomes of design reviews. Indeed, these findings provide significant opportunities for improving CIA practices at WDS. Adoption of a new CIA approach was aligned with the process improvement initiatives taken at WSD. Before instantiation of PMCIA a series of discussions were carried out with the process engineer and Web developers/architects. These discussions were mainly related to the adoption and implementation of a new CIA approach in *Project Release_B*. Finally, PMCIA was introduced as a supporting process for change management at WSD. There were other tool supports available to execute CIA activity including RequisitePro®, TestDirector® and Archium. RequisitePro®, TestDirector® was used for the purpose of traceability analysis in step#1 and Archium tool was used to retrieve design rules and design constraints in step#2 of PMCIA. We have described instantiation of PMCIA in Phase 3. The validation of PMCIA is further described and covered in Phase 4.

7.6 Phase 3: Instantiation of PMCIA

In Phase 3, PMCIA as a new CIA approach was instantiated in *Project Release_B*. The researcher initially spent time at the WDS site to introduce PMCIA. A one-day workshop was conducted with an example of how to have a practice run with the case study participants. Besides that the researcher monitored the instantiated PMCIA and observed Web developers/architects while executing it. This was an unique opportunity because the researcher could observe *Project Release_B* from its beginning (November 2009) till the end (January 2010).

The details of instantiation of PMCIA and execution of its three steps in *Project Release_B* are given in Appendix I. During the instantiation of PMCIA, mainly business processes changes and identification of impacts on architecture design resulting from business processes changes were examined. Details of business process changes and results of change impact analysis are presented in Section 7.6.2.

7.6.1 Phase_3: Data collection and analysis

The following sections describe data collection and analysis approaches employed during Phase 3. A number of data sources were used including business processes specifications, architecture design specifications, project weekly meeting minutes and project measurement and analysis data. The details of change requests in Project Release_B were stored using a tool called IBM RequisitePro®.

Similar to Phase 2, the approaches used for data collection includes documents/tool inspections, project repository inspections, informal discussions with project teams and non-participatory observations in meetings. On some occasions, after weekly project meetings, SEPG meetings and document inspections, the researcher identified issues or questions that needed to be answered. The researcher then initiated discussions to address those issues with relevant project team members. These discussions were not recorded; instead the researcher took extensive notes or wrote the results of the discussion in the case study notes immediately after the discussion was completed.

The collected data during Phase 3 are listed as follows.

1. Total number of changes made in business processes,
2. Total number of identified change impacts to architecture design to address business processes changes (using PMCIA),
3. Total number of identified change impacts to various system artefacts (detailed design, code) to address business processes changes (using PMCIA)

Mainly quantitative analysis was used for the data collected after the instantiation of PMCIA in Project Release_B. The first focus of quantitative analysis was to quantify the changes such as 'business processes change'. The second focus was to quantify change impacts as 'identified change impacts on architecture design (and on other system artefacts) resulting from business processes changes and their comparison with Project Release_A. The comparisons of data presented in Phase 2 (in relation to the old CIA) and Phase 3 (in relation to a new CIA approach - PMCIA) provide the necessary basis for validation of PMCIA in Phase 4.

7.6.2 Phase_3: Case study findings from project release_B

The findings from Phase 3 provide valuable information about business processes changes and results of change impact analysis. In the following sections we describe those findings in details such as business processes changes and identified change impacts to address those business processes change during *Project Release_B*.

7.6.2.1 Business processes change

All the changes in *Project Release_B* were reported using change request forms and details of each change was stored in RequisitePro®. A total of seven change requests were reported during the first four weeks, given in Appendix I. All these changes were scheduled in the project plan for implementation during *Project Release_B*. Of all the change requests, four out of seven were related to the changes made to business processes. The rest of the change requests, three out of seven, were related to code reviews, performance issues, and UAT issues as shown in Figure 7.8.

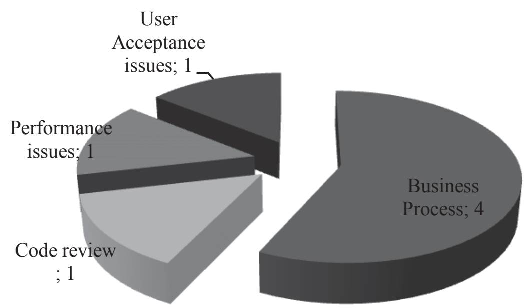


Figure 7.8: Percentage of different types of changes during *Project Release_B*

7.6.2.2 Identified Change Impacts on Architecture design

The traceability from business processes changes to architecture requirements and further down to architecture design was used to identify starting impact sets in *Project Release_B*. Further, an adequate identification of change impacts on architecture design was supported by employing design decisions information such as design rules and design constraints. By employing design rules and design constraints, change impacts on architecture design were identified during *Project Release_B* and called as estimated

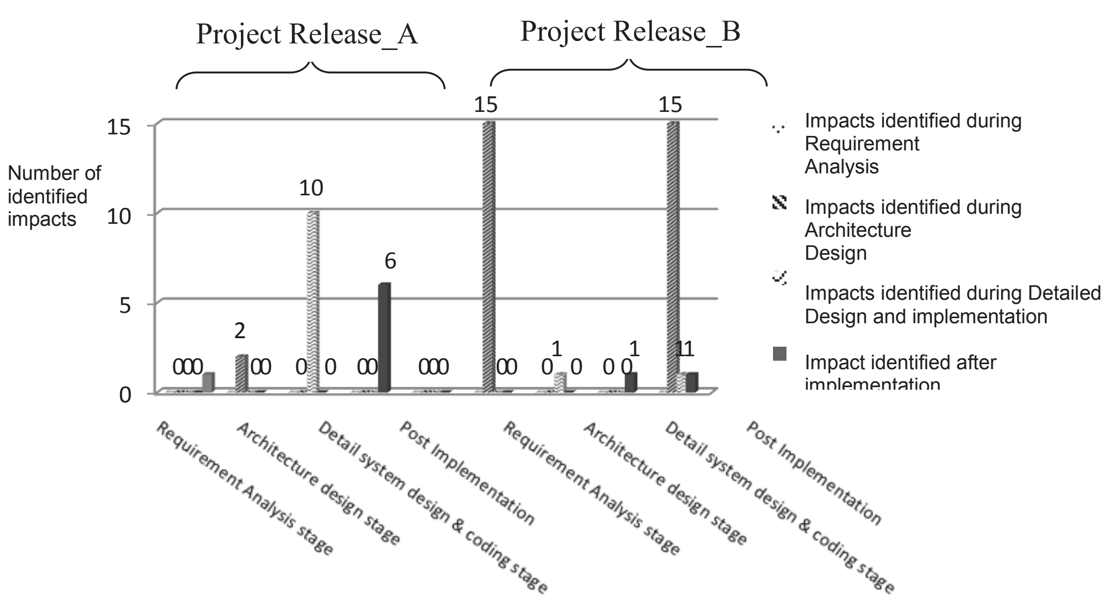


Figure 7.9: Identified impacts at architecture design and later stages of system development to address business processes changes during *Project Release_A and Project Release_B*

impact sets. Indeed, estimated impact sets subsume starting impact sets and can be considered as an expansion of starting impact sets.

In the right hand side of Figure 7.9, data related to *Project Release_B* shows the number of change impacts identified at architecture design to address business processes changes. These impacts at architecture design were identified based on the use of PMCIA in *Project Release_B*. In order to address business processes changes, the total number of change impacts identified in *Project Release_B* was seventeen. Out of these seventeen, fifteen change impacts were identified at architecture design to address business processes changes. Most of these identified impacts were the consequence of making changes in business processes and partially due to the ripple effect of changes made at architecture design. The data indicates an increase in the number of change impacts (as compared to *Project Release_A*, shown in the left side, Figure 7.9) identified at architecture design in *Project Release_B*. This is a clear indication that most of the change impacts are identified at the architecture design stage instead of being late during subsequent system development stages such as detailed design and coding. Conversely in *Project Release_A*, most of the change impacts were overlooked at architecture design and these overlooked impacts were identified late, only during detailed design and coding phase as shown in Figure 7.9.

7.6.3 Phase_3: Summary of the change impact analysis_2

In this phase, findings related to business processes changes and identified impacts on architecture design are presented in Figure 7.8 and 7.9 respectively. The instantiation of PMCIA and the resulting findings leads the researcher and project team members towards an understanding of how early identification of change impacts in Web systems can be achieved.

7.7 Phase_4: Validation of the Process Model of CIA (PMCIA)

Mainly, the activities in Phase 4 were associated with the validation of PMCIA that was instantiated in *Project Release_B*. The instantiation of PMCIA at WSD was aimed at facilitating Web developers/architects for early identification of change impacts in Web systems. Therefore, Phase 4 of this case study was intended to gather information from the perspectives of participants on adopting PMCIA as a new CIA approach in the WSD project.

7.7.1 Phase_4: Data collection and analysis

Interviews were used as a main data collection technique in Phase 4. The interview questionnaire for post-study interviews was a modified version of the pre-study questionnaire that was developed in Phase 2. Changes to the post-study questionnaire were mainly focused on a set of questions to validate the PMCIA in order to support early identification of change impacts (i.e. identification of impacts on architecture design resulting from business processes changes) in Web systems. The post-study questionnaire can be found in Appendix H. Additionally, inspection of project documents and the project repository, and consultation of the notes taken during meetings (including project weekly meetings and SEPG meetings) occurred.

During the interviews, the participants' opinion on PMCIA was the main topic sought. A set of open-ended and close-ended questions were constructed for post study interviews. These questions were intended to gain insight into the participants' perspective on the instantiation of PMCIA in comparison with the old CIA approach. The participants' responses were analysed by using content analysis and descriptive analysis.

According to the schedule, face-to-face interviews were conducted and audio recorded along with the notes taken on questionnaires by the researcher. At the end of post-study interviews, all recorded interviews were transcribed by the researcher. Additionally, the interview data was analysed using both qualitative and quantitative analysis (descriptive analysis in the form numbers and percentages).

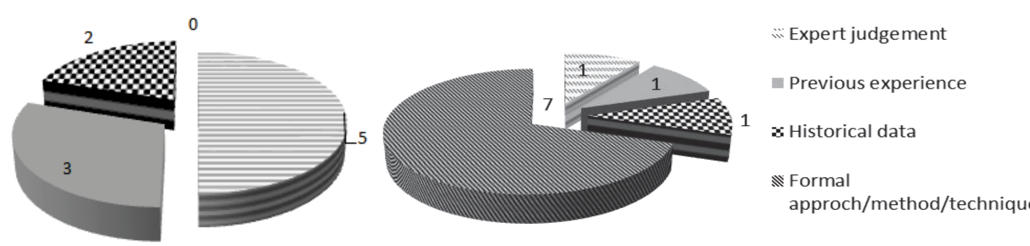
Most of the participants who participated in this post-study interview were the same WSD staff who had already participated in the pre-study interviews. In *Project Release_B* all interview participants were involved in examining business processes change, identifying estimated impacts on architecture design and implementing changes at architecture design while executing PMCIA from start to end. Four weeks after the instantiation of PMCIA (as a new approach) face-to-face interviews were conducted in three days. The interview schedule and questionnaire were emailed to the participants a week before conducting the interviews.

7.7.2 Phase_4: Findings from validation of the process model of CIA (PMCIA)

This section presents the findings from interviews that were conducted to validate the PMCIA. Interview findings were mainly based on the Web developers and architects' perspective in relation to PMCIA. Details of findings and observations received from post-study interviews are described in the following sections.

7.7.2.1 Current CIA practices

It was observed during post-study interviews that the instantiated PMCIA is a structured, organised and step by step approach to facilitate CIA during *Project Release_B*. It was also identified that the PMCIA described in three steps along with their underlying activities was a major improvement towards change impacts identification at WDS. In general, PMCIA encouraged Web developers to systematically adopt necessary components of CIA and to rigorously incorporate design decision information to support impact analysis activity.



Project Release_A Project Release_B

Figure 7.10: Comparison of possible ways to perform change impact analysis in two different releases by using two different CIA approaches

Firstly, the data in relation to the importance of different mechanisms of CIA in *Project Release_B* and its comparison with *Project Release_A* are illustrated in Figure 7.10. Examining the results reveals that in *Project Release _A*, five of the participants relied on expert judgement, whereas in *Project Release _B,* seven participants mainly relied on (a) formal method/approach (PMCIA), one participants used (b) previous experience, (c) expert judgement and (d) historical data respectively along with PMCIA to perform change impact analysis. These results indicate that participants mainly relied on PMCIA as a formal approach along with other minimal contributing factors such as their own judgement, experience and historical data.

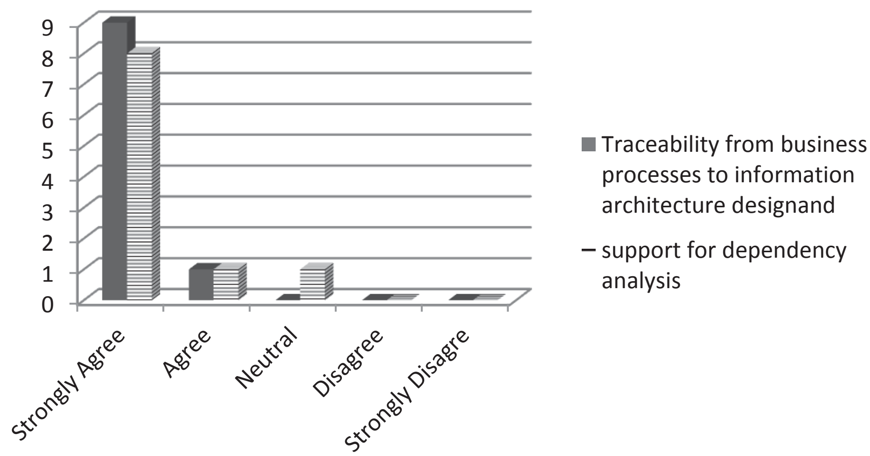


Figure 7.11: Participants' agreement on traceability and dependency analysis in relation to PMCIA

Secondly in relation to PMCIA, participants responded to (a) traceability from business processes to information architecture design (nine participants = strongly agree and one participant = agree) and (b) support for dependency analysis (8 participant = strongly agree, one participant =agree and one participant = neutral). Figure 7.11 shows the data related to two main aspects of PMCIA (i) traceability and (ii) dependency analysis during PMCIA.

The results illustrated in Figure 7.11 reveal a very positive result towards traceability and dependency analysis employed during PMCIA. Having support for both traceability analysis and dependency analysis in a single approach is an important feature of PMCIA.

Thirdly, most of the participants responded very positively towards the availability of information to carry out traceability and dependency analysis (i.e. for the breadth and depth of change propagation respectively). Figure 7.12 illustrates the data to depict the availability of important information in PMCIA- a new CIA approach and its comparison with old CIA approaches. When participants were asked what supporting information for CIA was available in PMCIA, most participants responded to the questions (a) Is sufficient information available to identify the ripple effect at architecture design? (b) Is sufficient information available to trace down change impacts to the level of impacted IA area? (c) Is there support to employ design decisions information to investigate the possible dependencies among architecture design entities?

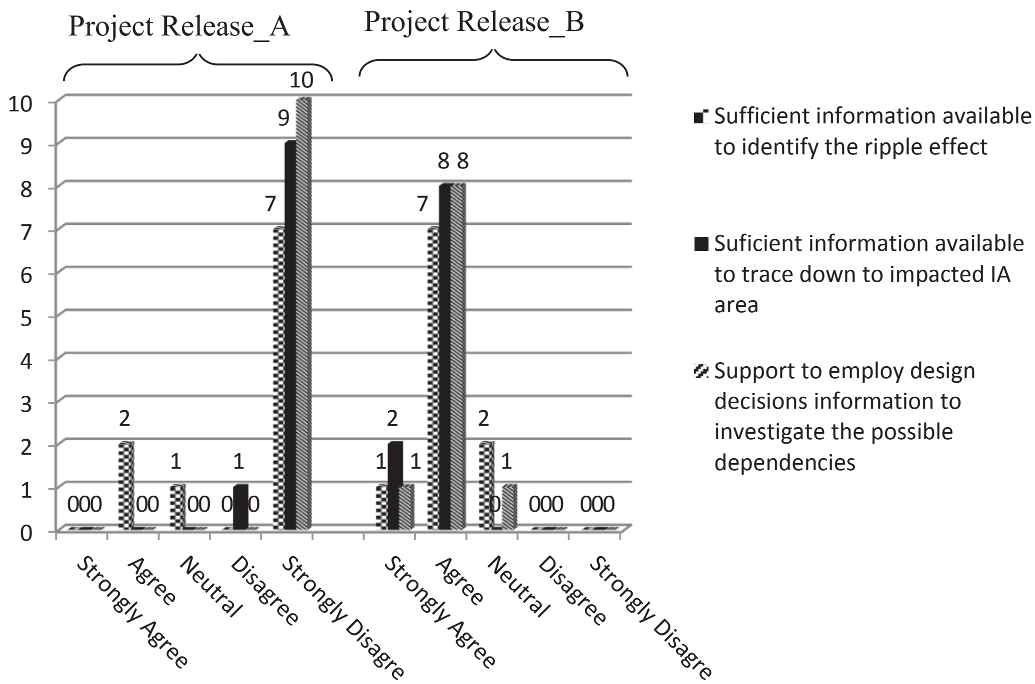


Figure 7.12: Participants' response on availability of information using old CIA approaches and PMCIA during *Project Release _A* and *Project Release _B* respectively

as '*agree'.* A few participant responded as strongly agree and neutral in relation to the availability of supporting information. However, no participant responded as neither disagree nor strongly disagree for any of the above questions.

In Figure 7.12, participants agreement both on employing sufficient information to trace down impacted architecture area and to identify the ripple effects at architecture design, indicates that PMCIA tends to support in-depth change impact analysis. This is true as PMCIA extends support for dependency analysis by employing design decisions information.

Fourthly, while referring to old CIA approaches during *Project Release_A*, important process aspects such as 'steps', 'activity descriptions' and 'feedback mechanism to evaluate the results of impacts after verifying affected architecture design' were reported as missing by the participants. Whereas, in *Project Release_B*, when participants were asked to show their agreement or disagreement on the scale of (in relation to PMCIA), seven participants responded as '*strongly agree'* and three participants responded as *'agree'* for (1) steps in PMCIA are easy to understand.

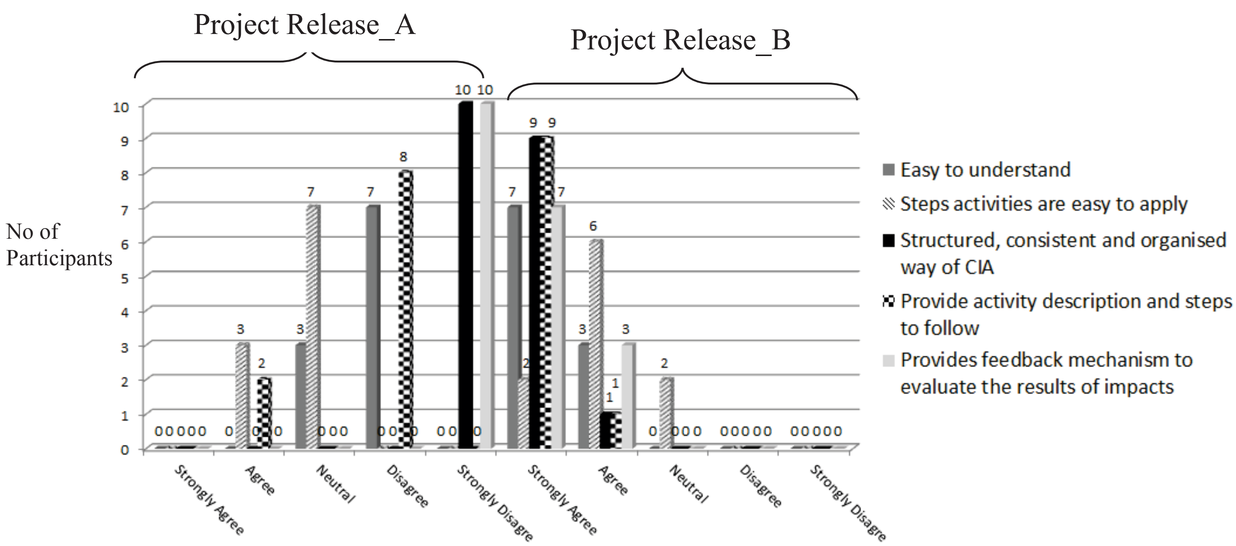


Figure 7.13: Participants' Agreement on the process aspects of PMCIA during *Project Release_B*

Similarly, most of the participants showed their agreement, as two responded 'strongly agree', six responded "agree' and two responded as "neutral' for (2) the steps activities are easy to apply (such as traceability matrix, forward tracing and dependency graph). However, nine participants responded as *'strongly agree'* and one participant responded as 'agree' both for (3) structured, consistent and organised way of performing impact analysis activity and (4) provide activity description and steps to follow. Additionally, there was an agreement for (5) provide feedback mechanism to evaluate the result of impacts after re-testing affected architecture design as seven participant responded as '*strongly agree' and* three participants responded as '*agree'*, as shown in Figure 7.13.

Fifthly, in relation to the effectiveness of PMCIA while identifying change impacts, participants were asked (a) Are you confident in identifying adequate impacts at architecture design based on supported information employed during PMCIA? Most participants responded positively for adequate identification of change impacts using PMCIA. One of the participants responded:

"*...in this project, we have identified direct impacts as well as indirect impacts and we appreciate thorough analysis and more satisfactory identification of change impacts. Important thing is that by mining design decision information, correct impacts can be identified. PMCIA use traceability links and dependency information and therefore covers breadth and depth of change propagations… we are confident that adequate change impact can be identified with PMCIA…"*

Likewise, one of the participants said:

"...the good point in PMCIA is that it cover both traceability and dependency analysis, and with that sufficient change impacts can be identified in Web systems projects... we are sure that most of impacts on architecture design can be identified by using design rules and constraints (coming from design decision information) as compared to old CIA approach where we don't utilise any design information"

Overall, PMCIA results in a more objective and thorough analysis of change impact during *Project Release_B*. As one of the participants stated:

"...PMCIA leads to a more objective (because it is based on defined steps and activities) and systematic analysis of change impacts and surely it reduces the chances that something has been missed during the analysis..."

From the participants' responses, it is evident that PMCIA employed both traceability information (to cover breadth of change propagation) and dependency information (to cover depth of change propagation). As a result, there were fewer tendencies to overlook impacts during CIA at architecture design. This finding implies that an approach focused both on traceability and dependency analysis can lead to an adequate identification of change impacts at architecture design. One participant, a project manager said:

"….PMCIA supports for a good and through analysis because it uses traceability and dependency analysis both. It minimises the possibility that any change impact can be overlooked at architecture design…from our recent experience, by using PMCIA we successfully identify a set of impacts that is neither overestimation nor underestimation of real impacts…but only correct set of impacts"

Likewise one participant reported:

"…PMCIA removed a lot of the guesswork. Previously I relied upon my experience, imagination and pretty much on an educated guess. I am more confident now that I hadn't left out anything using the PMCIA model."

Further, according to the description of effectiveness of CIA given in 7.5.3.2, PMCIA results in neither an over nor under estimation of identified impacts. Subsequently, PMCIA leads to a more effective (as compared to old CIA approaches) identification of change impacts.

Sixthly, there was a specific focus in PMCIA to support the identification of impacts on architecture design resulting from business processes changes. The participants were asked (a) Does process of CIA support early identification of change impacts in Web systems? 100% of the participants showed their agreement.

One of the participants stated:
"…Using PMCIA, we are able to identify more impacts on architecture design due to changes made in business processes…therefore early identification of impact in Web systems is possible due to reason as PMCIA provide a structured, consistent and systematic approach..."

Additionally, it was observed from the interview data that early identification of impacts in Web systems is an obvious need so as to reduce the unnecessary re-work during later stages of system development. Indeed, PMCIA supports early identification of change impacts in Web systems. Likewise one of the participants reported:
"*...most of change impacts on architecture design resulted from business processes changes are identified, as PMCIA greatly helps for early identification of change impacts in Web project. Next, it helps to reduce the re-work (cost of overlooked change impacts) during implementation phase.*"

Another participant responded:
"When using PMCIA I was able to identify possible impacts of business processes changes on architecture design. Whereas before mostly I can only found impacts of business processes changes during detailed design or once coding started…"

The responses from the interview participants revealed that early identification of change impacts was viable as (i) PMCIA supports to identify impacts on architecture design resulting from business processes changes and (ii) PMCIA employs both traceability (from business processes to architecture design entities) and dependencies (among architecture design entities) in a single comprehensive approach. Therefore, most of the change impacts were identified as early as they could be and thus reduced the complexity and high cost associated with overlooked impacts in later stages of system development.

Finally, regarding the two important characteristic of Web systems, participants were asked about (a) To what extent, did PMCIA address the two characteristics of Web systems? i.e. (i) business processes are intricately interwoven with Web systems and (ii) there is a tighter-connection between business processes and architecture design of Web systems. Most participants agreed that PMCIA tends to address the two characteristics of Web systems by catering the resulting focus of impacts identification on architecture design resulting from business processes changes. Likewise one of the participants stated:

"... from our experience in this project we have recognised that a change in business processes can noticeably effect on information architecture design. We also have experienced from previous system development that a potential change in business processes needs to be directly reflected at information architecture design to keep customer happy. Based on our current experience and previous understanding, our experience with the instantiation of PMCIA is really positive and greatly help to identify change impacts on architecture design (such as information architecture) and this definitely guide us toward a more successful Web-based business…"

One of the participants stated:

"…understanding of system characteristics is important for change impact analysis. In my view PMCIA support for two characterises of Web systems and thus address the identification of impacts on architecture design when changes made in business processes…"

A potential change in business processes may typically lead to fundamental changes in supporting IA design and it is mainly due to both the business processes being intricately interwoven with Web systems and there being a tight-coupling between business processes and architecture design of Web systems. The resulting emphasis of addressing these two characteristics is to support early identification of change impacts in Web systems. Firstly, PMCIA specifically focuses on the linkages between business processes and architecture design entities and thus essentially supports to identify impacts on architecture design resulting from business processes changes. Secondly, PMCIA focuses on the possible dependencies among architecture design entities and supports for an adequate identification of impacts on architecture design (before the implementation actually begins).

7.7.3 Phase_4: Summary of validation of process model of CIA

In Phase 4, we covered the activities related to the validation of PMCIA. An interview questionnaire was primarily used in Phase 4 to seek participants' perspectives on adopting PMCIA in *Project Release_B*. Findings from the interview data and important results were also discussed in this phase. There were quite positive responses received from interview participants in relation to validating PMCIA.

Findings from CIA approaches and outcomes of those approaches during *Project Release_A* (Phase 2) and *Project Release_B* (Phase 4) are described in order to better understand whether PMCIA supports to address both early identification of change impacts and the characteristics of Web systems. Post-study interview findings show very encouraging results while adopting PMCIA in Web system project. Indeed, the findings from post-study interviews lead to validating PMCIA for supporting early identification of change impact in Web systems. Additionally, the participants seemed mostly satisfied with PMCIA as an approach to address the characteristics of Web systems.

7.8 Chapter Summary

In this chapter, research activities conducted were aimed at addressing the research questions *RQ11: Does the proposed process model of CIA support Web developers/architects in early identification of change impacts in Web systems? RQ12: Does the proposed process model for CIA address the specific characteristics of Web* *systems?* From the data collection and analysis, it is concluded that the above research questions were indeed reaffirmed through a case study in a real industry setting.

Case study findings indicate that PMCIA supports for early identification of change impact in Web systems. These findings are articulated from a number of information sources including investigation change impact analysis results, observations from meetings (project and SEPG) during *Project Release_B* and post-study interview data. The case study findings also revealed that PMCIA employs both traceability and dependency information to support an adequate identification of change impacts on architecture design resulting from business processes. In essence, these findings serves as indicative proof of concept to support the claim that PMCIA successfully achieved the case study objectives described in Section 7.2.

While investigating 'does PMCIA address the characteristics of Web systems', it was observed that the resulting focus to address two characteristics of Web systems lies on adequately support identification of impacts on architecture design resulting from business processes changes. The insights gained from the analysis of the interview data indicate that firstly, PMCIA specifically focuses on the linkages between business processes and architecture design entities and thus mainly supports identification of impacts on architecture design resulting from business processes changes. Secondly, PMCIA focuses on possible dependencies among architecture design entities and subsequently supports an adequate identification of impacts on architecture design. Overall, the findings from post-study interviews indicate that PMCIA addresses the two characteristics of Web systems by taking into account the identification of impacts on architecture design resulting from business processes changes.

While considering what aspects of PMCIA contribute to early identification of change impacts in Web systems, it was identified that employing both traceability and dependency analysis in a single approach facilitates identifying impacts on architecture design resulting from business processes changes- early identification of change impacts in Web systems. Specific focus on the traceability from business processes to architecture design entities mainly supports to identify which design entities get affected by business processes changes (breadth of change propagation). Whereas, specific focus on analysing dependencies among design entities support to identify what design entities get affected by the modification of other design entities in architecture (depth of change propagation).

Additionally, the responses from participants indicate that employing design decision information support for the identification of dependencies among architecture design and mainly assist change impacts analysis. In particular, a more effective identification of impacts on architecture design was mostly achieved by using design rules and design constraints (as a set of design information), as compared to old CIA approaches where this important design information was not used.

Finally, case study findings also indicate that process aspects such as ‘steps’, ‘activity descriptions’ and ‘feedback mechanism used to evaluate the results of impacts after verifying affected architecture design’ were reported as important for a CIA approach. Indeed, steps followed during CIA, activity descriptions for each step, and feedback mechanism to evaluate the results of impacts after verifying affected architecture design are arguably important aspects of PMCIA.

Chapter 8: Conclusions

8.1 Introduction

In this thesis, we presented an investigation of the early identification of change impact in Web systems. By early identification of impacts we mean the identification of impacts on architecture design resulting from business processes changes in Web systems. It was noted earlier (Chapter 2 and 7) that failure in attempts to identify change impacts on architecture resulting from business processes changes leads to the problem where implementation (detailed design) begins before the impacts (on architecture) are adequately identified. As a consequence, many of the change impacts caused by business processes changes may go undetected or only are identified very late (Kagdi and Maletic, 2006) and often lead to unnecessary re-work during the later stages of system development (Michael, 2004). To address this issue, a systematic and rigorous approach termed as the process model of CIA (PMCIA) was developed and validated to support early identification of change impacts in Web systems.

In this chapter, the conclusions from the investigation of three phases (as described in Chapter 3) are presented. This chapter begins with revisiting research objectives and questions in Section 8.2, presenting research outcomes in 8.3, and describing implications for research and industry in Section 8.4 and 8.5 respectively. Further research limitations and suggestions for future work are presented in Section 8.6 and 8.7. In Section 8.8 are the concluding remarks.

8.2 Research Objectives and Questions Revisited

As stated in Chapter 1 and 3, the work undertaken in this thesis sets out to undertake exploratory and empirical research in Web system development that improves the understanding of linkage between business processes and architecture design of Web system and which supports early identification of change impacts in Web systems. In pursuing this research we identified several research objectives and underpinning research questions. As these questions formed the basis by which the research objectives of this thesis are satisfied, they will each be discussed in turn. Therefore, in this section the research objectives and research questions that motivated the work are revisited and summarised.

Research Objective 1: To analyse the characteristics of Web systems that need to be considered during Change impact analysis (CIA).

RQ1: What specific characteristics of Web systems need to be considered for CIA and why?

RQ2: What type of changes and their impacts are we interested in with regards to Web systems?

This group of questions can be answered primarily through an appropriate analysis of existing Web system development literature and CIA literature in general and particularly in relation to Web systems (see Chapter 2). We analysed the distinct nature of Web systems, system development approaches, nature and type of changes and their impacts in Web system (see Chapter 2 and 4).

For Question 1 we found that possibly, in Web systems, the combination of two characteristics in relation to CIA is most significant. These two characteristic include (i) Web systems are intricately interwoven with business processes (Bucknell et al., 2008, Yusop, 2009) and (ii) there is a tight connection between business processes and architecture design of Web systems (Lowe and Henderson-Sellers, 2003). It was observed that intricately interwoven characteristics and tight couplings between business processes and architecture design tend to differentiate Web systems from traditional software systems (see Section 2.3). It was argued that in Web systems, there is much more fine-grained evolution at the architecture level in order to address the intricately interwoven characteristics. Thus, partly as a consequence of intricately interwoven characteristics, the connections between business processes and architecture design solutions are much tighter than for traditional software systems. Given this tight-connection, any change in business processes often leads to fundamental changes at supporting architecture. The importance of these two characteristics in relation to CIA (determined both from the state of the art and state of the practice) guide us towards the distinct dependency of architecture design on business processes and to better understand the way in which business processes changes can impact on the architecture design of Web systems. It was also observed from practitioners during the interview based study that adequate understanding of related Web systems characteristics facilitates to identify impacts on architecture design resulting from business processes changes - early identification of change impacts in Web systems. Overall we concluded from the interview data that the specific Web systems characteristics need to be recognised in order to understand them adequately for the purposes of CIA (see Section 4.6.5)

For research question 2, given the two said characteristics of Web systems in relation to CIA, we found that (i) potential changes in business processes often leads to fundamental changes at supporting architecture (in Web systems) and (ii) while addressing the changes in business processes, a single change in architecture may have substantial impacts on other parts of the architecture and result in the ripple effects of the change being made to architecture. To support the intricately interwoven and tight connections between business processes and architecture design, it would be beneficial to identify what needs to be modified - specifically at architecture design to address business processes changes (see Section 2.4). This specific focus guided us toward the analysis of impacts on architecture design resulting from business processes changes in Web systems.

In our research we demonstrated how our approach took into consideration the important characteristics of Web system in relation to CIA and the type of changes and their impacts in the Web system (see Section 5.7).

Research Objective 2: To investigate scholarly/research literature for change impact analysis including existing approaches, methods, techniques and tools, and to identify gaps from the scholarly/research literature while adopting existing CIA approaches in Web systems.

RQ3: Why is CIA important for Web systems?

RQ4: What are the relative strengths and weaknesses of existing CIA methods, techniques and tools while adopting these CIA approaches in Web systems?

RQ5a: What does it mean by early identification of change impacts in Web systems?

RQ5b: Why is early identification of change impacts important in Web systems?

This group of questions can be answered through the analyses of 'existing CIA approaches and their adoption in Web systems', 'importance of CIA in Web systems' and 'the pressing need for early identification of change impacts in Web systems' (see Section 2.6 and 2.7)

For Question 3, we found that in Web system business processes changes are frequent and the degree of those changes and their resulting impacts as much higher than compared to traditional web systems. Given the multitude of connections from business processes to architecture design, it was observed that changes made to business processes can affect different aspects of architecture design including navigational design, informational design and transactional design. Hence both degree and immediacy of effects, and wide spread effects on architecture design resulting from business processes changes tend to differentiate the nature of changes and their impacts in Web systems (see Section 2.4). Therefore, we found that it is very important to perform the change impact analysis in Web systems in order to identify the consequences of making changes to system artefact and to support decision making with regards to those changes.

For Question 4, we found that one class of CIA approaches focus on traceability analysis to identify change impact from one system level artefact to another system level artefact. Traceability analysis follows a horizontal analysis approach, covers breadth of CIA and tends to support change impacts identification at a coarse-grain level of system artefacts. Another class of CIA approaches focuses on dependency analysis to identify impacts within a system artefact i.e. from one component of system artefact to another component of system artefacts. Dependency analysis follows a vertical analysis approach, covers depth of CIA and tends to focus on fine-grain level of system artefacts. We argued that to address the identification of impacts on architecture resulting from business processes changes, it will be beneficial to employ both traceability analysis (from business process to architecture design) and dependency analysis (within architecture design) (see Section 2.7.1 and 2.7.2). Most of the existing CIA approaches focused either on traceability or dependency analysis but failed to integrate both traceability and dependency analysis into a single approach to support adequate identification of change impacts (see Section 2.7.5). We also found that current research focus of architectural level CIA reveals that most of the approaches developed support software system domains, where designing activity begins as a predecessor of the requirement analysis phase and change identification and their assessment is typically initiated at architecture design level. Consequently, these existing CIA approaches are not adequate to be adopted in a Web system context.

For Question 5, we found that specific Web systems characteristics reveal that there is a special dependency of architecture design on business processes that needs to be addressed while performing CIA in Web systems. Further, this investigation guides us toward proposing early identification of change impacts in Web systems, that is, the identification of impacts on architecture design resulting from business process changes. In this thesis, we have referred to this as early identification of change impacts in Web systems. We argued that failure in attempts to identify change impacts on architecture resulting from business processes changes lead to the problem where implementation (detailed design) begins before the impacts (on architecture) are adequately identified. As a consequence, many of the change impacts caused by business processes changes may go undetected or are identified very late and often lead to unnecessary re-work during the later stages of system development.

Research Objective 3: To investigate industrial practices of CIA and to identify the limitations of existing CIA (in practice) for Web systems.

RQ6: What are the industrial practices of CIA in Web systems?
RQ7: How do the existing CIA practices and approaches support early identification of change impacts?
RQ8: What kind of information do Web developers currently use to perform change impact analysis?
RQ9: What are the high level features that CIA approaches should address in Web systems?

Questions related to research objective 3 was answered by investigating the commercial practices and identifying the industrial perspective in relation to existing CIA approaches, adopted in Web systems. These four research questions were satisfied by the interview-based study which we undertook. This study was based on face-to-face interviews with industry practitioners which looked at what is the industrial perspective of existing CIA approaches while adopted in Web systems, shortcomings of those existing approaches, information used and the set of features (that a CIA approach should possess) considered critical as they help practitioners to identify the effect of business processes changes on architecture design of Web systems.

Question 6 and Question 7 were satisfied by a series of interview questions. In general, participants indicated the importance of CIA and particularly the importance of early identification of change impacts in Web system. However, any systematic approach for impact analysis was largely not part of Web system development, so developers/architects were inclined to develop their own approaches. This is not unexpected given that most practitioners depend primarily on their experience, expert judgment, gut feeling, intuition and communication skills throughout the change impact analysis activity. It was also found from the interview-based study that failure to support early identification of change impacts in Web systems leads to the problem where implementation (detailed design) begins before the impacts (on architecture) are adequately identified. As a consequence, many of the change impacts caused by business processes changes may go undetected or are identified very late and often lead to unnecessary re-work during the later stages of system development.

Question 8 is supported by investigations with practitioners about what information they used for CIA. It was repeatedly referred to during interviews that business processes changes were frequent in Web systems and led to the situation where the traceability of those changes down to the level of architecture design were difficult to study. It was also observed from the interview data that the dependency analysis was difficult to perform due to the unavailability of adequate dependency information among design entities (see Section 4.6.1). It was reported in Chapter 5 and 7 that employing both traceability information (from business processes changes to architecture design) and dependency information such as design decision information (to identify dependency among architecture design entities) allows a deeper understanding in regard to early identification of change impacts in Web systems. Related research literature indicates that, in Web systems, all impacted design entities are not immediate but rather could be identified gradually while employing architecture knowledge (such as design decision information). We have demonstrated that, indeed, correct change impacts can be identified both through traceability and dependency analysis. It was also found that the consideration of Web systems characteristics also led to adequate identification of change impacts, and thus reduced the possibility that any potential impacts were overlooked.

Question 9 is supported by investigating the industrial perspectives towards a set of features deemed important (guided by the shortcoming of existing CIA approaches) for a CIA approach. Since there was no structured and organised approach available to be widely used by the participants, many of them felt the criticality of this need as it helps them to understand how changes in business process affect architecture design of Web systems. This is particularly important given that mostly ad hoc and unstructured CIA approaches were used in Web systems (see Section 4.5.1). Since the CIA approaches were not formalised, no tool support was available at the time of study. Therefore instead of tool support the major focus was on the basic process functions such as CIA steps, and their descriptions, flow of steps, employing traceability and dependency information, process guidelines and process validation (see Section 4.6.2).

Research Objectives 4: Based on the review of scholarly/research literatures and practices, extend the current theory by proposing an extension of existing CIA approaches to support early identification of change impacts in Web systems.

RQ10: Based on the review of scholarly literature and practices, is it possible to extend existing CIA approaches?

RQ11: Does the process model of CIA address the specific characteristics of Web systems?

RQ12: Does the process model of CIA support Web developers/architects in early identification of change impacts?

These three research questions in relation to research objective 4 were satisfied by the development of a process model and its validation studies which we undertook. These validation studies included (i) a demonstration with an exemplar system (as a proof of concept) which looked at whether or not PMCIA supports for early identification of change impacts, and (ii) a case study which investigated how PMCIA supports Web developers/architects in early identification of the change impacts in an industrial setting.

For Question 10, we critically reviewed the state of the art and investigated the state of the practice, particularly to understand the shortcomings in the existing CIA approaches to be adopted in Web systems. Primarily, investigating the state of the art suggests that current architecture CIA approaches may not be suitable to adopt in a Web systems context and indicates an extension of CIA approaches for Web systems. Findings from the review of scholarly research literature also highlighted the inadequacy of existing CIA approaches while addressing aspects including (i) linkages between business processes and architecture design of Web systems, (ii) identification of impacts on architecture design resulting from business processes changes, (iii) a process level support for CIA, and (iv) employing both traceability and dependency analysis (v) to address the problem where the implementation (detailed design) actually begins before change impacts are adequately identified in Web systems (see Section 2.7.5). Interestingly, these five aspects were already among the set of high-level features of CIA identified from the interview-based study, which confirms that our listed high-level features were indeed relevant and reflects a reasonable coverage of features that CIA approaches should address. Indeed, investigation of the state of the practices provided guidance on how to extend current CIA approaches (see Section 5.2). Investigation findings from the state of the art and the state of the practices enable us to propose a process model of CIA - as a combinatorial solution both contributions to the body of knowledge and industrial practices.

To answer Question 11, we provided supporting discussions in Chapter 2 and 5. Investigation of Web systems characteristics reveal that there is a special dependency of architecture design on business processes that needs to be addressed while performing CIA in Web systems. Further, this investigation guided us towards proposing early identification of change impacts in Web systems, that is, the impacts identification on architecture design resulting from business processes changes. It has been reported that business processes are interwoven with architecture design of Web systems. However, there is a lack of understanding of this characteristic partially due to the fact that the traceability from business processes to architecture design is not explicitly well-established. This leads to the problem of inadequate analysis of change impacts and failure to identify impacts on architecture design resulting from business processes changes. In PMCIA, we proposed a step (step#1 of PMIA) particularly focused on developing and employing traceability links from business processes to architecture design (see Section 5.5.1). Another Web systems characteristic is the tight-connections between business processes and architecture design of Web systems. As a consequence, a potential change in business processes may typically lead to fundamental and wide-spread changes in architecture design which are developed to support those business processes. A lack of understanding of this characteristic may complicate the impact analysis activity further, as Web developers/architects have to track several combinations of design entities in order to analyse the effect of changes at architecture design. This leads to the problem of overlooking impacts on architecture design resulting from business processes changes. In order to address this issue, we proposed to employ design decision information in step#2 of PMCIA (see Section 5.5.2). We found that design decisions reveal a set of information such as design rules and design constraints considered important to support change impact analysis. The explicit focus to utilise design rules and design constraints assisted us in the identification of possible dependencies between design entities and thus supported impact identification at architecture design level.

Question 12 is answered by conducting a case study in a commercial setting. In particular, the case study supported the claim that PMCIA facilitated Web developers/architects in early identification of impacts in Web systems or not. The findings from the case study provided the evidence that a systematics and structured approach along with the necessary infrastructure provide by PMCIA facilitates Web developers/architects in early identification of change impacts in Web system. The case study provided evidence that in current practices (that is, without any use of PMCIA) impacts on architecture design were largely overlooked and mainly based on the developer's own experience and intuition. However, the use of the PMCIA approach is likely to support developers in better understanding the nature of business processes changes and their consequences on architecture design. As a result, PMCIA truly, supports early identification of change impacts in Web systems. By executing a structured set of steps and process flows (in terms of traceability analysis, dependency analysis and validation of identified impacts) the developers can adequately and correctly identify the effects on business processes changes on architecture design and the ripple-effect (on architecture design) of the changes being made. Therefore, the Web developers will be able to remove much of their guesswork, and be more confident that they have not overlooked some significant impacts.

8.3 Research Outcome

This section discusses the important research outcomes the extent to which they met the four objectives listed in Section 8.2 above and in Chapter 1. The research outcomes show how the research presented in this thesis is related to the research objectives developed at different stages of our research work and how the proposed process model of CIA has emerged over a period of time.

The research described in this thesis has been widely published and presented at different conferences and scientific journals and through these publications this research have already been subjected to a considerable level of peer review. It is quite reasonable to constitute and report the research findings in small increments as early as possible, rather than waiting till the end of the research process. This ensures timely feedback from both researchers and the industry. This approach has helped the researcher to proceed in the right direction during the whole research process. A complete list of research publications that were produced during this research project can be found in Preface.

There were two objectives for the research activities conducted in Stage 1. The scope of Stage 1 covered a thorough examination and critical analysis of scholarly literature, commercial practices, existing theory on and around the area of change impact analysis and Web systems development. The objective of this review was to establish a theoretical foundation for our research as well as making new insights and contributions – identified the gap from the research literature while adopting current CIA approaches in Web system. Particularly, the literature review contributed to articulating the research context encompassing identification of impacts on architecture design of Web systems (Mehboob et al., 2009) and the possible use of design information while performing change impact analysis activity.

The findings from the investigation of Stage 1 guided us towards Stage 2, where we examined industrial practices of CIA in Web systems context. The second stage of the research was the interview-based investigation of current practices in relation to change impact analysis during Web systems development. Firstly, the findings from these interviews were helpful in validating the theoretical foundation of our research (established from the literature review). Secondly, interview findings were also helpful so as to deeply understand the high level features/needs for the possible extension of current change impact analysis approach (Mehboob and Zowghi, 2009).

The triangulation of results from the two methods (i.e. literature review-state of art and interviews-based investigation-state of practices) was important for the integrity of the research. Indeed, it was not possible to depend entirely on the literature due to the work needing to be grounded in practice, and also due to the general lack of adequate evidence and sufficient industry-based publications in the area of change impact analysis during Web system development. The results from *Stage1 (literature review)* and *Stage 2 (interviews-based investigation)* of the research provided both the motivation and input for the development of the process model of CIA - the third stage of the research as an extension of current CIA approaches.

In Stage 3 of the research, a process model of CIA (PMCIA) was developed to illustrate the findings both from literature and industry practices. PMCIA developed as an extension of existing CIA approaches, and illustrates how a CIA approach can be extended to support early identification of impacts in Web systems while taking into consideration the related characteristics of Web systems (Mehboob et al., 2011). Design decision information is specifically used to analyse the dependencies among design entities and thus the possible change impacts. The validity of PMCIA was demonstrated through an exemplar Web system (Mehboob et al., 2011) and case study (Mehboob and Khan, 2013).

8.4 Thesis Contributions Revisited

The major contributions of this thesis are discussed below. We have briefly covered thesis contributions in Section 1.6. There are several contributions that this thesis has made to the Web engineering and software engineering body of research by demonstrating a process model of CIA that is currently used in early identification of change impacts in Web systems. The contributions of this thesis are discussed below.

The first contribution of this research is through a novel synthesis of research literature which guides us toward key research areas in relation to change impact analysis for

Web systems, in terms of what has not been addressed by existing CIA methods and approaches. Additionally the investigation of research literature guides future researchers towards an improved understanding of Web systems characteristics considered important for change impact analysis and a set of shortcomings in the current CIA approaches while adopted in Web systems. These elicited shortcomings in the current CIA approaches and their relevance in Web systems provides Web developers with the necessary understanding and informed decision making for the selection of CIA approaches. Furthermore, the investigation of the current state of the practices in relation to Web systems provides an insight of industrial perspectives on the use of current architecture level CIA approaches. Particularly, the investigation the current state of the practices form a set of high level features/needs that a change impact analysis approach should support for Web systems. A comprehensive list of high level features/needs to cater to the nature of changes impacts in Web system are added knowledge for future researchers. This is a contribution to the research community as there is currently little empirical work on this aspect.

The second contribution of this thesis is a process model of CIA specifically developed for and validated in the Web systems development project. The process model of CIA offers and improves understanding of early identification of change impacts-identification of impacts on architecture caused by the changes in business process as well that caused by the changes that are being made to the architecture/system design. This process model supports a structured, rigorous and consistent approach towards change impact analysis focusing on steps, activities in each step, inputs and outputs and the validation of impact sets produced from the process model. Through the use of this process model, practitioners acquire an increased understanding of the role and importance of employing design information during the change impact analysis. The novelty of this process model is the extension of current CIA approaches to support early identification of change impacts during Web systems development.

The third contribution is the demonstration of how case study research can be applied for the validation of the process model of CIA, and its suitability for validating new approaches in a real Web project context. Findings from the case study support to address the problem in relation to change impact analysis - where the implementation (detailed design) actually begins before change impacts are adequately identified, and thus it becomes prohibitively complex and expensive to address change impacts in subsequent stages of Web systems development. The findings also indicate that the likelihood of identification of impacts during subsequent stages of Web systems development will be substantially reduced as Web architects/designers will have a better understanding of impacts on architecture resulting from business processes changes.

8.5 Implication for Research and Industry

The insights gained from the data collections are very useful for researchers as there is currently little empirical research being reported on the topic of CIA approaches to support early identification of change impacts in Web systems. Consideration of specific Web systems' characteristics in relation to CIA, nature of changes and their impacts are also useful for researchers to understand the shortcomings in current CIA approaches in dealing with change impact analysis for Web systems. Further, investigations of current commercial practices, give an insight of the industrial perspectives on the use/adoption of current CIA approaches in Web systems and particularly, a set of high level features/needs that a change impact analysis approach should support for those systems.

The findings reported from the research literature and commercial practices form a comprehensive body of knowledge for researchers. The comparison results of CIA approaches and their relevance in Web systems assists Web developers to understand the shortcomings of those approaches while adopted in Web systems, and take them into account to select an appropriate CIA approach.

Understanding of those shortcomings in the existing CIA approaches is important due to specific characteristics of Web systems in relation of CIA. It was observed that intricately interwoven characteristics and tight connections between business processes and architecture design tend to differentiate Web systems from traditional software systems. It was argued that in Web systems, there is a much more fine-grained evolution at the architecture level in order to address the intricately interwoven characteristics. Further, partly as a consequence of intricately interwoven characteristics, the connections between business processes and architecture design solutions are much tighter than for traditional software systems. Given this tight-connection, any change in business processes often leads to fundamental changes at supporting architecture. Additionally, the scale and scope of those impacts on the architecture design is sufficiently large and immediate, and they should not be overlooked during the initial system development. As the scope and scale of the impacts increases, we need to identify those impacts correctly and adequately at architecture design resulting from business process changes i.e. early identification of change impacts in Web systems.

One common criticism of research is that it is not practical enough to be adopted by industry. We have demonstrated through this research that a process model of CIA - as an extension of CIA approaches - by employing design decision information can support early identification of change impacts in Web systems. The question we address here is what does this mean for the industry? In other words, how practical is it for the industry to adopt our proposed process model of CIA?

In relation to supporting early identification of change impacts in Web systems with the use of PMCIA, there are several implications for practitioners that can be drawn from this study. Firstly, most of the web developers informally analyses change impacts on architecture design of Web systems based on their past experiences from similar projects, expert judgment and gut feeling. Instead, it is important to follow a formal and educated approach where web developers can study the relationship between business processes and architecture design, and the scope of change impacts at architecture design by taking into account the traceability link from business processes to architecture design – traceability analysis. Secondly, at architecture design, web developers can study the ripple effect of a proposed change at architecture design by employing design decision information - dependency analysis. From the study of early identification of change impacts in Web systems, it was found that both traceability and dependency analysis play an important role in adequate and correct identification of change impacts. Important new knowledge that came out from this research is the importance of early identification of change impacts in Web systems by means of both traceability and dependency analysis to cover breadth and depth of analysis respectively.

The findings reported on the systematic and structured approach along with the necessary infrastructure provided by PMCIA facilitates Web developers/architects will provide web developers with an understanding of how a process model could be used in Web systems projects. It also provides web developers with an understanding of what steps, activities in each steps, flow from one activity to another, required inputs/outputs and other required tools are required so as to execute change impact analysis and achieve the desired results. In particular, PMCIA provides practitioners with an understanding of how the impacts on architecture design resulting from business processes changes could be identified before detailed design and implementation actually begins, thus, reducing un-necessary re-work at subsequent stages of Web systems development.

8.6 Research Limitations

There are two limitations of the research activities conducted. The first limitation for this research is related to interviewer bias. In Stage 2 and 3 the researcher conducted face-to-face interviews. Neuman (2000) states that there are six categories of interview bias: (1) errors by respondent; (2) unintentional errors or interview sloppiness; (3) intentional subversion by the interviewer; (4) influence due to the interviewer's expectations; (5) failure of an interviewer to probe or to probe properly and; (6) influence on the answers due to the interviewer's appearance, tone, attitudes and reactions.

In relation to the interviews conducted for this research, the researcher reduced the bias by placing great (and careful) emphasis on ensuring an accurate record of responses. In order to ensure consistency of data, the researcher listened to the interview on the audio recorder and compared it with the transcribed notes and the notes taken during the interview sessions. In addition to this, the researcher also used probes to allow the Web developers to clarify their responses.

The second limitation for this research is related to the application of PMCIA in specific scenarios. PMCIA can be used to support early identification of change impacts in Web systems projects but it still has some limitations, as it is unable to be applied to some Web systems. For example, PMCIA cannot be used if the architecture design model or architecture design specifications are missing, not updated and/or not documented. Samples of Web systems, architecture design models, and design specifications are shown in Appendix E.

8.7 Suggestion for Future Work

Our work has focused on an improved understanding of change impacts identification on architecture design of Web systems that is particularly guided by the consideration of specific characteristics of Web systems in relation to CIA. The research artefact termed as PMCIA, as an extension of CIA approaches has shown that it can assist Web developers in identification of impacts on architecture design resulting from business processes changes - early identification of change impacts in Web systems. This work can, however, be taken much further. Some of the important future research directions are listed below. Some suggestions for future work include:

As discussed earlier in Chapter 5, the post and pre-interviews had some limitations, such as a limited sample size (number of participants). Therefore, it would be useful to re-examine PMCIA with a larger sample size. In a similar vein, our case study was limited to a single case. However, additional cases would provide more compelling evidence and make for more robust conclusions.

We realise that PMCIA as an extension of CIA approaches (particularly in a way which can assist Web developers in early identification of change impacts) is only the first step in solving the problems where the detailed design or implementation actually begins before change impacts are adequately identified on architecture design – leading to un-necessary re-work at subsequent stages of Web system development. In undertaking future work in this area, we hope to create improved and proven CIA techniques and tools for contributing to the creation of an appropriate development process for Web-enabled systems.

8.8 Concluding Remarks

In summary, an extension of the CIA approach has been developed in this research to identify impact on architecture design resulting from business processes changes in Web systems – we refer to this as "to support early identification of change impacts in

Web systems" in the thesis. This work has been evaluated to support the idea that it is able to facilitate adequate and correct identification of impacts on architecture design which arise from changes that are made to the business process in Web system. This, in turn, resolves the problem where the detailed design or implementation actually begins before change impacts are adequately identified at architecture design – leads to unnecessary rework at subsequent stages of Web systems development.

Our proposed process model of CIA, referred to as PMCIA, has made a number of significant contributions toward existing approaches and practices. First, consideration of specific Web systems characteristics in relation to CIA guided us to fill the shortcoming of existing CIA approaches (particularly an improved understanding of change impacts identification on architecture design of Web systems). Second, the PMCIA is able to cover the breadth and depth of change impact analysis activity by employing both traceability and dependency analysis, which leads to adequate and correct identification of change impacts. Thirdly, PMCIA as a systematic and structured approach along with the necessary infrastructure, facilitates Web developers/architects in Web systems projects to achieve the desired results. This leads us to the contribution: applying PMCIA to support identification of impacts on architecture design resulting from business processes changes – early identification of change impacts in Web systems. Hence PMCIA as an extension to existing CIA approaches will improve the current CIA practices for Web systems.

Finally, we can conclude that the proposed process of CIA will be a good start for improving the developmental process of Web systems in terms of adequate and correct identification of impacts on architecture design as it can resolve a crucial problem, as discussed above.

Appendix A

Invitation letter for interview participants

Dear Sir/Madam,

I am writing to seek your interest for participating in the interview for a research project entitled "A Framework for Change Impact Analysis in Web Systems Development". This interview is part of a research project at the University of Technology, Sydney, Australia. We believe that the results of the interviews will be useful for Web developers in formulating strategies for early identification of change impacts in their Web development projects.

The participants (web developers/architects) from your organization will involve in approximately 1 hour interview session. The research student undertaking the project, Mr. Zafar Mehboob, will contact you in the near future and can discuss details with you. Mr. Mehboob is a PhD candidate supervised by Associate Professor Didar Zowghi and Professor David Lowe at the Faculty of Engineering and Information Technology at UTS.

We will send formal interview information such as consent form and interview scripts prior to conducting the interview. Your anonymity will be ensured and your responses will be kept confidential. For interview the results will be recorded in a form which has no reference to you or your organization. Whilst your participation in this interview is obviously voluntary we would value your contributions. If you have any questions or concerns about participating in this interview please contact Mr. Zafar Mehboob via email at zafa@it.uts.edu.au or phone 02-95144446.

Yours sincerely,
Didar Zowghi,
Associate Professor,
Faculty of Engineering and Information Technology,
University of Technology, Sydney, Australia.
Phone +61 2 9514 1860
Fax +61 2 9514 4535

Zafar Mehboob
PhD student,
Faculty of Engineering and Information Technology,
University of Technology, Sydney, Australia.
Phone +61 2 9514 1860

Appendix B

CONSENT FORM FOR INTERVIEWS

I ___________________ agree to participate in the research project entitled "A Framework for Change Impact Analysis in Web Systems Development" being conducted by Zafar Mehboob (Faculty of Information Technology, zafar@it.uts.edu.au, contact no. +61 2 95144446).
I understand that the purpose of this study is to investigate the current practices, rationale, and reasoning behind change impact analysis activity and particularly on understanding the needs for possible extension of current change impact analysis techniques to support early identification of impact in the context of web system.

I understand that my participation in this research will involve 1 hour of participation in interview questionnaire and would be recorded (audio-taped) if required. I also understand that there will be no harm or risk for me in this research.

I am aware that I can contact A/Prof. Didar Zowghi (didar@it.uts.edu.au), Professor David Lowe (david.lowe@uts.edu.au) or Mr. Zafar Mehboob (zafar@it.uts.edu.au) if I have any concerns about the research. I also understand that I am free to withdraw my participation from this research project at any time and without giving any reason if I feel uncomfortable of the questions asked on the questionnaires.

I agree that the research data gathered from this project may be published in a form that does not identify me in any way.

____ ____/____/____
Signed by

____ ____/____/____
Witnessed by

NOTE:
This study has been approved by the University of Technology, Sydney Human Research Ethics Committee. If you have any complaints or reservations about any aspect of your participation in this research which you cannot resolve with the researcher, you may contact the Ethics Committee through the Research Ethics Officer (ph: +61 2 9514 9772 Research.Ethics@uts.edu.au <mailto:Research.Ethics@uts.edu.au>), and quote the UTS HREC reference number. Any complaint you make will be treated in confidence and investigated fully and you will be informed of the outcome.

Appendix C

Interview Questionnaires

A Framework for Change Impact Analysis in Web Systems Development

Brief Summary of Research

Introduction:
Change Impact Analysis (CIA) is typically used within the software development process to support identification of the consequences of making a change to the requirements, design, code and test artefacts of a system and, to support decision making with regard to that change. Most of current CIA methods and techniques are driven by both an understanding of the interdependencies between components or sub-systems, as well as the software development process from requirements through design and ultimately to the implementation. This means that different development processes and architectures may require different CIA processes and techniques. This is particularly relevant in the context of Web systems, given that these systems often have a specific set of characteristics related to differences from traditional systems both in their architecture and the process through which they are developed. We may then assume that current CIA techniques would benefit from being specifically designed or adapted to accommodate the specific characteristics of Web systems. This research proposes to extend the theory and technique of change impact analysis to assist early identification of change impact analysis in Web systems thus to facilitate change management process. In order to achieve these objectives, we are interested to collect information about the rationale and reasoning behind change impact analysis, and particularly by examining how current approaches can leverage/adopted/extended to analyse change impact at early stages of Web systems development.

Read to participants:
This research is targeted at Web developers/architects who have experience architecture level change(s) in web development projects. The focus of our interviews are to observe current CIA approaches widely adopted within web systems development and to identify the problems and issues while adopting change impact analysis practices to cater changes at higher level of design abstraction (business models vs. information architecture).

PART A: Change Impact Analysis in Web systems Development

1. What are the goal(s) for analysing change effect/impact during architecture design changes in web development projects?
   - To facilitates in decision making
   - To facilitates in change estimation
   - To facilitates in reducing unnecessary re-work
   - To facilitates in release planning
   - Others

2. How does a change in business processes can potentially impact on information architecture (IA)?
   - Information organisation and its structure
   - Information exchange of internal/external actors to architecture design?
   - the internal inter-relationships between information domains
   - interaction with information environment in which the system exists
   - Others

3. What are/can be the consequences of not analysing impacts on information architecture design (IA) from business processes changes?

4. What types of change impacts on IA and should be identified as a result of business processes changes?
   - Structure changes (e.g. changes to information dictionary; addition, deletion and modification of information components, connector or link and changes to their properties)
   - Behaviour changes (e.g. changes in navigational flow, information exchange etc.)
   - Location changes (e.g. changes in file structure or in a site map; web directory structure etc.)
   - Others

5. Which type of change impacts makes the biggest difference with respect to user satisfaction and ultimate success of project (key objective of IA)? Why?

6. (a) How do developers analyse change/change propagation/change effect/change impact? Do developers use any informal way (e.g. expert judgment, previous historical data, and previous experience etc) or any other formal method or techniques?
   (b) Do developers have a standard set of step/activity or process to trace change impacts from business models to information architecture? If yes please explain it, if not then explain how impacts are analysed in Web development projects?
7. (a) Do any technique or method (or any form of it) help developers to support identification of change impact in Web systems?
   (b) using a technique or method (or any form of it) do meet developers expectation/goals (as respond for question 1)? If not why not?
8. Do developers believe that there is a need for some approach that assists them in identifying the impacts on IA resulting from business processes changes? If yes why and if not why not?

9. What kind of information do developers currently use to perform architectural change impact analysis? (e.g. traceability/dependency information, project documentation, historical change data etc)

10. What other information do developers believe to be used to support the identification of impact on architecture design?

11. When impact (on architecture design) are analysed in Web systems? Or at what stage architecture level change impacts are analysed?

- early once the change begins to start,
- during the development of architecture,
- after the development of architecture,
- during architecture maintenance
- Others

12. When does Information Architecture (IA) get change frequently during Web systems development?

PART B: Company and Project details

Company Details

1. What is the Company name?
2. What is the size of the organisation developers work for?
   - Small (49 or less employees)
   - Medium (50 to 250 employees)
   - Large (more than 250 employees)

3. Which of the following business sectors does your organisation belong to?
   - Education
   - Finance and Insurance
   - Government Administration and Defense
   - Health and Community Service
   - IT Services Web,
   - Web system development, E-Business Services
   - Others

4. Approx. how many employees would you have involved in web development business?

5. What is the complexity of the applications that you develop? Can you describe the types of technologies you use and the scale of the projects that you typically handle?

Project Details

1. Would you please briefly describe the project that you will focus while answering interview questions?
2. What was the total project budget for this project?
3. How many years project spans till now? (How old is the project?)
4. At what stage the project is right now?
   - Analysis
   - Design
   - Coding/implementation
   - Testing
   - Maintenance

5. How many people in the following positions were involved in this project?
   - Project Manager
   - Business Analyst/Architect
   - Web Designer
   - Information Analyst/Architect

- Web Developer
- Tester
- Others

6. Briefly describe your responsibility in this project?

7. How many Web systems development projects have you undertake?

Definition of terms:
Web system- In this questionnaire, the term Web systems refers to those systems which utilise Web technologies as an integral element of a functionally complex system which typically incorporates interfaces beyond the organisational boundaries (which includes systems such as online systems, as well as a host of domain specific terms such as business-to-business (B2B), business-to-consumer (B2C), etc.)
Change Impact Analysis- It is an approach of identifying the potential consequences of a change, or estimating what needs to be modified to accomplish a change.
Business Processes- Business processes define the business approach and the role that the system plays in supporting the business.
Architecture-The software architecture of a program or computer is the structure or structures of the system, which comprises software components, the externally visible properties of those components, and the relationships among them.
Information Architecture(IA)- Information architecture provides the glue that allows content and high level functionality to effectively work in tandem to mediate among business goals, users' requirements, engineering constraints, and design solutions.
Early identification of Impact- Here by early identification of impact in Web systems we mean identification of impacts at architecture design (such as IA) resulting from business processes changes.

Appendix D

29 July 2008

Associate Professor Didar Zowghi
CB10.04.575
School of Software
Engineering & Information Technology
UNIVERSITY OF TECHNOLOGY, SYDNEY

Dear Didar,
UTS HREC 2008-205 – ZOWGHI, Associate Professor Didar, LOWE, Professor David (for MEHBOOB, Zafar PhD Student) – "A framework for change impact analysis in web systems"
Thank you for your response to my email dated 17 July 2008. Your response satisfactorily addresses the concerns and questions raised by the Committee, and I am pleased to inform you that ethics clearance is now granted.

Your clearance number is UTS HREC REF NO. 2008-205A

Please note that the ethical conduct of research is an on-going process. The National Statement on Ethical Conduct in Research Involving Humans requires us to obtain a report about the progress of the research, and in particular about any changes to the research which may have ethical implications. This report form must be completed at least annually, and at the end of the project (if it takes more than a year). The Ethics Secretariat will contact you when it is time to complete your first report.

I also refer you to the AVCC guidelines relating to the storage of data, which require that data be kept for a minimum of 5 years after publication of research. However, in NSW, longer retention requirements are required for research on human subjects with potential long-term effects, research with long-term environmental effects, or research considered of national or international significance, importance, or controversy. If the data from this research project falls into one of these categories, contact University Records for advice on long-term retention.

If you have any queries about your ethics clearance, or require any amendments to your research in the future, please do not hesitate to contact the Ethics Secretariat at the Research and Innovation Office, on 02 9514 9772.

Yours sincerely,

Dr Chris Zaslawski
Acting Chairperson
UTS Human Research Ethics Committee

Appendix E

1. Instantiation of Process model of CIA

We have adopted an example case study from (Acerbis et al., 2008), tailored it to describe online personal loan system and used it for proof by demonstration of PMCIA.

1.1 Web system details

The online personal loan system provides customers to search for available loan, browse for features, terms and conditions, application fee, and application criteria for loan. Customers are able to fill applications form, and get notification about acceptance/rejection of loan requests. Personal loan system covers the whole loan request process and allows the staff to evaluate the loan request, check the details of loan application, register the final decision for an application, and end a given application. If the request is approved, the customer can accept it and proceed with the allocated amount along with its periodic re-payments. The selected online personal loan system describes a simplified business process executed on Web by two actors: customer and bank staff. There are three important activities during loan request and approval process. We have described the specification of online personal loan system as below.

Preliminary Validation

During PreliminaryValidation activity, the staff may either stop the activity if the application is not valid (e.g., it does not contain meaningful or real data) or fill in a form with his notes to continue the activity. In the latter case some new data are added to the application using "modify unit", and e-mail is sent to the customer providing an application reference number. If the LoanRequest is rejected outright, both the activity and the application are ended. Once the LoanRequest passes the PreliminaryValidation and the staff's notes are added to it using a modify unit, the LoanRequest is assigned the subsequent check activity, and PreliminaryValidation activity ends.

Check Activity

The staff work on security check activity to ensure that the applicant has a clear record with his bank and the justice department. For this activity, the lists of applications to be validated are retrieved by means of index units. Then, selecting an entry from the index unit starts the respective activity. The staff performs the required checks and fills in a form, which updates the current data for the application. If the checks have been successfully performed then the application is ready to be assigned to Final Approval activity.

Final Approval

In the FinalApproval activity, the staffs either reject the request, terminating the LoanRrequest activity, or approve it by filling in a form with the acceptance data. The form submission updates the approved LoanRequest and optionally connects to it a set of loan proposals. Finally, the LoanRequest is assigned to the Choice activity and then FinalApproval activity ends.

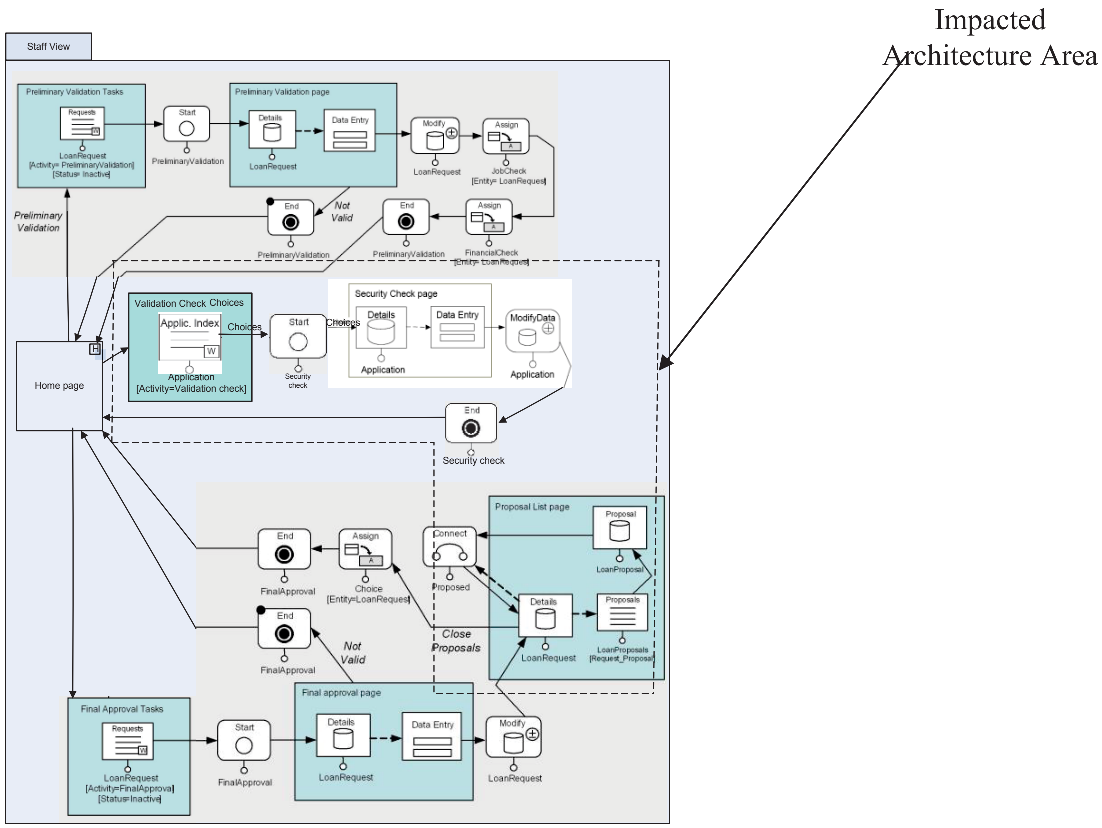


Figure E1: Navigational flow of loan issuance for staff view [Tailored version, adopted from (Acerbis et al., 2008)]

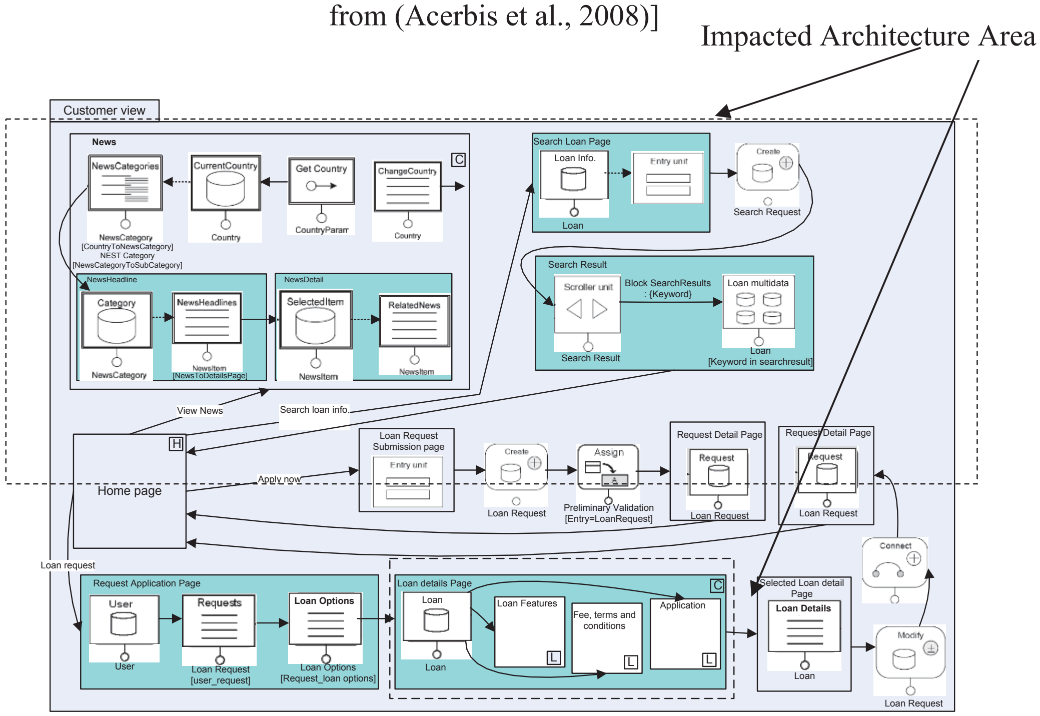


Figure E2: Navigational flow of loan issuance for customer view [Tailored version, adopted from (Acerbis et al., 2008)]

In next section we present a simplified illustration of hypertext model- describing the navigational design of online personal loan system by using WebML modelling notations (Acerbis et al., 2008). Typically in WebML, a hypertext model specifies system navigational flow and user navigation path that determines the execution and the composition of web pages representing the activities to be executed (Ceri et al., 2000b).

1.2 Overview of WebML and design models for exemplification

WebML describe a conceptual specification consisting of a data schema, describing application data, hypertexts and expressing the Web interface used to publish this data (Ceri et al., 2000b). WebML data model is the standard Entity-Relationship (E-R) model and widely used in general-purpose design tools. In WebML for the same data model, it is possible to define different hypertexts model (e.g., for different types of users or for different publishing devices) called site views. A site view is a graph of pages, allowing users from the corresponding group to perform their specific activities. Pages consist of connected units, representing at a conceptual level the atomic pieces of homogeneous information to be published. The content that a unit displays is extracted from an entity, selected by means of a selector, and validating complex logical conditions over the unit's entity. Units within a Web system design model are often related to each other through links carrying data from a unit to another, to allow the representation of the hypertext. Additionally, WebML also allows specifying update operations on the data underlying the structure (e.g., the creation/deletion of instances of an entity, or the creation and deletion of instances of a relationship). For a more details information about WebML readers are directed to (Acerbis et al., 2008, Brambilla et al., 2006, Ceri et al., 2000b). Based on WebML notations, a hypertext model (staff view and customer view) of loan request and approval activities are illustrated in figure E1and E2.

Figure E1 shows a fragment of WebML hypertext model for staff site view. From the home page three links allow the staff to reach three pages. First, where the preliminary validation process needs to be started. Once an application has been chosen, the preliminary validation starts for that application. Second-where the lists of the applications (for validation check) are shown by means of index units and security check activity starts. Third, where Loan Request is finally approved and optionally connects to a set of loan proposals.

Figure E2 shows a fragment of WebML hypertext model for the customer site view. From the home page four links allow the customer to reach four pages including view news, search loan, loan request submission and loan request application (after preliminary validation). First, from a particular news items, customer can further go to detail news and associated linked pages such as detail loan page. Second, customer can search loan information and browse the resulting information further. Third, a customer can request for a loan. However only after preliminary validation, fourthly, customer can access to loan request application page, and browse possible loan options and view other loan details such as loan features, fees, terms and conditions etc.

Both of the hypertext models in figure E1 and E2 tend to specify a specific aspect of information architecture i.e. navigational flow. Therefore, in the case study we will investigate the change impacts on information architectural design level resulting from changing business processes. In next section we will describe the proposed changes in the business processes of online personal loan system.

1.3 Changes in business processes

With the constant addition of new features in online personal loan system, discount rate has been introduced in the annual fee and interest rate of different types of loan. There were few changes also introduced in the eligibility criteria and additional terms & conditions along with normal loan request. To address these business process changes, necessary changes in information structure and navigational flow were also required. For example the introduction of discount rate and related information (such as new terms and conditions, eligibility criteria etc.) need an information structure and organization at the level of information architecture (IA). The necessary changes at information architecture are implemented accordingly to reflect the way gets changed business process will execute. Similarly, the introduction of discount rate also required to be communicated to customer prior to loan request. Thus the changes in business process necessary to be reflected at the level of information organisation, its access and its flow. To analyse the impact of changes at hypertext model (describing navigational flow) as illustrated in figure E1 and E2, we have used proposed process model of CIA on the given Web system and describe the instantiation in section 1.5, as follows. The main source of information while executing CIA process model was design decisions information that we have described in the next section.

1.4 CIA related Architecture Design information

In this case study, we have selected four design decisions from design decision repository, that address different navigational aspects such as (1) making information up-to-date and easy to explore, (2) helping users to search the desired information quickly (3) providing multiple navigation paths for different context, and (4) assisting users in knowing the current position while navigating. In the description of these design decisions, however, only design rules are documented, therefore we have only used design rules. These design rules supported us to discover the possible dependencies among design entities.

1.5 Instantiation of process Model of CIA

In the following section we have described three steps of our proposed CIA process model applied to this case study.

1.5.1 Step#1: Examine architectural change specifications and related information.

In step#1, three sources of information i.e. change request/information, architecture specifications and previous experience are used. From these sources of information, we have further derived two types of information (i) traceability information (traceability between Business Processes (BPs) and Information Architecture Requirements (IARs)) and (ii) dependency information (dependency between IARs and impacted design entities). In the presented case study, developers can determine the changes in business processes by analysing changes request/information form. The proposed changes in business processes are as:

- The introduction of discount rate during loan request (BP_loanrequest_01) is only applicable where applicant fulfilling the eligibility criteria as specified in BP_applic-criteria_03.
- The eligibility criteria include job check along with other normal check during loan request as a part of BP_loancheck_03.

- The changes to loan request features and their approval need to be communicated to the applicants by extending the communication mechanism specified in BP_communiaction_07.
- Search is being enhanced from simple search to selected search (BP_search_04) in order to facilitate specific search such as “reduced rate loan” etc

Traceability analysis - During traceability analysis, we have focused on the relationships between BPs and IARs and represented them in the traceability matrix, shown in figure E3. The relationships are expressed by putting a mark where the row of the business processes and the column of the architecture requirements intersect (① in figure 5.6).

The changes in the business processes of the given study can be scoped as (i) changes to loan activity, (ii) additional checks for eligibility of discount rate and (iii) communicate up-to-date changes to the customer. These three changes in business processes can also be used to determine the resulting scope of change in IA requirements by explicitly referring to the architecture specification and using previous experience. The set of IA requirements likely to be effected are highlighted in traceability matrix as shown in figure E3. These IA requirements include the mechanism to address an added feature ‘discount rate’ and its validation as a part of validation check choices. The changes to IA requirements are as:

IA Requirement / Business Process	⋮	**IA_Req_ loaninfo._06**	**IA_Req_ loan_05**	**IA_Req_ Communi. _07**	**IA_Req_ search_04**	⋮
…						
BP_loanrequest-01			✓			
BP_loanapproval-02			✓			
BP_applic-criteria_03		✓				
BP_communication_07			✓	✓		
BP_search_04					✓	
…						

Figure E3: Traceability Matrix

- The introduction of discount rate requires the presentation of discount rate information and eligibility criteria in IA_Req_Loaninfo._06 and its linkage to **loan details unit** (for customer view).
- The introduction of discount rate introduces additional **validation check choices, loan proposal entities** in IA_Req_loan_05 and further modification of information flow and its navigation during loan request and approval activity (IA_Req_communication_07).
- Communicating the additional set of information related to discount rate to applicants (IA_Req_communication_07) is required by means of **News component** (for customer view) in IA_Req_loan_05.
- In IA_Req_search_04, to facilitate specific search, **simple search unit** needs to be extended.

Analysing the traceability links further down to architecture design representation, is very beneficial to determine the impacted architecture area where the changes in IARs need to be implemented at the architecture design level (② in figure 5.6). By tracing each changed IA requirement to the level of architecture design specification, we have determined impacted architecture areas as shown by dotted lines in figure 8a-staff view and figure 8b-customer view. The details of how to determine the impacted architecture area is previously described in section 5.5.2.

Dependency information - After the identification of impacted architecture areas, dependency relationships between IARs and underpinning IA design entities (③ in figure 5.6) are explored. We have extracted the IA design entities by studying each IA requirements and picking up IA design entities as highlighted above in the bulleted IA requirements. All those design entities that correspond to changed IA requirements are considered as a starting impact set. For the ease of representation, we have assigned identity (as shown in table E2) to each IA design entities.

Table E2: List of design entities with their Ids

IA Design Entity	IDs
Validation check choice	DE1
Loan Proposal	DE2
Loan details unit	DE3
News components	DE4
Simple Search unit	DE5

Table E3: Dependencies between architecture requirements and design entities

Architecture Requirements	Design Entities
IA_Req_loan_05	DE1, DE2
IA_Req_loaninfo._06	DE3
IA_Req_Communi._07	DE4
IA_Req_search_04	DE5

Starting Impact Set (SIS) – The identification of traceability information (between BPs and IA requirements) and the dependency information between IARs and IA design entities allow us to identify the starting impact set as SIS = {DE1, DE2, DE3, DE4, DE5}

1.5.2 Step#2: Trace potential impacts. Following three sub-sections including finding design decisions, exploring design information, forward linking for design entity relationship describe the activities of step#2 and pre-requisite for tracing potential impacts of change on architecture design.

Finding Design Decisions – Six design decisions related to the set of design entities (i.e. SIS from setp#1) are retrieved from design decision repository (① in figure 5.7) by using search query. All those design decisions are selected where design entities (from SIS) are belong to and are referred in the description of design decisions. For example design entities such as News component, loan detail unit, and simple search (from SIS) belong to design decisions DD1, DD2 and DD4 respectively. However DD3, DD5 and DD6 are selected because one or more design entities from previously selected design decisions belong to DD3, DD5 or DD6. These design decisions will be analysed further to specifically focus on design rules.

Exploring Design Decisions information – In this case study, we found that design decisions repository was mostly comprised of design decisions that were typically used during the hypertext modelling. Further, from selected design decisions' descriptions, we have retrieved design rules as Web designers/architects have only documented design rules during design decision activity. These design rules were recorded in a form of natural language text. Design rules and their underlying design entities were further analysed (② in figure 5.7). These design rules acted as a connector and further revealed potential relationships among design entities. Table E4 provides the details of 6 selected design decisions and other information such as design decision identifier-DD#, Design entities-DEs and Design Rules.

Table E4: Design decision and other design information

Design Decision 1	
DD#	**DD-09: Pat_News_latestupdate**
DEs	**News component class, set-based class, News-presentation class, Navigation class**
Design Rules	1. DD-09_DR1: Each news component as presentation object is *navigated* from one news item to the other as a navigational context by using Set-Based Navigation. 2. DD-09_DR2: Each news component is *explored* by landmark and *specified* in more than one context and linked to the other context. For example from "latest update" context to "loan" context by using InContext Object (News).
Design Decision 2	
DD#	**DD-18: Pat_Landmark_section**
DEs	**Loan detail unit class , landmark class, menu class, set-based class, active reference class**
Design Rules	1. DD-18_DR1: The addresses list of all section (landmark) are passed as a list of parameters (hypermedia pointers) to the controller object (for example to *implement* loan detail unit) 2. DD-18_DR2: Controller object *generate* a menu list and make the landmarks accessible from every node of that menu using a single click. 3. DD-18_DR3: Display method of controller object instantiate each menu list *specified* by set-based navigation and *referenced* by landmark Incontext and 4. DD-18_DR4: loan detail unit is represented by a landmark and the implementation of landmark is *referred* by active reference.
Design Decision 3	
DD#	**DD-21a: Pat_Set-basednav._context**
DEs	**Navigational class, set-based class, Incontext class**
Design Rules	1. DD-21_DR1: Each navigational object is *identified* by a set-based object by providing *access* to content through navigational link and *defined* by an InContext object for that particular item and context whenever the same object shared by other context. 2. DD-21_DR2: The semantic relationship between set-based objects *supports* inter and intra –set navigation facilities, such as indexes and links for letting the user navigate to the 'next' and 'previous' elements of the current one in his traversal.
Design Decision 4	
DD#	**DD-29: Pat_Simplesea_search**
DEs	**Navigational class, simplesearch class, set-based navigation class**
Design Rules	1. DD-29_DR1: Simple search objects can be *extended* with additional method of getcategory() to support selected search where user has the option to select a category in which the search will be performed. 2. DD-29_DR2: Simple search based objects are a set of navigational objects in which the search string may appear. These navigational objects are *explored* by the set-based navigation pattern, *represented* by active reference objects and viewed as a set of related presentable objects to show the pages containing the search result in browser.
Design Decision 5	
DD#	**DD-21b: Incontext_sharedcontext**
DEs	**Incontext class, navigational class, semantic link class,**

Design Rules	1. DD-21b_DR1: InContext object defined as subclasses of InContextSequential, inherit anchors for sequential navigation and for backtracking to the context index.
Design Decision 6	
DD#	**DD-22: Active Reference**
DEs	**Incontext class, navigational class, semantic link class,**
Design Rules	1. DD-22_DR1: Each controller object keeps track of the user's current position in navigating the *hierarchy of information* and pass on this current position to active reference object. 2. DD-22_DR2: Each active reference object keep track of the user's navigation path (parameters) starting from the root of the hierarchy to current position (receives the user's current position from the Controller) and generates a presentational object showing the whole navigation path.

Table E5: Design information extracted from table E4

Design Rule	Design entity		
DD-09_DR1	News component	Set-based navigation	Presentation Object
DD-09_DR2	News component	InContext (News)	Landmark
DD-18_DR1	Landmark	Addresslist	Landmark Controller
DD-18_DR2	Landmark Controller	menuelist	
DD-18_DR3	Landmark/menu list	Landmark InContext	Landmark Set-based Navigational link
DD-18_DR4	Loan detail unit	Landmark	Landmark Active Reference
DD-21a_DR1	Navigational Object	Navigational link	InContext Object
DD-21a_DR2	Navigation link	Inter_navigational link	Indexes (Intra-navigational links)
DD-29_DR1	Selectedsearch Object	Simplesearch Object	
DD-29_DD2	Selectedsearch Object	Selectedsearch Set-based Navigation	Selectedsearch Active reference objects
DD-21b_DR1	InContext Object	InContext Sequential Object	
DD-22_DR1	Active reference Controller Object	Active reference object	
DD-22_DR2	Active reference object	Presentational object	

***Forward tracing of design entity relationships* –** We have discussed in section 5.3 that design rules can form the basis to analyse design entity relationships. By forward tracing, as shown in figure E4, design entity relationships among design entities are revealed from underlying design rules and can be represented by directed links (③ in figure 5.7). The design entity relationships are explored by textual analyses. The procedure followed for textual analysis is first reading the textual description of design rules and then from each design rules pick DEs from the description of those design rules. As a result, we end up with a number of DEs, as shown in table E5, that are related to each other under a design rule. Interestingly, note that there is not any design rule associated with design entities such as validation check choices-DE1 and loan proposal list-DE2, the two entities from SIS. This means that DE1 and DE2 are not related to each other and may

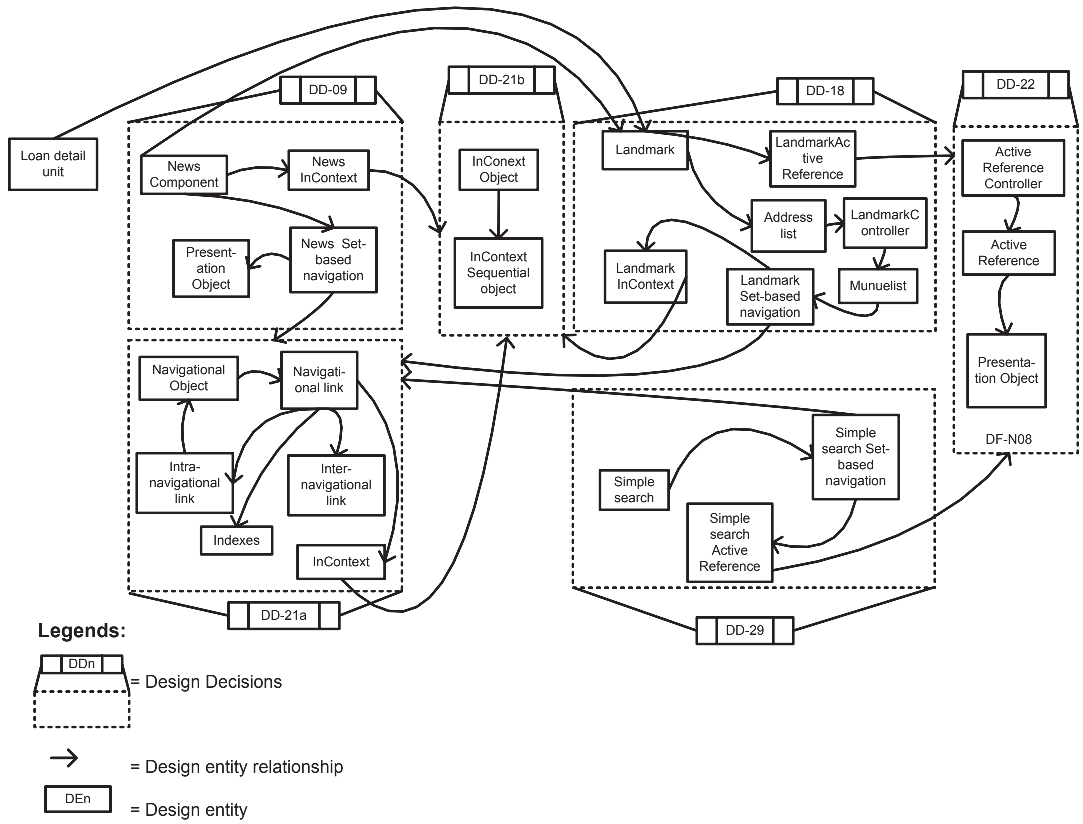


Figure E4: Forward tracing of design entity relationships

not reveal any ripple effect when they actually are being modified. Therefore we have not considered DE1 and DE2 further for analysis. However, other design entities from table E3 such as loan detail unit-DE3, News component-DE4 and simple search unit-DE5 had associated design rules. Therefore only these three design entities have been considered during the identification of design entity relationships (figure E4) and further will be considered for dependency analysis (figure E5).

Backward tracing for dependency graph – By backward tracing of design entity relationship (as shown in ③ of figure 5.7), a dependency graph has been developed (④ in figure 5.7) to represent the possible dependencies among design entities. For the ease of representation, we have assigned identity to each design entity of figure E4 as shown in table E6, and further used them to develop the dependency graph. As shown in figure 11, each node in the graph represents a design entity, and each edge in the graph represents dependency relationship between two design entities. Additionally, the dotted edges (such as e25,12 and e24,23) indicate that e24,23 do not reveal any dependency relationship from *InContext Sequential Object* (DE24) to *InContext Object*(DE23) backward traced for identified design entity relationship from design rule DD-21b_DR2. The underlying design rule for DE24 and DE23 states that 'Instantiate anchors for sequential navigation and for backtracking from the context object'. From this design rule it can be understood that with the instantiation of anchor from *InContext Sequential Object* (DE23) there is a dependency with *InContext Object*(DE24) as there is only a parameter pass required. Further if parameter get changed there is no effect on anchor instantiation and therefore there is no dependency as such. Indeed, there is a design entity relationship between *InContext Object* and *InContext Sequential Object* but there is no dependency between them while backward traced. Similarly, e25,12 do not reveal any dependency relationship from *Active reference controller*(DE25) to *Landmark Active reference*(DE12) while backward traced for identified design entity relationship from design rule. The underlying design rule for DE12 and DE25 states that 'Controller object keeps track of the user's current position in navigating the hierarchy of information'. Again here is the parameter passing between *Landmark Active reference* and *Active reference controller.* If there are more landmarks being introduced (as the case of discount rate loan) the *active reference controller* object keeps track of current position and displays it. Indeed, there is a design entity relationship between *Landmark Active reference* and *Active reference controller* but there is no dependency between them while backward traced.

Table E6: List of Design entities with their Ids for figure E4 and figure E5

Design entity	**IDs**	**Design entity**	**IDs**
News component	DE4	Simple search Set-based navigation	DE16
News InContext	DE6	Simple search Active reference	DE17
Presentation Objects	DE7	Navigational Objects	DE18
News Set-based navigation	DE8	Navigational Link	DE19
Reduce rate Landmark	DE9	Intra-navigational link	DE20
Landmark InContext	DE10	Inter-navigational link	DE21
Landmark set-based navigation	DE11	Indexes	DE22
Landmark Active reference	DE12	InContext Object	DE23
Addresslist	DE13	InContext SequentialObject	DE24
Munuelist	DE14	Active Reference Controller	DE25
Landmark Controller	DE15	Active Reference	DE26
Simple search	DE5	Presentation Object	DE27

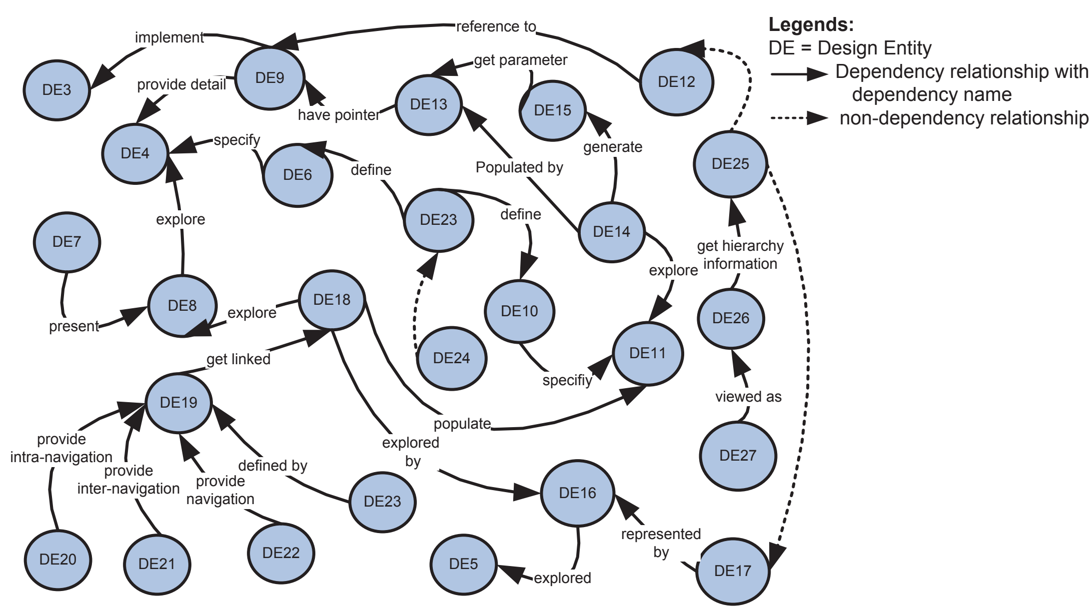


Figure E5: Dependency graph- Backward tracing of design decision information

Once the dependency graph has been developed, the potential impacts of changes (i.e. set of estimated impacts on IA design) are determined by taking into account every design entity that depends on other design entities directly and indirectly. For example impacted design entity such as DE13 can be determined with direct dependency (represented by e14,13) from DE14 to DE13 where as DE4 can only be determined by using chains of dependency, from DE14 to DE4 such as DE14→DE13→DE9→DE4. We have replaced the edge 'e' with the name of the dependency relationship to better understand the dependency graph. We have determined the name of dependencies from design rule that were documented in natural language text and represented in italic in table E4.

***Estimated Impact Set (EIS)* –** This is the set of design entities that are estimated to be affected after exploring the dependencies among design entities through dependency graph. This set consists of both primary impact (i.e. SIS) and secondary impacts. Thus EIS subsumed the SIS and therefore can be seen as an expansion of the SIS. Considering SIS and further exploring other possible dependent design entities, as shown in figure 10, it is determined that,

EIS = {DE1, DE2, DE3, DE4, DE5, DE6, DE7, DE8, DE9, DE10, DE11, DE12, DE13, DE14, DE15, DE16, DE17, DE18, DE19, DE20, DE21, DE22, DE23, DE24, DE25, DE26, DE27, DE28}

1.5.3 Step#3: Perform architectural changes. The design entities identified (as EIS) in step#2 are specifically modified by making changes at architecture design representation. This execution validated the EIS and might further discover other impacted design entity (if any). Therefore, while a change is being performed at architecture design, there may likely to be other impact sets categorised as (i) Discovered Impact Set (DIS)- represents an under estimation of impacts, and (ii) False Positive Impact Set (FPIS)- represents an over-estimation of impacts.

***Actual Impact Set (AIS)* –** The output of process model of CIA is Actual Impact Set (AIS) that is the set of design entities actually modified as a result of implemented changes. The EIS plus additions of the DIS and deletions of FBIS represents the AIS. In this case study after making appropriate modifications at IA design, we had neither identified any new impacted IA design entity nor falsified any already estimated IA design entities from EIS. Therefore both EIS and DIS are null set. Neither under-estimation nor over-estimation indicates the accuracy of EIS as determined in step#2. Thus AIS can be presented as below.

AIS = EIS = {DE1, DE2, DE3, DE4, DE5, DE6, DE7, DE8, DE9, DE10, DE11, DE12, DE13, DE14, DE15, DE16, DE17, DE18, DE19, DE20, DE21, DE22, DE23, DE24, DE25, DE26, DE27, DE28}

This impact set represents the design entities that are impacted due to initial change made in business process i.e. introduction of discount rate in Web based personal loan system. The instantiation of PMCIA supports to identify impacts on architecture resulting from business processes changes as discussed in section 5.6. Thus the instantiation of process model of CIA provides an indicative proof of concept to support early identification of change impact in Web systems.

In Web based personal loan system, Web designers/architects had maintained design specifications and documented the design decisions to describe what has been designed and why. However, it was not possible to systematically explore and investigate the implicit relationships among design entities that were revealed from design rules. From design decision repository, we determined that each design decision included the details such as design decision id, participating design entities during design decision, design rules, design rationale, trade-off etc. Through instantiation of PMCIA, We successfully employ design rules to investigate the possible dependencies between design entities and subsequently identify impacts on IA of personal loan system.

We have illustrated our proposed process model through an example case study that reasonably provides an initial demonstration of its feasibility. The dependencies were derived from design rules and then further used for impact identification, particularly when the modification of a design entity had consequences to other related design entities.

Appendix F

Consent Form

UNIVERSITY OF TECHNOLOGY, SYDNEY
CONSENT FORM for INTERVIEWS IN CASE STUDY

I ___________________ agree to participate in the research project entitled "A Process Model of Change Impact Analysis for Web Systems Development [UTS HREC 2008-2005]" being conducted by Zafar Mehboob (Faculty of Engineering and Information Technology, zafar@it.uts.edu.au, contact no. 95144446).

I understand that the purpose of this study is to evaluate the effectiveness of the proposed research approach in supporting early identification of change impacts during the Web system development process.

I understand that my participation in this research will involve 1-2 hours of responding to a pre and post-study interview questionnaire and would be recorded (audio-taped) if required. I also understand that there will be no harm or risk for me in this research.

I am aware that I can contact Prof. Didar Zowghi (Didar@it.uts.edu.au), Prof. David Lowe (david.lowe@uts.edu.au) or Zafar Mehboob (zafar@it.uts.edu.au) if I have any concerns about the research. I also understand that I am free to withdraw my participation from this research project at any time and without giving any reason if I feel uncomfortable of the questions asked on the questionnaires.

I agree that the research data gathered from this project may be published in a form that does not identify me in any way.

____ ____/____/____
Signed by

____ ____/____/____
Witnessed by

NOTE:
This study has been approved by the University of Technology, Sydney Human Research Ethics Committee. If you have any complaints or reservations about any aspect of your participation in this research which you cannot resolve with the researcher, you may contact the Ethics Committee through the Research Ethics Officer (ph: +61 2 9514 9772 Research.Ethics@uts.edu.au <mailto:Research.Ethics@uts.edu.au>), and quote the UTS HREC reference number. Any complaint you make will be treated in confidence and investigated fully and you will be informed of the outcome.

Appendix G

Evaluation of Change Impact Analysis Approach
Pre-Study Interview Questionnaire

About the Interview

The purpose of this interview is to investigate Web designers/architects' views and opinions about current change impact analysis approach at WSD organization. It has been decided to perform an evaluation for measuring the current change impact analysis (CIA) approach as a baseline.

- The information you provide will contribute to research being carried out by UTS researchers. Additionally, it will also help your organisation to evaluate and improve its change impact analysis approach.
- There are 3 sections and it will takes 40-60 minutes of your time to answer all questions to complete this interview
- There is no right or wrong answer; we are seeking your opinion and perception about the current change impact analysis approach. Your opinion is important to us.
- Remember, the interviewees' details are strictly confidential. All responses will be aggregated; no individual will be identified from the aggregated responses.

Thanks for your consent to participate in this interview.

NOTE:
This study has been approved by the University of Technology, Sydney Human Research Ethics Committee. If you have any complaints or reservations about any aspect of your participation in this research which you cannot resolve with the researcher, you may contact the Ethics Committee through the Research Ethics Officer (ph: +61 2 9514 9772 Research.Ethics@uts.edu.au <mailto:Research.Ethics@uts.edu.au>), and quote the UTS HREC reference number. Any complaint you make will be treated in confidence and investigated fully and you will be informed of the outcome.

Section 1: Participant information

1. Participant name:

2. Participant role/responsibility in project:

3. Web development background: (e.g. domain , technologies used etc)

4. CIA experience at architecture level: (e.g. number of year of experience and number of projects etc)

5. Type of development on small, medium and large Web system using a wide variety of technologies:

Section 2: Change Impact analysis Approach

1. How would you rate your understanding of the current change impact analysis (CIA) approach?

   □ Excellent □ Very Good □ Good □ Fair □ Not sure

2. To what extent is the CIA approach important for you while assessing the impacts on architecture level and why?

   □ Very important □ Important □ Somewhat important □ Not important □ Not sure

3. How do you analyse change propagation/change effect/change impact? Do you use any informal way (e.g. expert judgment, previous historical data, and previous experience etc) or any other formal method or techniques? Please explain

4. Do you have a standard set of steps/activities or a process to trace change impacts from business process to Information Architecture? If yes please explain it, if not then explain how impacts are analysed?

5. What information is available in relation to CIA and what information do you use during analysis activity? (e.g. traceability/dependency information, project documentation, historical change data etc)

6. To what extent, CIA approach addresses the two characteristics of Web systems. i.e. (i) co-evolution of business processes and solution under development, (ii) tighter linkage between business process and supporting architecture.

7. Please indicate the importance of the following items in relation to current CIA approach (prioritize 1= Very important, 2=Important 3= Somewhat important 4= Not important 5=Not sure) and why?

   - Identification of change impacts on Information Architecture resulting from changing business process.

- Identification of how architecture changes affect on other part of architecture design
- Support for early identification of change impacts
- Structured, consistent and organised way of performing impact analysis activity

8. Please indicate the extent to which you agree (disagree) with each of the following statements in relation to current CIA approach. (1= strongly agree, 2= agree, 3= neutral, 4= disagree, 5= strongly disagree)

   - Do current CIA approach support to links between business process and Information Architecture(IA) entities to maintain the traceability between them
   - Is sufficient information available to trace down to impacted IA area of the proposed change
   - Is sufficient information available to identify the ripple effect at architecture design level
   - Support to employ design decisions information to investigate the possible dependencies between architecture design entities

9. Does any approach/technique/method (or any form of it) help you to identify early change impact at architecture design?

10. By using a approach/technique/method (or any form of it) do you meet your expectation/goals or as you have described in Question 6, 7, and 8?

<u>Section 3: Process Aspect of Change Impact Analysis</u>

11. Please indicate the extent to which you agree (disagree) with each of the following statements. (1= strongly agree, 2= agree, 3= neutral, 4= disagree, 5= strongly disagree)

    - Current CIA approach is easy to understand
    - Current CIA approach is easy to use
    - Does current CIA approach provide steps to follow?
    - Does current CIA approach provide activity description?
    - In this project release has current CIA approach been instantiated?
    - In general has current CIA approach been instantiated at organisation level for system development?
    - Do current CIA approach provide feedback mechanism to evaluate the result of impact set after re-testing affected architecture design

12. Are you confident in identifying the impact at IA level based on the supported information and adopted CIA approach? (If Yes Why? If not then Why not?)

13. How would you rate the ease of assessing change impacts based on the information available during CIA approach?

□ Very poor □ Poor □ Good □ Very Good □ Excellent

14. Do you believe that you need assistance in understanding and identifying the impacts on IA resulting from changes in business processes? If yes why and if not why not?

15. What is the nature of the assistance that you need in 4 above?

Section 4: Additional Comments

Please add any additional comments you may have on the current CIA approach and adopted practices.

Appendix H

Evaluation of Process Model for Change Impact Analysis

Post-Study Interview Questionnaire

About the Interview

Recently an extended version of change impact analysis process has been introduced and instantiated in current release of your project at WSD organization. In order to identify the benefits of new implemented process model for CIA, it is necessary to conduct a post-study interview on the feedback and usage of new process.

- The information you provide will contribute to research being carried out by UTS researchers. Additionally, it will also help your organisation to evaluate and improve its change impact analysis approach.
- There are 4 section and it will takes 60-90 minutes of your time to answer all questions to complete this interview
- There is no right or wrong answer; we are seeking your opinion and perception about the newly implemented process model for CIA. Your opinion is important to us.
- Remember, the interviewees' details are strictly confidential. All responses will be aggregated; no individual will be identified from the aggregated responses.

Thanks for your consent to participate in this interview.

NOTE:
This study has been approved by the University of Technology, Sydney Human Research Ethics Committee. If you have any complaints or reservations about any aspect of your participation in this research which you cannot resolve with the researcher, you may contact the Ethics Committee through the Research Ethics Officer (ph: +61 2 9514 9772 Research.Ethics@uts.edu.au <mailto:Research.Ethics@uts.edu.au>), and quote the UTS HREC reference number. Any complaint you make will be treated in confidence and investigated fully and you will be informed of the outcome.

Section 1: Participant information

1. Participant name:

2. Participant role/responsibility in project:

3. Web development background: (e.g. domain , technologies used etc)

4. CIA experience at architecture level: (e.g. number of year of experience and number of projects etc)

5. Type of development on small, medium and large Web systems using a wide variety of technologies:

Section 2: Process model for Change Impact Analysis as a new approach and its comparison with current organisational practice

1. How did you apply process model for CIA while assessing change impacts on architecture?

2. How did you analyse change impact at architecture level with process model for CIA? and without the process model ? Please explain

3. What information did you use during analysis activity? (e.g. traceability/dependency information, project documentation, design decision repository, historical change data etc)

4. Do you think your organisation should adopt process model for CIA as an extension of current CIA approaches to identify impacts on Web systems' architecture (such as IA)? If yes, why? If not. Why not?

5. To what extent, the process model for CIA addresses the two characteristics of Web systems. i.e. (i) co-evolution of business processes and solution under development, (ii) tighter linkage between business process and supporting architecture.

6. Please indicate the extent to which you agree (disagree) with each of the following statements in relation to the new process model for CIA (1= strongly agree, 2= agree, 3= neutral, 4= disagree, 5= strongly disagree)

    - Support to link between business process and Information Architecture(IA) entities to maintain the traceability between them
    - Sufficient information is available to trace down to impacted IA area of the proposed change
    - Sufficient information is available to identify the ripple effect at architecture design level
    - Support to employ design decisions information to investigate the possible dependencies between architecture design entities

7. Does process model for CIA help you to identify early change impact at architecture design?

Section 3: Process Aspect of Change Impact Analysis

8. Please indicate the extent to which you agree (disagree) with each of the following statements in relation to the new process model for CIA. (1= strongly agree, 2= agree, 3= neutral, 4= disagree, 5= strongly disagree)

    - The steps are easy to understand
    - The steps' activities are easy to apply (such as traceability matrix, forward tracing and dependency graph)
    - Support for early identification of change impacts
    - Structured, consistent and organised way of performing impact analysis activity
    - Provide activity description and steps to follow?
    - Can be instantiated at organisation level for system development.
    - Support feedback mechanism to evaluate the result of impact set after re-testing affected architecture design

9. Are you confident in identifying the impact at IA level based on supported information and adopted process model for CIA? (If Yes Why? If not then Why not?)

10. How would you rate the ease of assessing change impacts based on available information for process model?

    □ Very poor □ Poor □ Good □ Very Good □ Excellent

11. Do you believe that the process model as an extension of CIA approaches assists you in understanding and identifying the impacts on IA resulting from changes in business processes? If yes why and if not why not?

Section 4: CIA Process Model and its applicability

12. Can you clarify why and how the new CIA process model can assist you in identifying early impacts at IA level?

13. Do you think the new process model for CIA addresses the deficiencies identified in pre-study interviews or in old CIA approaches?

14. By using the new process model can the impacts on IA be properly identified resulting from changing business needs?

15. Does the new CIA process model assists you in identify more of impacts on IA level than the old CIA approaches?

16. Do you think that Web designer/architect spend less time to identify the change impacts on IA by using the new CIA process model ?

17. Does the new CIA process model help you to identify change impacts on IA earlier than the old CIA approaches?

Section 5: Additional Comments

18. Please add any additional comments you may have on the adopted new process model for CIA.

Appendix I

In the following section we describe the proposed business processes in Web system *Project Release_B* selected in this case study. Additionally we execute each step of PMCIA for the selected Web system project.

1. Proposed Business processes changes

The following are business processes changes and their details that were extracted from RequistPro®. These changes were approved and scheduled (as described in project plan) for implementation during *Project Release_B*. Following are the list of changes in business processes.

BP_01_003: The business process gets changed to support a collaborative clinical system. A patient has the facility and can visit to any other clinic where the relevant doctors offer his/her services irrespective of the appointment made in any other clinic.

BP_02_004: To support privacy of information flow in a collaborative environment, some dynamic control information moved to label-based access control policy (LBAC) instead of role-based access control (RBAC) policy. LBAC policy patient name as a label and allow clinical staff/doctor from any clinic to gain access and search for patient medical records; and also able to navigate within a patient context (contextual navigation).

Due to the changes made at BP_02_004, a additional sub-requirement were introduced in BP_02_004 and described as:

BP_02_004a: Contextual navigation is allowed within related clinical data including doctors consulted, clinic visited, clinical reports, laboratory report etc.

2. Step#1 : Examine Architectural Changes

In order to address above changes in business processes, necessary changes need to be made in architecture requirements of the system. This was done by tracing the links from business processes to information architecture requirements through RequistPro® tool. During step#1 of process of CIA, for the traceability analysis, the relationships between changed Business Processes and impacted Information Architecture Requirements are identified as represented by traceability matrix, shown in figure I1.

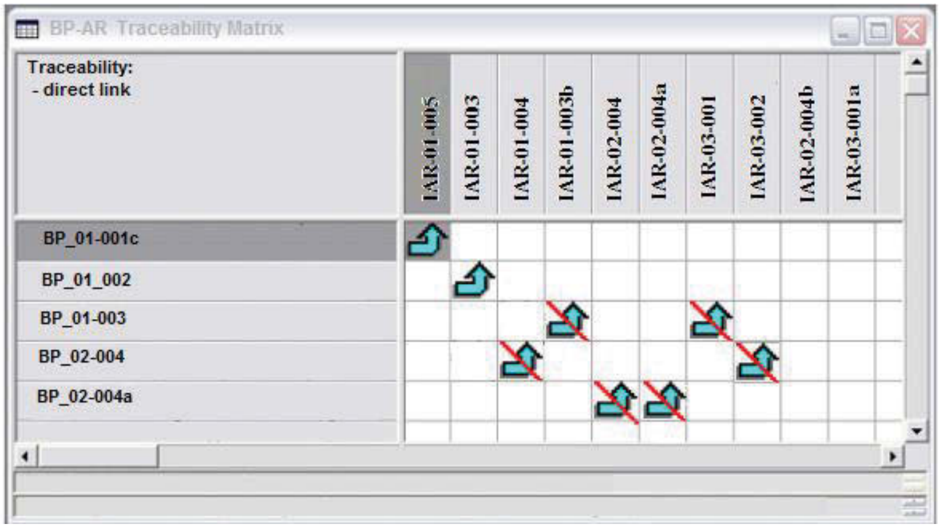


Figure I1: Traceability Matrix showing the relationships between changed business processes and impacted Information Architecture requirements (adopted from *Project Release_B* of CMS project- from WSD)

The following are the list of the changes made at architecture requirements specification:

IAR_01_003b: For labelled-based access control policy (LBACP), control access from *Patient* records are assigned to the permitted *Clinical staff.* Therefore the relationship of *Patient* record objects with *Clinical staff* objects need to be changed to cater the access rights and privacy of information flow.

IAR_02_004a: To support the navigation within the *patient context*, contextual navigation is introduced by using in-house navigational context component. The contextual navigation component provides support both for *doctors* and *clinical staff* to explore the related patient context such as previous doctors visited, clinic visited, clinical reports and laboratory reports etc.

IAR_02_004b: Due to the introduction of Contextual Navigation, the *DescriptionArea* now consists of *CategorizationSchema*, *ContentDescription*, *Category* and *ContextualNavigational* as a necessary vocabulary for content-based retrieval of information and documents.

IAR_03_001: Collaborative system environment introduced change where different version of *Services* at different clinic need to be bundled by a *Version Set* with additional *Region* classification. A specialization of *VersionSet* is the (logical) *Document*, which bundles different *DocumentVersion* across different clinics. Additionally, each *ServiceObject* has to link with region as an entry in the *DescriptiveObject*.

IAR_03_002: Following labeled-based access control policy (LBACP), each *Role* Object in the *ProcessArea* gets modified to address the labelled associated with each *Role*. Additionally changes will come in the *Action* and *ProcessTraces* of a PorcessArea and will be addressed by the introduction of *Breadcrumbs* design component.

2.1 Starting Impact Set

The dependency information was examined from each architecture requirement to the architecture design representation at the level where architecture requirements were modelled as architecture design representation. The examination of dependencies supported Web designers/architects to study all those design entities in architecture design representation on which a particular architecture requirement depends on. The following table was developed to represent the design entities on which impacted architecture requirements depend on. These design entities are the one thought to be initially affected by changing business processes. Therefore, we have the starting impact set as shown in second column of table I1.

IA Requirements	IA Design Entities
IAR_01_003b	patient, clinical staff
IAR_02_004a	patient context, doctor, clinical staff
IAR_02_004b	DescriptionArea, CategorizationSchema, ContentDescription, Category
IAR_03_001	Services, Version Set, Region, VersionSet, Document, DocumentVersion, ServiceObject, DescriptiveObject
IAR_03_002	Role, Action, ProcessTraces

Table I1: Dependencies between architecture requirements and design entities

3. Step#2: Trace potential impacts

The design decision information (i.e. design rules or design constraints) were determined related to each design entity (from starting impact set). The design rules and design constraints were extracted from design knowledge repository tool named as Archium. These design rules or design constraints supported Web designers/architects to exhibit dependency relationships among design entities. As shown in table I2, following are the list of six design decisions along with design rules and constraints that Web designer/architect had identified from design knowledge repository. For the sake of visibility each design entity is made italic and the relationships between design entities were made underlined that were there in the underlying design rules and design constraints.

Design Decision 1	
Design Decision ID	DD_SecPl_001
Design Entity	Patient, Context, Doctor, ClinicalStaff
Design Rules	1. *Patient's* records are associated with labels that specify to which *Contexts* these records are permitted to be exposed. 2. A *Context* can only be *Doctor* or *ClinicalStaff* type.
Design Rationale	In collaborative environment some dynamic control information could be moved to LBACP. When collaborative environment changes, it

	only needs to modify these control policies rather than the general policies.
Design Decision 2	
Design Decision ID	DD_DesA_002
Design Entity	DescriptiveObject, CategorizationScheme, ContentDescription, Category
Design Rules	1. Each descriptive area (*DescriptiveObject*) is represented by a *CategorizationScheme* consisting of *ContentDescription* and *Category*. 2. *CategorizationScheme* is an aggregation of *Category*. 3. *Context* is explained by *DescriptiveObject* as a necessary vocabulary for context and content-based retrieval of data/document.
Design Rationale	Descriptive area contains basic concepts for describing the content or role of information design objects on a high semantic level. This includes *ContentDescription* and *Categories*, which are grouped into *CategorizationSchemes*. Thus the descriptive area provides the necessary vocabulary for the context and content-based retrieval of data and documents.
Design Decision 3	
Design Decision ID	DD_ServA_001
Design Entity	Service, DocumentVersion, VersionSet, Document, ServiceObject, DescriptiveObject, StoragePlace, Store
Design Rules	**1.** *Service* are aggregated into *DocumentVersion*. The different version of services at a clinic can be bundled by a *VersionSet*. A specialization of *VersionSet* is the (logical) *Document*, which bundles different *DocumentVersion*. **2.** Each *Service* are described by *DescriptiveObject* **3.** Each *VersionSet* is stored at *StoragePlace* information unit.
Design Rationale	The service information area contains concepts for the description of the type and version history of electronic documents and other information resources, as well as their manual dependencies and their structured decomposition. The Service concept denotes all kind of information elements such as data items or decision representation objects. Storage area describes at which *StoragePlace* a particular *VersionSet* is located, documents management system or external tool it is stored. A Storage Place forms part of a store, such as a document management system, when a user edits a document with the appropriate tool, the changes can automatically be correlated with their representation.
Design Decision 4	
Design Decision ID	DD_Her_004
Design Entity	SourceDocument, DestinationDocument, Context
Design	1. Each *SourceDocument* on the system site should provide at least

Constraints	two but no more than five links to other related *DestinationDocuments* and at least one link that provide access to categories of related documents *Context*.
Design Rationale	Pre-defined hierarchical structure is used to improve the usability of the desired information search and access.
Design Decision 5	
Design Decision ID	DD_ProA_001
Design Entity	ProcessElement, UserElement, ProcessElement, ProcessTrace, Action
Design Rules	**1.** *ProcessElement* define *Action* and *ProcessTrace*, where each *ProcessTrace* is characterized by *Action*. **2.** *UserElement* is defined by *Role* and *User*, where each user belongs to a *Role*.
Design Rationale	Process area contains the concepts needed to represent the *ProcessObjects* that create, use, or modify *ServiceObject*. They comprise general process definition (*Actions*) as well as *Processtraces* (information flow) resulting from concrete executions of the *Actions* by *Users*.
Design Decision 6	
Design Decision ID	DD_Sty_001
Design Entity	Context, Processtrace, DescriptiveObject
Design Constraints	1. The uniform pipe-and-filter architecture style constrained *Context*, *Processtrace* and *DescriptiveObject* to a single interface type i.e. "uniform component interface".
Design Rationale	The uniform pipe-and-filter style motivated by the generality principle, in order to obtain two desirable qualities that will become the architectural properties of reusable and configurable components when that style is instantiated within architecture. The pipe-and-filter style enables several positive architectural properties when used within a system that requires data transformations between components.

Table I2. Design decisions related to each design entity in SIS

3.1 Directed dependency graph

By forward tracing, as shown in figure I2a, Web designers/architects represented the design entity relationships as a directed links that supported to associate design entities together under a design rule and design constraints. As shown in figure I2b, by backward tracing of design entity relationships, Web designers/architects developed a dependency graph to represents the possible dependencies among design entities.

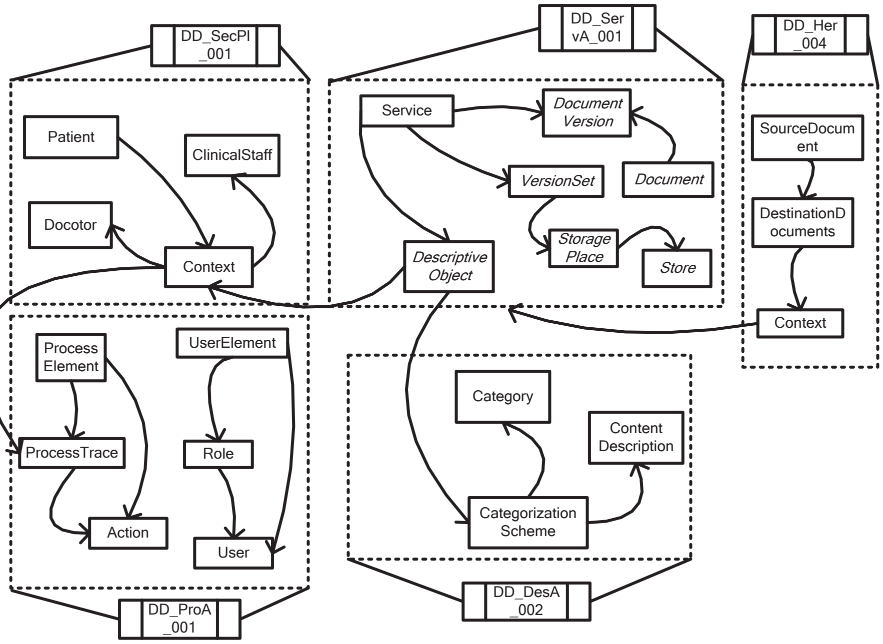


Figure I2a: Forward tracing for design entity relationships

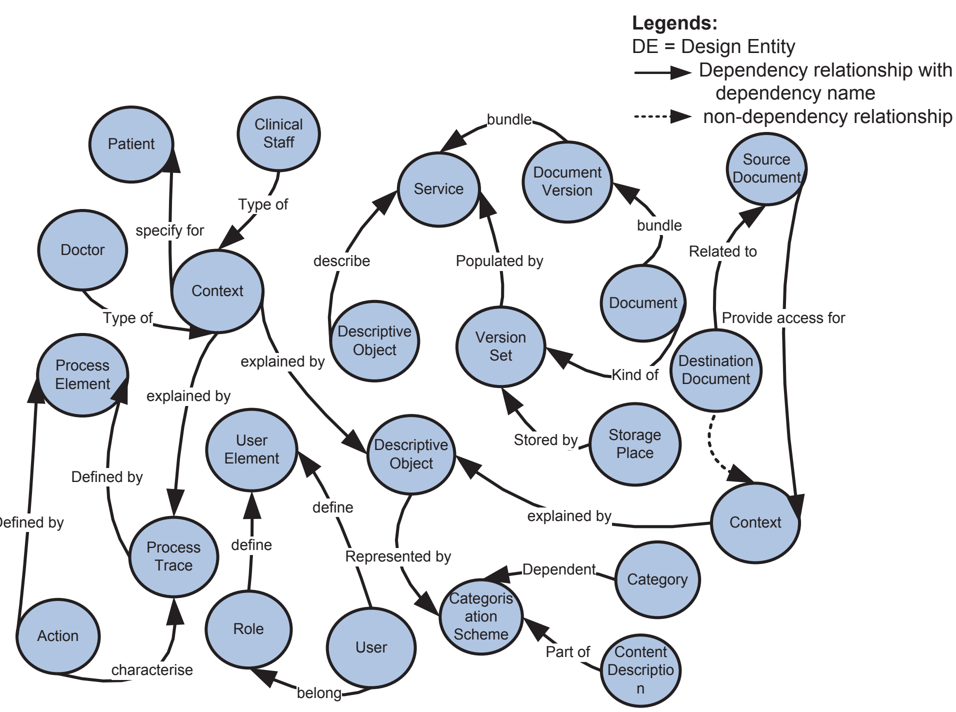


Figure I2b: Dependency graph- Backward tracing of design entity relationships

3.2 Estimated Impact Set (EIS#)

Once the dependency graph has been constructed, Web designers/architects determined the potential impacts of a change (i.e. set of estimated impacts on IA resulting from changing business processes) after taking into account every design entity that depends on other design entities directly and indirectly. For example impacted design entity such 'Context' can be determined with direct dependency (named as 'type of') from "ClinicalStaff" to 'Context' where as 'CategorisationScheme' can only be determined by using chains of dependency, from 'ClinicalStaff' to 'CategorisationScheme' such as ClinicalStaff→Context→DescriptiveObject→CategorisationScheme. The dependency name are specifically used in the dependency graph that helped Web designers/architects to better understand the dependencies and their severity impacted among design entities. Web designers/architects have determined the name of dependencies from design rule that were documented in natural language text and as represented in italic for the purpose of visualisation.

All those design entities that were related with other design entities by a dependency name are all the elements of estimated impact set- SIS. This is the set of design entities that are estimated to be affected after exploring the latent dependencies by means of dependency graph approach. This set consists of first order impacts (i.e. SIS) and n-order impacts, where 'n' is the number of intermediate dependencies between the source design entity and destination design entity. Thus EIS subsumed the SIS and therefore can be seen as an expansion of the SIS.

3.3 Prioritization of impacts

Indeed, prioritisation of impacts identified from PMCIA was done on request of case study organisation and were not part of the PMCIA as such.

As shown in figure I2b, in the directed graph there were 10 nodes, 1 to 10. Each node represents a design entity connected to other design entity of information architecture. Each design entity has a number of dependent design entities as well as design entities on which a particular design entity depends on. In graph theory in-degree and out-degree indicate the number of links coming in and going out from a particular node. We describe in-degree and out-degree in relation with a design entity as follows:

Fan-in: Fan-in or in-degree of a design entity indicates the number of known design entities depends on that particular design entity (Bohner and Arnold, 1996).

Fan-out: Fan-out or out-degree degree of a design entity indicates the number of known design entities on which a particular design entity depends on (Bohner and Arnold, 1996).

Design Entity	Fan-in (in-degree)	Fan-out (out-degree)	Prioritization of design entity
Patient	1	0	5
Clinical staff	0	1	5
Context Exposure	2	3	1
Doctor	0	1	5
Process Element	2	0	4
Action	0	2	4
Process Trace	2	1	3
Role	1	1	4
User	0	2	4
UserElement	2	0	4
Service	3	0	3
Descriptive Object	2	2	2

Categorisation Scheme	3	0	3
Category	0	1	5
Content Description	0	1	5
Version Set	2	1	3
Storage Place	0	1	
Document	0	2	4
Document Version	1	1	4
Destination Document	0	1	5
Context	1	1	4
Source Document	1	1	4

Table I3. Illustrating fan-in and fan-out for each design entity from EIS

Table I3 shows fan-in and fan-out of each design entity from the estimated impact set. Having knowledge of fan-in and fan-out for each design entity, Web designers/architects prioritised the impact set and thus had a more-informed decision making in relation to implementing the changes at architecture design level.

4. Step#3: Perform Architectural Changes

IA design entities identified (as EIS) were modified by Web designers/architects specifically by making changes at architecture design representation. This execution validated the EIS and further falsifying estimated design entity as a False Positive Impact. While performing architectural changes these steps were iterative in nature and used for discovering new design entities (i.e. DIS) and falsifying estimated design entities (i.e. False Positive Impact Set-FPIS) if any. After making appropriate modifications at IA design, Web designers/architects had neither falsified any estimated impact nor identified any new impacts. Indeed, neither discovering new design entities nor falsifying estimated design entities indicate the accuracy of EIS as discovered by using the dependency graph as shown figure I2b. It has been shown that by using PMCIA, estimated impact set (EIS) was accurate and had not been over-estimated nor under-estimated by Web designers/architects.

4.1 Actual Impact Set

The output of PMCIA was Actual Impact Set (AIS) that was the set of design entities actually modified as a result of implemented changes. The EIS plus additions of DIS and deletions of FBIS represents the AIS (Bohner and Arnold, 1996). Therefore architecture impact set can be represented as, $AIS = (EIS \cup DIS) \setminus FPIS$. In the case study, Web designers/architects had not identified any new impact as DIS and had not falsified any impacts as FPIS. Thus AIS can be presented as shown in table I3 with the prioritization of each design entity as an impacted element.

This impact set represents the IA design entities that were impacted due to resulting change in business process i.e. collaborative clinical system and information access control policy. By using PMCIA in *Project Release_B*, Web designers/architects successfully identity the early impacts as discussed in section 7.6.4. These early impacts served to address obvious need to maintain sophisticated and up-to-date information architecture that consequently leads to high degree of satisfied user interaction and thus a successful web-based clinical system. Additionally, based on the early impact set, Web designers/architects developed adequate knowledge about the consequences of making a business process change at information architecture level and thus minimised the complexity and difficulty in recognising and managing change impacts at later stages of project development.

References

AALST, W. M. P. V. D., BENATALLAH, B., CASATI, F., CURBERA, F. & VERBEEK, E. (2007) Business process management: Where business processes and web services meet. *Data & Knowledge Engineering,* 61**,** 1-5.

ACERBIS, R., BONGIO, A., BRAMBILLA, M., BUTTI, S., CERI, S. & FRATERNALI, P. (2008) Web Applications Design and Development with WebML and WebRatio 5.0 IN PAIGE, R. F. & MEYER, B. (Eds.) *Objects, Components, Models and Patterns* Berlin, Springer Berlin Heidelberg.

ADAM, S. & DOERR, J. (2008) How to better align BPM & SOA – Ideas on improving the transition between process design and deployment. *Workshop on Business Process Modeling, Development and Support (BPMDS-'08) in Conjunction with CAise'08.* Montpellier, France.

AGARWAL, R. & TANNIRU, M. R. (1990) Knowledge acquisition using structured interviewing: an empirical investigation. *J. Manage. Inf. Syst.,* 7**,** 123-140.

ALBIN, S. T. (2003) *The Art of Software Architecture: Design Methods and Techniques* Indianapolis, USA, Wiley publishing Inc.

ALLEN, R. J. (1997) A formal approach to software architecture. *School of Computer Science.* Pittsburgh, Carnegie Mellon.

AOYAMA, M. (2002) Metrics and analysis of software architecture evolution with discontinuity. *Proceedings of the International Workshop on Principles of Software Evolution.* Orlando, Florida, ACM.

ATHULA, G. (2002) Web engineering: managing the complexity of web systems development. *Proceedings of the 14th international conference on Software engineering and knowledge engineering.* Ischia, Italy, ACM.

AVERSANO, L., BODHUIN, T. & TORTORELLA, M. (2005) Assessment and impact analysis for aligning business processes and software systems. *Proceedings of the 2005 ACM symposium on Applied computing.* Santa Fe, New Mexico, ACM.

AVERSANO, L., CANFORA, G., CIMITILE, A. & LUCIA, A. D. (2001) Migrating Legacy Systems to the Web: an Experience Report. *Proceedings of the Fifth European Conference on Software Maintenance and Reengineering.* IEEE Computer Society.

AVERSANO, L. & TORTORELLA, M. (2009) Business Process-Aware Maintenance Task: A Preliminary Empirical Study. *Proceedings of the 2009 European Conference on Software Maintenance and Reengineering.* IEEE Computer Society.

AVGERIOU, P., KRUCHTEN, P., LAGO, P., GRISHAM, P. & PERRY, D. (2007) Architectural knowledge and rationale: issues, trends, challenges. *SIGSOFT Softw. Eng. Notes,* 32**,** 41-46.

BABAR, M. A., AND, I. G. & KITCHENHAM, B. (2006) A Framework for Supporting Architecture Knowledge and Rationale Management IN DUTOIT, A. H., MCCALL, R., AND, I. M. & PAECH, B. (Eds.) *Rationale Management in Software Engineering.* Berlin, Springer Berlin Heidelberg.

BABAR, M. A. & GORTON, I. (2007) A Tool for Managing Software Architecture Knowledge. *Proceedings of the Second Workshop on SHAring and Reusing architectural Knowledge Architecture, Rationale, and Design Intent.* IEEE Computer Society.

BABAR, M. A., NORTHWAY, A., GORTON, I., HEUER, P. & NGUYEN, T. (2008) Introducing Tool Support for Managing Architectural Knowledge: An Experience Report. *Proceedings of the 15th Annual IEEE International*

Conference and Workshop on the Engineering of Computer Based Systems. IEEE Computer Society.

BADRI, L., BADRI, M. & ST-YVES, D. (2005) Supporting Predictive Change Impact Analysis: A Control Call Graph Based Technique. *Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC'05).* Taipei, Taiwan, IEEE Computer Society.

BARESI, L., COLAZZO, S., MAINETTI, L. & MORASCA, S. (2006) W2000: A Modelling Notation for Complex Web Applications IN MENDES, E. & MOSELY, N. (Ed.) *Web Engineering,* Berlin, Springer Berlin Heidelberg

BARESI, L., GARZOTTO, F. & PAOLINI, P. (2001) Extending UML for Modeling Web Applications. *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34),* IEEE Computer Society.

BARNA, P., FRASINCAR, F., HOUBEN, G.-J. & VDOVJAK, R. (2003) Methodologies for Web Information System Design. *Proceedings of the International Conference on Information Technology: Computers and Communications.* IEEE Computer Society.

BASS, L., CLEMENTS, P. & KAZMAN, R. (1998) *Software architecture in practice*, Addison-Wesley Longman Publishing Co., Inc.

BASS, L., CLEMENTS, P. & KAZMAN, R. (2003) *Software Architecture in Practice,* Boston, MA, Pearson Education, Inc.

BASSON, H. (2001) A Change Propagation Model and Platform For Multi-Database Applications. *Proceedings of the IEEE International Conference on Software Maintenance (ICSM'01).* IEEE Computer Society.

BAZELEY, P. (2002) Issues in Mixing Qualitative and Quantitative Approaches to Research. IN BUBER, R., GADNER, J. & RICHARDS, L. (Eds.) *Applying qualitative methods to marketing management research.* Vienna, Austria, Palgrave Macmillan.

BENBASAT, I., GOLDSTEIN, D. K. & MEAD, M. (1987) The case research strategy in studies of information systems. *MIS Q.,* 11**,** 369-386.

BENGTSSON, P., LASSINGB, N., BOSCHC, J. & VLIETD, H. V. (2004) ALMA: Architecture-Level Modifiability Analysis. *Journal of. Systems and Software,* 69**,** 129-147.

BENNETT, K. H. & VACLAV, T. R. (2000) Software maintenance and evolution: a roadmap. *Proceedings of the Conference on The Future of Software Engineering.* Limerick, Ireland, ACM.

BERG, K. V. D. (2006) Change Impact Analysis of Crosscutting in Software Architectural Design. *Workshop on Architecture-Centric Evolution (ACE 2006) in collaboration with 20th European Conference on Object-Oriented Programming-ECOOP 2006.* Nantes, France.

BERG, K. V. D., CONEJERO, J. M. & HERNÁNDEZ, J. (2005) Disentangling crosscutting in AOSD: a conceptual framework. *2nd European Interactive Workshop on Aspects in Software.* Brussels, Belgium, ACM.

BOCHICCHIO, M. A. & LONGO, A. (2004) UWA+: bridging Web systems design and Business process modeling. *International Workshop on Web Engineering in conjunction with ACM Hypertext 2004.* Santa Cruz, California, USA ACM Computer Society.

BOCHICCHIO, M. A. & LONGO, A. (2005) Integrating Web Systems Design and Business Process Modeling. IN KOCH, N., VALLECILLO, A. & ROSSI, G. (Eds.) *Workshop on Model-driven Web Engineering in conjunction with 5th International Conference on Web Engineering-ICEW'05.* Sydney, Australia.

BOEHM, B. & BASILI, V. R. (2001) Software Defect Reduction Top 10 List. *Computer,* 34**,** 135-137.

BOER, F. S. D., BONSANGUE, M. M., GROENEWEGEN, L. P. J., STAM, A. W., STEVENS, S. & TORRE, L. V. D. (2005) Change impact analysis of enterprise architecture, information reuse and integration. *IEEE International Conference on on Information Reuse and Integration (IRI'05).* Las Vegas, Nevada, USA, IEEE Computer Society.

BOHNER, S. A. (1996) Impact analysis in the software change process: a year 2000 perspective *International Conference on Software Maintenance (ICSM '96).* Monterey, CA, USA, IEEE Computer Society

BOHNER, S. A. (2003) Extending Software Change Impact Analysis into COTS Components. *Proceedings of the 27 th Annual NASA Goddard/IEEE Software Engineering Workshop (SEW-27'02).* IEEE Computer Society.

BOHNER, S. A. & ARNOLD, R. S. (1996) *Software Change Imapct Analysis* Los Almitos, CA, USA, IEEE Computer Society Press.

BOOCH, G. (2001) The architecture of Web applications. *develoeperWorks.*

BOSCH, J. (2004) Software Architecture: The Next Step IN GOOS, G., HARTMANIS, J. & LEEUWEN, J. V. (Eds.) *First European Workshop on Software Architecture-EWSA'04*. St Andrews, UK, Springer Berlin / Heidelberg.

BOSE, P. (1998) Change analysis in an architectural model: a design rationale based approach. *Proceedings of the third international workshop on Software architecture.* Orlando, Florida, United States, ACM.

BRAMBILLA, M., CERI, S., COMAI, S., FRATERNALI, P. & MANOLESCU, I. (2003) Specification and Design of Workflow-Driven Hypertexts. *Journal of Web Engineering,* 1**,** 163-182

BRAMBILLA, M., CERI, S., FRATERNALI, P. & MANOLESCU, I. (2006) Process modeling in Web applications. *ACM Trans. Softw. Eng. Methodol.,* 15**,** 360-409.

BRAMBILLA, M., PRECIADO, J. C., LINAJE, M. & SANCHEZ-FIGUEROA, F. (2008) Business Process-Based Conceptual Design of Rich Internet Applications. *Proceedings of the 2008 Eighth International Conference on Web Engineering.* IEEE Computer Society.

BRATTHALL, L., JOHANSSON, E. & BJRN, R. (2000) Is a Design Rationale Vital when Predicting Change Impact? A Controlled Experiment on Software Architecture Evolution. *Proceedings of the Second International Conference on Product Focused Software Process Improvement.* Springer-Verlag.

BRATTHALL, L. & JORGENSEN, M. (2002) Can you Trust a Single Data Source Exploratory Software Engineering Case Study? *Empirical Software Engineering,* 7**,** 9-26.

BREWER, J. D. & HUNTER, A. (1989) *Multimethod Research: A Synthesis of Styles* ,Newbury Park, CA, Sage Publication Inc.

BUCKNELL, A., LOWE, D. & ZOWGHI, D. (2008) A Method for Analysing Web Systems Model Alignment. *Proceedings of the 14th Australian World Wide Web conference.* Ballina, Australia.

BURDMAN, J. (1999) *Collaborative Web development: strategies and best practices for Web teams*, Addison-Wesley Longman Ltd.

BURGE, J. E. & BROWN, D. C. (2006) Rationale-Based Support for Software Maintenance IN DUTOIT, A. H., MCCALL, R., MISTRÍK, I. & PAECH, B. (Eds.) *Rationale Management in Software Engineering.* Berlin, Springer Berlin Heidelberg.

BURGE, J. E. & BROWN, D. C. (2008) SEURAT: integrated rationale management. *Proceedings of the 30th international conference on Software engineering.* Leipzig, Germany, ACM.

BURNS, R. B. (2000) *Introduction to Research Methods,* Thousand Oaks, California, Sage Publications, Inc.

BURSTEIN, F., and GREGOR, S. (1999) The Systems Development or Engineering Approach to Research in Information Systems: An Action Research Perspective, IN HOPE B. & YOONG P. (Eds.) *10th Australasian Conference on Information Systems*, Victoria University of Wellington, New Zealand, PP-122-134.

CANFORA, G. & CERULO, L. (2006) Fine grained indexing of software repositories to support impact analysis *Proceedings of the 2006 international workshop on Mining software repositories* Shanghai, China ACM.

CAPILLA, R., DUEAS, J. C. & NAVA, F. (2010) Viability for codifying and documenting architectural design decisions with tool support. *J. Softw. Maint. Evol.,* 22**,** 81-119.

CAPILLA, R. & NAVA, F. (2008) Extending Software Architecting Processes with Decision-Making Activities IN MEYER, B., NAWROCKI, J. R. & BARTOSZWALTER (Eds.) *Balancing Agility and Formalism in Software Engineering.* Berlin, Springer

CAPILLA, R., NAVA, F. & CARRILLO, C. (2008a) Effort Estimation in Capturing Architectural Knowledge. *Proceedings of the 2008 23rd IEEE/ACM International Conference on Automated Software Engineering.* IEEE Computer Society.

CAPILLA, R., NAVA, F. & DUENAS, J. C. (2007) Modeling and Documenting the Evolution of Architectural Design Decisions. *Proceedings of the Second Workshop on SHAring and Reusing architectural Knowledge Architecture, Rationale, and Design Intent.* IEEE Computer Society.

CAPILLA, R., NAVA, F., MONTES, J. & CARRILLO, C. (2008b) ADDSS: Architecture Design Decision Support System Tool. *Proceedings of the 2008 23rd IEEE/ACM International Conference on Automated Software Engineering.* IEEE Computer Society.

CAPILLA, R., NAVA, F., P.REZ, S. & DUE, J. C. (2006) A web-based tool for managing architectural design decisions. *SIGSOFT Softw. Eng. Notes,* 31**,** 4.

CERI, B. S., FRATERNALI, P. & BONGIO, A. (2000a) Web Modeling Language (WebML): A Modeling Language for Designing Web Sites. *Ninth International World Wide Web Conference.* Amsterdam, Netherlands.

CERI, S., FRATERNALI, P. & BONGIO, A. (2000b) Web Modeling Language (WebML): a modeling language for designing Web sites. *Comput. Netw.,* 33**,** 137-157.

CERI, S., FRATERNALI, P. & MATERA, M. (2002) Conceptual Modeling of Data-Intensive Web Applications. *IEEE Internet Computing,* 6**,** 20-30.

CESARE, S. D. & SERRANO, A. (2006) Business Process and Information Systems Design Collaboration. *39th Annual Hawaii International Conference on System Sciences (HICSS'06).* Los Alamitos, California, IEEE Computer Society Press.

CHAOS (2009) *CHAOS Summary 2009: The 10 laws of CHAOS*, Boston, USA, Teh Standish Group International.

CHEN, C. L., SHAO, D. & PERRY, D. E. (2007) An Exploratory Case Study Using CBSP and Archium. *Proceedings of the Second Workshop on SHAring and Reusing architectural Knowledge Architecture, Rationale, and Design Intent.* IEEE Computer Society.

CHEN, W. K. (1997) *Graph Theory and its Engineering Applications,* Singapore, World Scientific Publishing Co. pvt Ltd.

CHEVALIERA, A. & IVORY, M. Y. (2003) Web site designs:Influence s of designer's expertise and design constraints. *Int. J. Human-Computer Studies,* 58**,** 57-87.

CHOI, H., CHOI, Y. & YEOM, K. (2006a) An Integrated Approach to Quality Achievement with Architectural Design Decisions. *Journal of Software,* 1**,** 40-49.

CHOI, H., YEOM, K., CHOI, Y. & MOON, M. (2006b) An Approach to Quality Achievement at the Architectural Level: AQUA. IN GORRIERI, R. & WEHRHEIM, H. (Eds.) *International Conference on Formal Methods for Open Object-Based Distributed Systems-FMOODS 2006.* Bologna, Italy, Springer.

CHRISTODOULOU, S. P., STYLIARAS, G. D. & PAPATHEODROU, T. S. (1998) Evaluation of hypermedia application development and management systems. *Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space.* Pittsburgh, Pennsylvania, United States, ACM.

CMMI (2006) CMMI for Systems Engineering/Software Engineering ver 1.2. CA, SEI-Carnegie Mellon University.

COHEN, L. & HOLLIDAY, M. (1996) *Practical Statistics for Students,* London, SAGA Publications Company, .

COLIN, P. (1993) Software-Engineering Research Revisited. *IEEE Softw.,* 10**,** 19-28.

CRESWELL, J. W. (2002) *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches,* California, Sage Publication Inc.

CRESWELL, J. W. (2007) *Qualitative Inquiry and Research Design.* London, Sage Publication Inc.

CRESWELL, J. W. (2009) *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches* Thousand Oaks, California, Sage Publications, Inc.

CROTTY, M. (1998) *The Foundations of Social Research: Meaning and Perspective in the Research Process,* St Leonard, NSW, Allen & Unwin.

CUNNINGHAM, J. B. (1997) Case study principles for different types of cases *Quality and Quantity,* 31**,** 401-423.

CUSUMANO, M., MACCORMACK, A., KEMERER, C. F. & CRANDALL, W. (2003) Software Development Worldwide: The State of the Practice. *IEEE Software,* 20.

DAVENPORT, T. H. (1993) *Process Innovation,* Boston, MA, Harvard Business School Press.

DAVID, M. & SUTTON, C. D. (2004) *Social Research* Thousand Oaks, CA, Sage Publications Inc.

DAVIDE, F., GIOVANNI, C. & PHILIPPE, K. (2008a) Value-Based Design Decision Rationale Documentation: Principles and Empirical Feasibility Study. *Proceedings of the Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008).* IEEE Computer Society.

DAVIDE, F., RAFAEL, C. & GIOVANNI, C. (2008b) A value-based approach for documenting design decisions rationale: a replicated experiment. *Proceedings of the 3rd international workshop on Sharing and reusing architectural knowledge.* Leipzig, Germany, ACM.

DEMARCO, T. (1995) On System Architecture. *Monterey Workshop on Specification-Based Software Architectures.* Monterey, CA.

DESHPANDE, Y. & HANSEN, S. (2001) Web Engineering: Creating a Discipline among Disciplines. *IEEE MultiMedia,* 8**,** 82-87.

DESHPANDE, Y., MURUGESAN, S., GINIGE, A., HANSEN, S., SCHWABE, D., GAEDKE, M. & WHITE, B. (2002) Web Engineering. *Journal of Web Engineering,* 1**,** 3-17.

DEY, I. (1993) *Qualitative Data Analysis A User-Friendly Guide for Social Scientist,* New York, Routlege- Taylor and Francis group.

DISTANTE, D., CANFORA, G., TILLEY, S. & HUANG, S. (2006) Redesigning legacy applications for the web with UWAT+: a case study. *Proceedings of the 28th international conference on Software engineering.* Shanghai, China, ACM.

DISTANTE, D., ROSSI, G. & CANFORA, G. (2007a) Modeling business processes in web applications: An analysis framework. *Proceedings of the 2007 ACM symposium on Applied computing.* Seoul, Korea, ACM.

DISTANTE, D., ROSSI, G., CANFORA, G. & TILLEY, S. (2007b) A comprehensive design model for integrating business processes in web applications. *International Journal of Web Engineering and Technology,* 3**,** 43 - 72

DISTANTE, D. & TILLEY, S. (2005) Conceptual Modeling of Web Application Transactions: Towards a Revised and Extended Version of the UWA Transaction Design Model. *Proceedings of the 11th International Multimedia Modelling Conference.* IEEE Computer Society.

DUEÑAS, J. C. & CAPILLA, R. (2005) The Decision View of Software Architecture *European Workshop on Software Architecture -EWSA'05.* Pisa, Italy, Springer Berlin / Heidelberg.

EASTERBROOK, S. (2007) Empirical research methods for software engineering. *In Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering* (ASE '07). ACM, New York, NY, USA.

EASTERBROOK, S., SINGER, J., STOREY, M.-A. & DAMIAN, D. (2008) Selecting Empirical Methods for Software Engineering Research. IN SHULL, F., SINGER, J. & SJØBERG, D. I. K. (Eds.) *Guide to Advanced Empirical Software Engineering.* London, Springer.

EICHINGER, C., (2006) Web application architecture. IN KAPPEL, G. et al. (Eds.) *Web Engineering: The Discipline of Systematic Development of Web Applications*, John Wiley and Sons, New York, pp. 65–84.

ELGUEZABAL, L. B. M. A. G. B. (2005) Business Process Definition Languages Versus Traditional Methods Towards Interoperability. *4th International Conference COTS-Based Software Systems-ICCBSS 2005.* Bilbao, Spain, Springer-Verlag.

EMERSON, R. M., FRETZ, R. I. & SHAW, A. L. L. (1995) *Writing Ethnographic Fieldnotes,* Chicago, The University of Chicago Press.

EVERNDEN, R. & EVERNDEN, E. (2003) *Information First by Roger- Integrating Knowledge and Information Architecture for Business Advantage,* London, Elsevier.

FALESSI, D., CANTONE, G. & KRUCHTEN, P. (2008a) Value-Based Design Decision Rationale Documentation: Principles and Empirical Feasibility Study. *Proceedings of the Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008).* IEEE Computer Society.

FALESSI, D., CAPILLA, R. & CANTONE, G. (2008b) A value-based approach for documenting design decisions rationale: a replicated experiment. *Proceedings of the 3rd international workshop on Sharing and reusing architectural knowledge.* Leipzig, Germany, ACM.

FARENHORST, R., LAGO, P. & VLIET, H. V. (2007) Effective Tool Support for Architectural Knowledge Sharing IN OQUENDO, F. (Ed.) *European Conference Software Architecture-ECSA 2007.* Madrid, Spain, Springer-Verlag

FENG, T. & MALETIC, J. I. (2006) Applying Dynamic Change Impact Analysis in Component-based Architecture Design. *Proceedings of the Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing.* IEEE Computer Society.

FIELDING, R. T. & TAYLOR, R. N. (2000) Principled design of the modern Web architecture. *Proceedings of the 22nd international conference on Software engineering.* Limerick, Ireland, ACM.

FITZPATRICK, J. J. & KAZER, M. W. (2006) *Encyclopedia of Nursing Research,* New York, Springer Publishing Company.

FLYVBJERG, B. (2007) Five misunderstandings about case-study research. IN SEALE, C., GOBO, G., AND, J. F. G. & SILVERMAN, D. (Eds.) *Qualitative Research Practice: Concise Paperback Edition.* London and Thousand Oaks, Sage Publications Inc.

FOSTER, H., UCHITEL, S., MAGEE, J., KRAMER, J. & HU, M. (2005) Using a Rigorous Approach for Engineering Web Service Compositions: A Case Study. *Proceedings of the 2005 IEEE International Conference on Services Computing.* IEEE Computer Society.

FRANCISCO NAVA, R. C., JUAN C. DUEÑAS (2007) Processes for Creating and Exploiting Architectural Design Decisions with Tool Support. IN OQUENDO, F. (Ed.) *First European Conference on Software Architecture-ECSA 2007* Aranjuez, Spain, Springer.

GALLAGHER, K. B. & LYLE, J. R. (1991) Using Program Slicing in Software Maintenance *IEEE Transactions on Software Engineering,* 17**,** 751-761.

GALLAHER, M. P., O'CONNOR, A. C., DETTBARN, J. L. & GILDAY, L. T. (2004) Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry. Gaithersburg, Maryland, NIST.

GALLEN, C. (2010) Shopping by Mobile Will Grow to $119 Billion in 2015 New York, ABI Research

GALLIERS, J., SUTCLIFFE, A. & MINOCHA, S. (1999) An impact analysis method for safety-critical user interface design. ACM Trans. Comput.-Hum. Interact., 6, 341-369.

GARRIDO, A., ROSSI, G. & DISTANTE, D. (2009) Systematic improvement of Web Applications design *Journal of Web Engineering,* 8**,** 371-404.

GARRIDO, A., ROSSI, G. & SCHWABE, D. (1997) Pattern Systems for Hypermedia. *in Pattern Language of Programming- PloP'97.* Allerton Park Monticello , Illinois , USA IEEE Computer Society.

GARTNER (2010a) Gartner Highlights Key Predictions for IT Organizations and Users in 2010 and Beyond. Gartner Inc.

GARTNER (2010b) Gartner Says Worldwide IT Spending To Grow 4.6 Percent in 2010. Gartner Inc.

GILBERT, N. (1996) *Researching social life,* Thousand Oaks, California, Sage Publication Ltd.

GILLHAM, B. (2000) *Case study research methods,* London Continuum.

GINIGE, A. & MURUGESAN, S. (2001a) The Essence of Web Engineering: Managing the Diversity and Complexity of Web Application Development. *IEEE Multimedia,* 8**,** 22-25.

GINIGE, A. & MURUGESAN, S. (2001b) Web Engineering: An Introduction. *IEEE MultiMedia,* 8**,** 14-18.

GODFREY, M. W. & GERMAN, D. M. (2008) The Past, Present, and Future of Software Evolution. *Frontiers of Software Maintenace track at IEEE Intl. Conf. on Software Maintenance (ICSM-08).* Beijing, China, IEEE Computer Society.

GOMEZ, J., CACHERO, C. & PASTOR, O. (2001) Conceptual Modeling of Device-Independent Web Applications. *IEEE MultiMedia,* 8**,** 26-39.

GORDILLO, S., ROSSI, G., MOREIRA, A., ARAUJO, J., VAIRETTI, C. & URBIETA, M. (2006) Modelling and Composing Navigational Concerns in Web Applications. Requirements and Design Issues. *Fourth Latin American Web Congress* Cholula, Mexico. IEEE Computer Society.

GREENE, J. C., CARACELLI, V. J. & GRAHAM, W. F. (1989) Toward a Conceptual Framework for Mixed-Method Evaluation Designs. *Educational Evaluation and Policy Analysis,* 11**,** 255-274.

GRIFFIN "Griffin Porject", http://www.vu.nl/en/about-vu-amsterdam/mission-and-profile/the-griffin/index.asp.

GROUP, T. O. (2010) *World-Class Enterprise Architecture: Framework Guidance and TOGAF™ 9 Example*. San Francisco, CA, The Open Group Adoption Strategies Working Group.

HADAR, E. & HADAR, I. (2006) Meta-Architecture Tools: the Working Engines of the Company's Architectural Evolution Practioner Poistion. *In Workshop on Architecture-Centric Evolution (ACE 2006) in conjunction with 20th European Conference on Object-Oriented Programming ECOOP 2006,* Nantes, France.

HAMEL, J., DUFOUR, S. & FORTIN, D. (1993) *Case Study Methods (Qualitative Research Methods),* California, Sage Publications Inc.

HARRIS, K. L. (1996). Content analysis in negotiation research: A review and guide. *Behavior Research Methods Instruments and Computer,* 28, 3, 458-467.

HARRISON, N. B., AVGERIOU, P. & ZDUN, U. (2007) Using Patterns to Capture Architectural Decisions. *IEEE Softw.,* 24**,** 38-45.

HENVER, A. R., MARCH, S. T., PARK, J., & RAM, S. (2004). Design science research in information systems. *Management Information Systems Quarterly,* 28,1, 75-105.

HOFMEISTER, C., NORD, R. & SONI, D. (2000) *Applied Software Architecture,* Reading, Massachusetts, Addison Wesley Longman Inc.

HOLCK, J. (2003) 4 Perspectives on Web Information Systems. *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03)*. IEEE Computer Society.

HOST, M. & RUNESON, P. (2007) Checklists for Software Engineering Case Study Research *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007.* Madrid, Spain IEEE Computer Society.

HUANG, L. & SONG, Y.-T. (2006) Dynamic Impact Analysis Using Execution Profile Tracing. *Fourth International Conference on Software Engineering Research, Management and Applications (SERA'06).* Seattle, Washington, USA, IEEE Computer Society.

HULL M, E. C., TAYLOR, P. S., HANNA, J. R. P. & MILLAR, R. J., 2002. Software development processes - an assesment. *Information and Software Technology*, 44, 1-12.

IBRAHIM, S., IDRIS, N. B., MUNRO, M. & DERAMAN:, A. (2006) Software Traceability Validation For Change Impact Analysis of Object Oriented Software. *International Conference on Software Engineering Research and*

Practice & Conference on Programming Languages and Compilers. Las Vegas Nevada, USA, CSREA Press.

IEEE (2000) IEEE Recommended Practice for Architecture Description of Software-Intensive Systems. New York, IEEE Computer Society.

IIVARI, J. & LYYTINEN, K. (1998) Research on information systems development in Scandinavia—unity in plurality. *Scand. J. Inf. Syst.,* 10**,** 135-185.

INAGANTI, S. & BEHARA, G. K. (2007) Service Identification: BPM and SOA Handshake. *BPTrends (online Journal),* 1.

ISAKOWITZ, T., STOHR, E., & Balasubramanian, P. (1995) RMM: A Methodology for Structuring Hypermedia Design. *Communications of the ACM (CACM)*, 38, 8, 34-44.

JACYNTHO, M. D., SCHWABE, D. & ROSSI, G. (2002) A Software Atchitecture for Structuring Complex Web Applications. *Journal of Web Engineering,* 1**,** 37-60.

JANSEN, A., AVGERIOU, P. & VEN, J. S. V. D. (2009) Enriching software architecture documentation. *The Journal of Systems and Software,* 82**,** 1232-1248.

JANSEN, A. & BOSCH, J. (2005) Software Architecture as a Set of Architectural Design Decisions. *Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture.* IEEE Computer Society.

JANSEN, A., BOSCH, J. & AVGERIOU, P. (2008) Documenting after the fact: Recovering architectural design decisions. *J. Syst. Softw.,* 81**,** 536-557.

JANSEN, A., VEN, J. V. D., AVGERIOU, P. & HAMMER, D. K. (2007) Tool Support for Architectural Decisions. *Proceedings of the Sixth Working IEEE/IFIP Conference on Software Architecture.* IEEE Computer Society.

JANSSEN, M., KUK, G. & WAGENAAR, R. W. (2008) A survey of e-government business models in the Netherlands. *Government Information Quarterly,* 25**,** 202-220.

JOHNSON, R. B. & ONWUEGBUZIE, A. J. (2004) Mixed Methods Research: A Research Paradigm Whose Time Has Come. *Educational Researcher,* 33**,** 14-26.

JONSSON, P. (2007) Exploring Process Aspects of Change Imapct Analysis. *System and Software Engineering,.* PhD Theses, Blekings, Blekinge Institute of Technology.

JONSSON, P. & LINDVALL, M. (2005) Impact Analysis IN AURUM, A. & WOHLIN, C. (Eds.) *Engineering and Managing Software Requirements.* Berlin Heidelberge, Springer Berlin Heidelberg.

KAGDI, H. & MALETIC, J. I. (2006) Software-Change Prediction: Estimated+Actual. *Second International IEEE Workshop on Software Evolvability (SE'06).* Philadelphia, Pennsylvania, USA, IEEE Computer Society.

KAGDI, H., MALETIC, J. I. & SUTTON, A. (2005) Context-Free Slicing of UML Class Models. *Proceedings of the 21st IEEE International Conference on Software Maintenance.* IEEE Computer Society.

KALAKOTA, R., ROBINSON, M., (1999) *E-Business:roadmap for success*, Boston, MA, Addison-Wesley.

KALAKOTA, R., ROBINSON, M., (2003) *From e-Business to services: why and why now?* In:Services Blueprint: Roadmap for Execution, Boston, MA, Addison-Wseley.

KAZMAN, R., ABOWD, G., BASS, L. & CLEMENTS, P. (1996) Scenario-Based Analysis of Software Architecture. *IEEE Softw.,* 13**,** 47-55.

KAZMAN, R., BASS, L., WEBB, M. & ABOWD, G. (1994) SAAM: A Method for Analyzing the Properties of Software Architectures. *Proceedings of the 16th*

international conference on Software engineering. Sorrento, Italy, IEEE Computer Society Press.
KIM, T., SONG, Y.-T., CHUNG, L. & HUYNH, D. T. (2000) Software architecture analysis: A dynamic slicing approach. *Journal of Computer and Information Science,* 1**,** 91-103.
KITCHENHAM, B., PICKARD, L. & PFLEEGER, S. L. (1995) Case Studies for Method and Tool Evaluation. *IEEE Softw.,* 12**,** 52-62.
KITCHENHAM, B. A. (1996) Evaluating Software Engineering Methods and Tool Part 2: Selecting an appropriate evaluation method. *ACM SIGSOFT Software Engineering Notes,* ACM Publisher.
KLEIN, H. K. & MYERS, M. D. (1999) A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Q.,* 23**,** 67-93.
KNAPP, A., KOCH, N., ZHANG, G. & HASSLER, H.-M. (2004) Modeling Business Processes in Web Applications with ArgoUWE IN BAAR, T., STROHMEIER, A., MOREIRA, A. & MELLOR, S. J. (Eds.) *7th International Conference Modelling Languages and Applications.* Lisbon, Portugal, Springer Berlin Heidelberg.
KOCH, N., KRAUS, A., CACHERO, C. & MELIA, S. (2004) Integration of Business Processes in Web Application Model. *Journal of Web Engineering,* 3**,** 22-49.
KONG, X., LIU, L. & LOWE, D. (2005a) Separation of Concerns: a Web Application Architecture Framework. *Journal of Digital Information,* 6.
KONG, X., LIU, L. & LOWE, D. (2005b) Supporting Web user interface prototyping through information modeling and system architecting. *IEEE International Conference on e-Business Engineering-ICEBE'05.* Beijing, China, IEEE Computer Society.
KONG, X., LIU, L. & LOWE, D. (2005c) Web System Trace Model Using a Web Application Architecture Framework. *Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05) on e-Technology, e-Commerce and e-Service.* IEEE Computer Society.
KOREL, B., SINGH, I., TAHAT, L. & VAYSBURG, B. (2003) Slicing of state-based models. *International Conference on Software Maintenance - ICSM '03* Amsterdam, The Netherlands, IEEE Computer Society.
KOTONYA, G. & SOMMERVILLE, I. (1998) *Requirements Engineering: Processes and Techniques,* Boston, John Wiley & Sons, Inc.
KRIPPENDORFF, K. (2004) *Content analysis: An introduction to its methodology,* Thousand Oaks, California, Sage Publications, Inc. .
KRUCHTEN, P. (1995) Architectural Blueprints - the 4+1 View Model of Software Architecture. *IEEE Software,* 12**,** 42-60.
KRUCHTEN, P. (2004) An Ontology of Architectural Design Decisions. *2nd Groningen Workshop on Software Variability Management.* Groningen, NL.
KRUCHTEN, P., LAGO, P. & VLIET, H. V. (2005) Building up and reasoning about architectural knowledge. *QoSA-Quality of Software Architecture.* Vasteras, Sweden, Springer-Verlag.
KRUCHTEN, P., LAGO, P. & VLIET, H. V. (2006) Building Up and Reasoning About Architectural Knowledge. IN HOFMEISTER, C., CRNKOVIC, I. & REUSSNER, R. (Eds.) *Quality of Software Architectures.* Berlin, Springer Berlin / Heidelberg.
KRUCHTEN, P., LAGO, P., VLIET, H. V. & WOLF, T. (2005b) Building up and Exploiting Architectural Knowledge. *Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture.* IEEE Computer Society.

KVALE, S. (1996) *Interviews: An Introduction to Qualitative Research Interviewing,* London, Sage Publication Inc.

LALLCHANDANI, J. T. & MALL, R. (2008) Slicing UML architectural models. *SIGSOFT Softw. Eng. Notes,* 33**,** 1-9.

LANA S. AL-SALEMA & SAMAHAB, A. A. (2007) Eliciting Web application requirements – an industrial case study, *Journal of Systems and Software,* 80**,** 294-313

LANGE, C. F. J. & CHAUDRON, M. R. V. (2004) An Empirical Assessment of Completeness in UML Designs. *Proceedings of EASE – International Conference on Empirical Assessment in Software Engineering*, Institution of Electrical Engineers (IEE) Press, Edinburgh, Scotland.

LANGE, C. F. J. & CHAUDRON, M. R. V. (2005) Managing Model Quality in UML-Based Software Development. *Proceedings of the 13th IEEE International Workshop on Software Technology and Engineering Practice-STEP '05,* Budapest, Hungary, IEEE Computer Society Washington, DC, USA

LASSING, N., BENGTSSON, P., VLIET, H. V. & BOSCH, J. (2002) Experiences with ALMA: architecture-level modifiability analysis. *J. Syst. Softw.,* 61**,** 47-57.

LASSING, N., RIJSENBRIJ, D. & VLIET, H. V. (1999a) Towards a Broader View on Software Architecture Analysis of Flexibility. *Proceedings of the Sixth Asia Pacific Software Engineering Conference.* IEEE Computer Society.

LASSING, N. H., RIJSENBRIJ, D. B. B. & VLIET, J. C. V. (1999b) Flexibility of the ComBAD Architectures. *Proceedings of the TC2 First Working IFIP Conference on Software Architecture (WICSA1).* Kluwer, B.V.

LAUDON, K. C. & TRAVER, C. G. (2007) *E-Commerce: Business, technology, society,* 3rd Edition, New York Pearson Prentice Hall.

LAUDON, K. C. & TRAVER, C. G. (2010) *E-Commerce: Business, technology, society,* 6th Edition, New York Pearson Prentice Hall.

LAW, J. & ROTHERMEL, G. (2003) Whole program Path-Based dynamic impact analysis *Proceedings of the 25th International Conference on Software Engineering* Portland, Oregeon USA, IEEE Computer Society.

LEE, A. S. (1989) A scientific methodology for MIS case studies. *MIS Q.,* 13**,** 33-50.

LEE, L. & KRUCHTEN, P. (2008a) A Tool to Visualize Architectural Design Decisions. *Proceedings of the 4th International Conference on Quality of Software-Architectures: Models and Architectures.* Karlsruhe, Germany, Springer-Verlag.

LEE, L. & KRUCHTEN, P. (2008b) Visualizing Software Architectural Design Decisions IN MORRISON, R., BALASUBRAMANIAM, D. & FALKNER, K. (Eds.) *Second European Conference on Software Architecture-ECSA 2008.* Paphos, Cyprus, Springer.

LEE, M., OUTT, A. J. & ALEXANDER, R. T. (2000) Algorithmic Analysis of the Impacts of Changes to Object-oriented Software. *International Conference on Technology of Object-Oriented Languages and Systems (TOOLS USA '00).* Santa Barbara, CA, USA, ACM Press.

LEEDY, P. D. & ORMROD, J. E. (2004) *Practical Research: Planning and Design* New Jersey, Prentice Hall.

LEHMAN, M. M. & RAMIL, J. F. (2001) An approach to a theory of software evolution. *Proceedings of the 4th International Workshop on Principles of Software Evolution.* Vienna, Austria, ACM.

LEHMAN, M. M. & RAMIL, J. F. (2002) Software Evolution and Software Evolution Processes. *Ann. Softw. Eng.,* 14**,** 275-309.

LEVY, P. S. & LEMESHOW, S. (2008) *Sampling of Populations: Methods and Applications,* New York, John Wiley and Sons, Inc.

LINCOLN, Y. & GUBA, E. G. (1985) *Naturalistic Inquiry,* Newbury Park, California, Sage Publications, Inc.

LINDVALL, M. (1997) An Emprical Study of Requirements-Driven Impact Analysis in Object-Oriented Software Evolution. *Department of Computer and Information Sciences.* Linkoping, University of Linkoping.

LIU, Y., ZHANG, S. R. & FANG, M.Q. (2007) Ecological Analysis on Evolution of Information Systems. Integration and Innovation Orient to E-Society, 2, 308-315.

LOSHIN, D. (2001) *Enterprise knowledge management: the data quality approach* San Francisco, California, Morgan Kaufmann Publisher.

LOSHIN, D. (2003) The Business Value of Information Flow Modeling Knowledge Integrity *Information Management Magazine,* 1.

LOWE, D. (2000) A Framework for Defining Acceptance Criteria for Web Development Projects IN G. GOOS, HARTMANIS, J. & LEEUWEN, J. V. (Eds.) *Web Engineering.* Berlin, Springer Berlin / Heidelberg.

LOWE, D. (2003) Web System Requirements. *Requirement Eng. Journal*, 10.

LOWE, D. & EKLUND, J. (2001) Development issues in specification of Web systems. *AWRE'2001: 6th Australian Workshop on Requirements Engineering.* Sydney, Australia.

LOWE, D. & EKLUND, J. (2002) Client Needs and the Design Process in Web Projects. *Journal of Web Engineering,* 1**,** 23-36.

LOWE, D. & HENDERSON-SELLERS, B. (2001a) Impacts on the development process of differences between web systems and conventional software systems. *International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet-SSGRR 2001.* L'Aquila, Italy.

LOWE, D. & HENDERSON-SELLERS, B. (2001b) Web Development: Addressing Process Differences. *Cutter IT,* 14**,** 11-17.

LOWE, D. & HENDERSON-SELLERS, B. (2003) Characterising Web Systems: Merging Information and Functional Architectures. IN SHI, N. & MURTHY, V. K. (Eds.) *Architecture Issues of Web-Enabled Electronic Business* Hershey, PA, USA, IDEA Group Publishing.

MACLEAN, A., YOUNG, R. M., BELLOTTI, V. M. E. & MORAN, T. P. (1991) Questions, options, and criteria: elements of design space analysis. *Hum.-Comput. Interact.,* 6**,** 201-250.

MOCKUS, A. & Votta, L. G., (2000) Identifying Reasons for Software Changes using Historic Databases. *Proceedings of the 2000 International Conference on Software Maintenance,* IEEE Computer Society, pp 120-130.

MANOLESCU, D., MANOLESCU, D. A. & JOHNSON, R. (2000) A Micro-Workflow Component for Federated Workflow *OOPSLA2000 Workshop on Implementation and Application of Object-Oriented Workflow Management Systems.*

MARK, S., MIKE, L. & LAURIE, W. (2007) Prioritization of Regression Tests using Singular Value Decomposition with Empirical Change Records. *18th IEEE International Symposium on Software Reliability, 2007 (ISSRE '07).* Trollhättan, Sweden, IEEE Computer Society.

MATTSSON, A., LUNDELL, B. & LINGS, B. (2008) Modelling architectural design rules in UML.

MCDONALD, A. & WELLAND, R. (2001) Web Engineering in Practice. *Fourth WWW10 Workshop on Web Engineering.* Hong Kong, ACM Press.

MCCRICKARD, D. S. & ABOWD, G. D. (1996) Assessing the impact of changes at the architectural level: a case study on graphical debuggers. *Proceedings of the 1996 International Conference on Software Maintenance.* IEEE Computer Society.

MEDIA, M. R. (2010) U.S. Federal IT Market Forecast 2011-2015. Market Research media.

MEHBOOB, Z. (2007) Change Impact Analysis (CIA) in Web systems. *In 9th annual cross disciplinary Postgraduate Student Conference,* UTS, Sydney, Australia.

MEHBOOB, Z., ZOWGHI, D. & LOWE, D (2008) Supporting Integrated Dependency Model for Change Impact Analysis in Web Systems. *Proceeding of Fourteenth Australasian World Wide Web Conference-AusWeb08,* Ballina, Australia.

MEHBOOB, Z. & ZOWGHI, D. (2009) Industrial Perspectives on Architecture level Change Impact Analysis in Web Systems Evolution. *Proceeding of 11th IEEE International Symposium on Web Systems Evolution, in conjunction with the 25th IEEE International Conference on Software Maintenance.* Edmonton, Canada, IEEE computer Society.

MEHBOOB, Z., ZOWGHI, D. & LOWE, D. (2009) An Approach for Comparison of Architecture Level Change Impact Analysis Methods and Their Relevance in Web Systems Evolution. *Proceedings of 20th Australian Software Engineering Conference-ASWEC 2009.* Gold Coast, Queensland, Australia, IEEE Computer Society.

MEHBOOB, Z., ZOWGHI, D. & LOWE, D. (2011) A Process Model of Change Impact Analysis for Web Systems Using Design Knowledge. *Proceedings of 8th European, Mediterranean & Middle Eastern Conference on Information Systems-EMCIS2011.* Athens, Greece.

MEHBOOB, Z., KHAN, A. M., (2013) Approach for Change Impact Analysis in Web systems - An Industrial Case study. *International Journal of Software Engineering and Its Applications (IJSEIA)*, 7, 6.

MEHBOOB, Z. (2013) Adopting A Process Model of Change Impact Analysis for Web systems - A Comparative Study. *International Journal of Information Technology (IJSIT),* 28, 3.

MEHTA, A. & HEINEMAN, G. T. (1999) Architectural Evolution of Legacy Systems. *23rd International Computer Software and Applications Conference,* Phoenix, AZ, USA. IEEE Computer Society.

MELTZOFF, J. (1998) *Critical thinking about research: Psychology and related fields,* Washington, DC American Psychological Association

MENAND, L. (1997) *Pragmatism: A Reader,* New York, Vintage.

MENDES, E., MOSLEY, N. & COUNSELL, S. (2006) The Need for Web Engineering: An Introduction. IN MENDES, E. & MOSLEY, N. (Eds.) *Web Engineering.* Berlin, Springer-Verlag Berlin Heidelberg.

MERRIAM-WEBSTER (2003) Merriam-Webster Incorporated. http://www.m-w.com/cgi-bin/dictionary.

MICHAEL, A. C. (2004) Who is liable for bugs and security flaws in software? *Commun. ACM,* 47**,** 25-27.

MIKE J. VAN WIE, RAJAN., P. & CAMPBELL, M. I. (2003) Representing Product Architecture. *Proceedings of ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference-DETC'03.* Chicago, Illinois, USA.

MILES, M. B. & HUBERMAN, A. M. (1994) *Qualitative Data Analysis: An Expanded Sourcebook* Thousand Oaks, California, Sage Publications, Inc.

MITTERMEIR, R. T. (2001) Software evolution: let's sharpen the terminology before sharpening (out-of-scope) tools. *Proceedings of the 4th International Workshop on Principles of Software Evolution.* Vienna, Austria, ACM.

MORVILLE, P. & ROSENFELD, L. (2006) *Information Architecture for the World Wide Web,* Sebastopol, CA, O'Reilly Media.

MURUGESAN, S. (2008) Web Application Development: Challenges And The Role Of Web Engineering. IN ROSSI, G., PASTOR, O., SCHWABE, D. & OLSINA, L. (Eds.) *Web Engineering: Modelling and Implementing Web Applications.* London, Springer.

MURUGESAN, S. & DESHPANDE, Y. (2001) Web Engineering, Software Engineering and Web Application Development. *Lecture Notes In Computer Science.*

MURUGESAN, S. & DESHPANDE, Y. (2005) Web Engineering: Introduction and Perspectives IN MURUGESAN, S. & DESHPANDE, Y. (Eds.) *Web Engineering: Principles and Techniques.* Idea Group Publishing.

MURUGESAN, S., DESHPANDE, Y., HANSEN, S. & GINIGE, A. (2001) Web Engineering: a New Discipline for Development of Web-Based Systems IN MURUGESAN, S. & DESHPANDE, Y. (Eds.) *Web Engineering: Managing Diversity and Complexity of Web Application Development* Berlin, Springer-Verlag.

NACHMIAS, D. & NACHMIAS, C. F. (2007) *Research Methods in the Social Sciences,* United Kindom Worth Publishers.

NEDSTAM, J., KARLSSON, E.-A. & HOST, M. (2004) The Architectural Change Process. *Proceedings of the 2004 International Symposium on Empirical Software Engineering.* IEEE Computer Society.

NEUMAN, W. L. (2000) *Social Research Methods: Qualitative and Quantitative Approaches,* Boston, , Allyn and Bacon.

NORTON, K. S. (2001) Applying Cross-Functional Evolutionary Methodologies to Web Development. *Web Engineering, Software Engineering and Web Application Development,* Springer-Verlag.

NUNAMAKER, J. F., CHEN, M. & PURDIN, T. D. M. (1990) Systems development in information systems research. *J. Manage. Inf. Syst.,* 7**,** 89-106.

NURMULIANI, N. (2007) Investigating Requirements Volatility during Software Development Projects: An Empirical Study. PhD Theses, *Faculty of Information Technology,* Sydney, University of Technology Sydney.

NUSEIBEH, B. (2001) Weaving Together Requirements and Architectures. *Computer,* 34**,** 115-117.

OATES, B. J. (2006) *Researching Information Systems and Computing,* Los Angeles, CA, Sage Publication.

O'NEAL, J. S. (2001) Analyzing the Impact of Changing Requirements. *Proceedings of the IEEE International Conference on Software Maintenance (ICSM'01).* Trento, Italy, IEEE Computer Society.

ORLIKOWSKI, W. J. & BAROUDI, J. J. (1991) Studying Information Technology in Organizations: Research Approaches and Assumptions. *Information Systems Research,* 2**,** 1-28.

ORSO, A., APIWATTANAPONG, T. & HARROLD, M. J. (2003) Leveraging Field Data for Impact Analysis and Regression Testing. *Proceedings of the 9th European software engineering conference held jointly with 11th ACM*

SIGSOFT international symposium on Foundations of software engineering Helsinki, Finland ACM Publisher.

ORSO, A., APIWATTANAPONG, T., LAW, J., ROTHERMEL, G. & HARROLD, M. J. (2004) An Emprical Comparison of Dynamic Imapct Analysis Algorithms. *Proceedings of the 26th International Conference on Software Engineering,* Scotland, UK, IEEE Computer Society.

PAIANO, R., GUIDO, A. L. & PANDURINO, A. (2009a) *Designing Complex Web Information Systems: Integrating Evolutionary Process Engineering,* Hershey, New York, IGI Global.

PAIANO, R., GUIDO, A. L. & PANDURINO, A. (2009b) Web Information System Design Methodologies Overview *Designing Complex Web Information Systems: Integrating Evolutionary Process Engineering.* IGI Global Publishing.

PALMER, J. D. (1997) Traceability. IN THAYER, R. H. & DORFMAN, M. (Eds.) *Software Requirements Engineering.* 2nd Edition, Los Alamitos, CA, Wiley-IEEE Computer Society Press.

PAN, Y., PAN, D., & CHEN, M.-H. (2005) Slicing Component-Based Systems. *Proceeding of 10th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'05).* IEEE Computer Society.

PASTOR, O., SCHWABE, D., ROSSI, G. & OLSINA, L. (2002) Web-Oriented Software Technology. *Proceedings of the Workshops and Posters on Object-Oriented Technology.* Springer-Verlag.

PATTON, M. Q. (2002) *Qualitative Research & Evaluation Methods,* Thousand Oaks, CA, Sage Publication.

PERRY, D. E., SIM, S. E. & EASTERBROOK, S. M. (2004) Case Studies for Software Engineers. *Proceedings of the 26th International Conference on Software Engineering.* Scotland, UK, IEEE Computer Society.

PERRY, D. E. & WOLF, A. L. (1992) Foundations for the study of software architecture. *SIGSOFT Softw. Eng. Notes,* 17**,** 40-52.

PERVAN, G. & MAIMBO, H. (2005) Designing a case study protocol for application in IS research. *The Ninth Pacific Conference on Information Systems.* Bangkok, Thailand.

PFLEEGER, S. L. (1998) *Software Engineering: Theory and practices*, New Jersey, Prentice Hall.

PILONE, D. & PITMAN, N. (2005) *UML 2.0 in a Nutshell,* Sebastopol, CA, O'Reilly Media.

POHL, K., BRANDENBURG, M. & GÜLICH, A. (2001) Integrating Requirements and Architecture Information: A Scenario and Meta-Model based Approach *Seventh International Workshop on Requirements: Foundation for Software Quality (REFSQ'01).* Interlaken, Switzerland.

POHL, K. & SIKORA, E. (2007) Structuring the Co-design of Requirements and Architecture IN SAWYER, P., PAECH, B. & HEYMANS, P. (Eds.) *13th Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'07).* Trondheim, Norway, Springer Berlin / Heidelberg.

PRESSMAN, R. S. (2001) Web Engineering: An Adult's Guide to Developing Internet-Based Applications. *Cutter IT Journal,* 14**,** 2-5.

PRESSMAN, R. S. (2005) *Software Engineering: A Practitioner's Approach,* New York, NY, McGraw-Hill.

PRESSMAN, R. S. & LOWE, D. (2008) *Web Engineering- A practitioner's Approach,* New York, McGraw Hill.

QUINTERO, A. M. R. (2009) Surveying navigation modelling approaches. *Int. J. Comput. Appl. Technol.,* 33**,** 327-336.

RAFLA, T., ROBILLARD, P. N. & DESMARAIS, M. (2006) Investigating the impact of usability on software architecture through scenarios: A case study on Web systems. *J. Syst. Softw.,* 79**,** 415-426.

RAMESH, B. & JARKE, M. (2001) Toward Reference Models for Requirements Traceability *IEEE Transaction Software Engineering,* 27**,** 58-93

RAMESH, B., PRIES-HEJE, J. & BASKERVILLE, R. (2002) Internet Software Engineering: A Different Class of Processes. *Ann. Softw. Eng.,* 14**,** 169-195.

RAMLER, R., WOLFMAIER, K. & WEIPPL, E. (2004) From Maintenance to Evolutionary Development of Web Applications: A Pragmatic Approach IN KOCH, N., FRATERNALI, P. & MARTINWIRSING (Eds.) *4th International Conference on Web Engineering-CWE 2004.* Munich, Germany, Springer Berlin Heidelberg.

RANGANATHAN, C. & GANAPATHY, S. (2002) Key dimensions of business-to-consumer web sites. *Inf. Manage.,* 39**,** 457-465.

REINA-QUINTERO, A. M. (2008) Surveying navigation modelling approaches. *International Journal of Computer Applications in Technology,* 33**,** 327 - 336

REINGRUBER, M. C. & GREGORY, W. W. (1994). *The Data Modeling Handbook*. Wiley-QED, New York.

RICCA, F. (2004) Analysis, Testing and Re-Structuring of Web Applications. *Proceedings of the 20th IEEE International Conference on Software Maintenance.* Chicago, USA, IEEE Computer Society.

RICCA, F. & TONELLA, P. (2000) Web Site Analysis: Structure and Evolution. *Proceedings of the International Conference on Software Maintenance (ICSM'00),* California, USA, IEEE Computer Society.

RICCA, F. & TONELLA, P. (2001) Analysis and testing of Web applications. *Proceedings of the 23rd International Conference on Software Engineering.* Toronto, Ontario, Canada, IEEE Computer Society.

RICHARD, V., DAVID, A., BOB, W. & TREVOR, W.-H. (2002) *Developing web information systems,* Oxford, UK, Butterworth-Heinemann.

ROBBINS, J. E., HILBERT, D. M. & REDMILES, D. F. (1996) Using critics to analyze evolving architectures. *Joint proceedings of the second international software architecture workshop (ISAW-2) and international workshop on multiple perspectives in software development (Viewpoints '96) on SIGSOFT '96 workshops.* San Francisco, California, United States, ACM.

ROBSON, C. (2002) *Real world research: a resource for social scientists and practioner-researchers,* Oxford, Blackwell Publishers Ltd.

ROMBACH, H. D. (1990) Design Measurement: Some Lessons Learned. *IEEE Softw.,* 7**,** 17-25.

ROSEN, K. H. (2006) *Discrete Mathematics and Its Applications,* LA,USA, McGraw-Hill.

ROSSI, G. H., SCHMID, H. A. & LYARDET, F. (2003) Customizing Business Processes in Web Applications. IN BAUKNECHT, K., TJOA, A. M. & QUIRCHMAYR, G. (Eds.) *4th International Conference E-Commerce and Web Technologies,EC-Web '03.* Prauge, Czech Republic, Springer-Verlag.

RUNESON, P. & HOST, M. (2009) Guidelines for conducting and reporting case study research in software engineering. *Empir Software Eng,* 14**,** 131-164.

RYAN, G. W. & BERNARD, H. R. (2003) Techniques to Identify Themes. *Field Methods,* 15**,** 85-109.

RYDER, B. G. & TIP, F. (2001) Change Impact Analysis for Object Oriented Programs. *Workshop on Program Analysis for Software Tools and Engineering (PASTE'01),* Snowbird, Utah, USA, ACM Press.

SADOU, N., TAMZALIT, D. & OUSSALAH, M. (2005) A unified Approach for Software Architecture Evolution at different abstraction levels. *Proceedings of the Eighth International Workshop on Principles of Software Evolution.* Lisbon, Portugal, IEEE Computer Society.

SANGAL, N. (2007) Lightweight Dependency Models to Manage Software Architecture. *Proceedings of the Sixth Working IEEE/IFIP Conference on Software Architecture*. Mumbai, India, IEEE Computer Society.

SAVAGE, J. E. (1999) *Models Of Computation- Exploring the Power of Computing,* Menlo Park, California, Addison Wesley Publishing Company.

SCHARL, A. (2000) *Evolutionary Web Development,* London, Springer-Verlag.

SCHWABE, D., ROSSI, G. & BARBOSA, S. (1996) Systematic Hypermedia Design with OOHDM, *In 7th ACM International Conference on Hypertest'96*, Washington, USA.

SCHWABE, D. & ROSSI, G. (1998) Developing Hypermedia Applications using OOHDM, *In 9th ACM International Conference on Hypertest'98*, Pittsburgh, USA.

SCHWABE, D., ESMERALDO, L., ROSSI, G. & LYARDET, F. (2001) Engineering Web Applications for Reuse. *IEEE MultiMedia,* 8**,** 20-31.

SCHWABE, D., ROSSI, G., ESMERALDO, L. & LYARDET, F. (2000) Web Design Frameworks: An Approach to Improve Reuse in Web Applications. IN MURUGESAN, S. & DESHPANDE, Y. (Eds.) *Web Engineering: Managing Diversity and Complexity of Web Application Development* Berlin, Springer Berlin / Heidelberg.

SEACORD, R. C., PLAKOSH, D. & LEWIS, G. A. (2003) *Modernizing Legacy Systems: Software Technologies, Engineering Process and Business Practices*, Addison-Wesley Longman Publishing Co., Inc.

SHAW, D. G. A. M. (1993) An Introduction to Software Architecture. *Advances in Software Engineering and Knowledge Engineering,* 1**,** 1-42.

SHERRIFF, M., LAKE, M. & WILLIAMS, L. (2007) Empirical Software Change Impact Analysis using Singular Value Decomposition, IBM, North Carolina State University, Raleigh.

SHINA, G. (1999) Build a component architecture for ecommerce, *e-Business Advisor*. Advisor Publications, Inc.

SIMSION, G. & WITT, G. (2005) *Data Modeling Essentials,* San Franscisco, USA, Morgan Kaufmann Publishers.

SINGER, J. & VINSON, N. G. (2002) Ethical Issues in Empirical Studies of Software Engineering. *IEEE Trans. Softw. Eng.,* 28**,** 1171-1180.

SNEED, H. M. (2001) Impact Analysis of Maintenance Tasks for a Distributed Object-oriented System. *Proceedings of the IEEE International Conference on Software Maintenance (ICSM'01).* Florence, Italy, IEEE Computer Society.

SOMMERVILLE, I. (2007) *Software engineering* New York, Pearson Education Limited.

STAFFORD, J. A., WOLF, A. L. & CAPORUSCIO, M. (1998) Aladdin: A tool for architecture-level dependence analysis of software systems. University of Colorado, Dept. of Computer Science. CO, USA

STAFFORD, J. A., WOLF, A. L. & CAPORUSCIO, M. (2003) The Application of Dependence Analysis to Software Architecture Descriptions. IN BERNARDO,

M. & INVERARDI, P. (Eds.) *Lecture Notes in Computer Science.* Berlin, Springer-Verlag

STAKE, R. E. (1995) *The art of case research,* Thousand Oaks, CA, Sage Publications, Inc.

STATS, I. W. (2009) World Internet Usage Statistics News and World Population Stats. Internet World Stats.

STEIN, L. D. (2000) Profit, the prime directive. *WebTechniques,* 5**,** 14–17.

STOCKDALE, R. & STANDING, C. (2002) A framework for the selection of electronic marketplaces: a content analysis approach. *Internet Research,* 12**,** 221-234.

STRAUSS, A. & CORBIN, J. M. (1998) *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory,* Newbury Park, CA, Sage Publications.

SYMON, G. & CASSELL, C. (1998) *Qualitative Methods and Analysis in Organizational Research: A Practical Guide,* Thousand Oaks, CA, Sage Publications.

TANG, A., JIN, Y., HAN, J. & NICHOLSON, A. (2005) Predicting Change Impact in Architecture Design with Bayesian Belief Networks. *Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture.* IEEE Computer Society.

TANG, A., NICHOLSON, A., JIN, Y. & HAN, J. (2007) Using Bayesian belief networks for change impact analysis in architecture design. *J. Syst. Softw.,* 80**,** 127-148.

TAYLOR, R.N., Hoek, A.V.D., (2007) Software Design and Architecture The once and future focus of software engineering, *Proceeding of Future of Software Engineering*, IEEE Computer Society, 226-243.

TONELLA, P. & RICCA, F. (2002) Dynamic Model Extraction and Statistical Analysis of Web Applications *Proceedings of the Fourth International Workshop on Web Site Evolution (WSE'02).* Montréal, Canada, IEEE Computer Society.

TONGRUNGROJANA, R. & LOWE, D. (2005) Web Information Exchange Diagrams for UML. IN LOWE, D. & GAEDKE, M. (Eds.) *5th International Conference on Web Engineering-ICWE,* Sydney, Australia, Springer.

TONGRUNGROJANA, R. & LOWE, D. (2004a) Understanding business impacts of changes to information designs: A case study. *The Tenth Australian World Wide Web Conference.* Gold Coast, Australia.

TONGRUNGROJANA, R. & LOWE, D. (2004b) WIED: A Web Modelling Language for Modelling Architectural-Level Information Flows. *Journal of Digital Information,* 5.

TORRES, V. & PELECHANO, V. (2006) Building Business Process Driven Web Applications. IN DUSTDAR, S., FIADEIRO, J. L. & SHETH, A. (Eds.) *4th International Conference on Business Process Modelling-BPM 2006.* Vienna, Austria, Springer Berlin / Heidelberg.

TRUJILLO, S., AZANZA, M., DIAZ, O. & CAPILLA, R. (2007) Exploring Extensibility of Architectural Design Decisions. *Proceedings of the Second Workshop on SHAring and Reusing architectural Knowledge Architecture, Rationale, and Design Intent.* Minneapolis, MN, USA, IEEE Computer Society.

TURBAN, E., LEE, J. K. & CHUNG, M. (2000) *Electronic Commerce: A managerial perspective,* New York, Prentice-Hall.

TURBAN, E., LEE, J. K., KING, D., LIANG, T. P. & TURBAN, D. (2010) *Electronic Commerce,* Upper Saddle River, New Jersey, Prentice Prentice Hall.

TURVER, R.J. & MUNRO, M. (1994) An Early impact analysis technique for software maintenance, *Journal of Software Maintenance: Research and Practice*, 6, 1, 35-52.

TYREE, J. & AKERMAN, A. (2005) Architecture Decisions: Demystifying Architecture. *IEEE Softw.,* 22**,** 19-27.

VLAANDEREN, K., VALVERDE, F. & PASTOR, O. (2009) Model-Driven Web Engineering in the CMS Domain: A Preliminary Research Applying SME IN CORDEIRO, J. F. A. J. (Ed.) *Enterprise Information Systems.* Berlin, Springer Berlin Heidelberg.

WALSHAM G (1993) Interpreting Information Systems in Organizations. Wiley, Chichester.

WALTER, Z. & SCOTT, G. (2006) Management issues of internet/web systems. *Commun. ACM,* 49**,** 87-91.

WAN-KADIR, W. M. N. & LOUCOPOULOS, P. (2004) Relating evolving business rules to software design. *Journal of Systems Architecture,* 50**,** 367-382.

WANG, Z., SHERDIL, K. & MADHAVJI, N. H. (2005) ACCA: An Architecture-Centric Concern Analysis Method. *Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture.* IEEE Computer Society.

WHITE, S. A. (2004) Introduction to BPMN, New York, IBM Corporation.

WIERINGA, R. J. & HEERKENS, J. M. G. (2006) The methodological soundness of requirements engineering papers: a conceptual framework and two case studies. *Requir. Eng.,* 11**,** 295-307.

WILLIAMS, B. J. & CARVER, J. C. (2010) Characterizing software architecture changes: A systematic review. *Information and Software Technology,* 52**,** 31–51.

WRIGHT, J. & DIETRICH, J. (2008) Survey of existing languages to model interactive web applications. *Proceedings of the fifth on Asia-Pacific conference on conceptual modelling,* Wollongong, NSW, Australia, Australian Computer Society, Inc.

WOHLIN, C., HOST, M., OHLSSON, M.C., REGNELL, B., RUNESON, P. & WESSLEN, A., (2000) *Experimentation in software engineering—an introduction,* Norwell, MA, USA, Kluwer Academic Publishers.

WOHLIN, C., RUNESON, P., HOST, M., OHLSSON, M. C., REGNELL, B., & WESSLEN, A., (2012) *Experimentation in Software Engineering*, Springer, New York.

XIAOYU, Z. (2006) Modeling Multi-tier web Applications for testing and maintenance. Ph.D. thesis, *State University of New York,* Albany, New York, USA.

XIAOYU, Z. & MEI-HWA, C. (2007) Maintaining Multi-Tier Web Applications. *IEEE International Conference on Software Maintenance- ICSM 2007.* Paris, France, IEEE Computer Society.

YIN, R. K. (2002) *Case Study Research: Design and Methods,* California, Sage Publications Inc.

YIN, R. K. (2003) *Case Study Research Design and Methods,* California, Saga Publications Inc.

YUSOP, N. (2009) Issue Resolution and Scope Clarification in Web Systems Development: A Qualitative Study, Ph.D. thesis, *Faculty of Information Technology.* Sydney, University of Technology Sydney.

YUSOP, N., LOWE, D. & ZOWGHI, D. (2005) Impact of Web System on their domain. *Web Engineering,* 4**,** 313-338.

YUSOP, N., LOWE, D. & ZOWGHI, D. (2006) A Survey of Issue Resolution on the Incremental Refinement of the System Scope in Web System Development.

International Conference on Web Engineering-ICWE'06. Palo Alto, California, IEEE Computer Society.

ZHAO, J. (1998) Applying Slicing Technique to Software Architectures. *Fourth IEEE International Conference on Engineering Complex Computer Systems (ICECCS'98),* California, USA IEEE Computer Society.

ZHAO, J. (2001) Using Dependence Analysis to Support Software Architecture Understanding. *New Technologies on Computer Software,* 1**,** 135-142.

ZHAO, J. & XIANG, L. (2005) Architectural Slicing to Support System Evolution. IN KHAN, K. M. & ZHANG, Y. (Eds.) *Managing Corporate Information Systems Evolution and Maintenance.* Hershey, PA, USA, Idea Group Publishing.

ZHAO, J., YANG, H., XIANG, L. & XU, B. (2002) Change impact analysis to support architectural evolution, *Journal of Software Maintenance* 14**,** 317-333

ZIMMERMANN, T., WEISSGERBER, P., DIEHL, S. & ZELLER, A. (2005) Mining Version Histories to Guide Software Changes. *IEEE Trans. Softw. Eng. ,* 31 429-445.