Industrial Perspectives on Architecture Level Change Impact Analysis in Web Systems Evolution

Zafar Mehboob
University of Technology,
Sydney, Australia.
zafar@it.uts.edu.au

Didar Zowghi
University of Technology,
Sydney, Australia.
didar@it.uts.edu.au

Abstract

Change impact analysis (CIA) approaches are mostly developed based on the understanding of both system development process and system architecture. It is thus reasonable to assume that if the development process or system architecture is fundamentally different, then CIA approaches must address these differences accordingly. This is particularly relevant in the context of Web systems, given that these systems often have a specific set of characteristics related to differences both in the architecture and the process through which they are developed. To investigate this issue we have interviewed web designers/architects and reported the industrial perspectives on using current architecture CIA approaches. The interview findings reveal a set of elicited needs that an architecture CIA approach should address to be suitably adopted during Web systems evolution. Additionally, systematic classification of these needs indicates the focus areas and their relevance in Web systems. Finally, the study findings provide guidance for the possible extension of architecture CIA approaches specifically for Web systems.

1. Introduction

CIA is a crucial part of system maintenance and evolution, as systems are generally exposed to changing requirements. Bohner and Arnold [1] define CIA as '*identifying the potential consequences of a change, or estimating what needs to be modified to accomplish a change*'. Much of the research about CIA is focused on system code level, although CIA undoubtedly plays an important role in the entire system life cycle such as architecture design [2-4].

The development and maintenance of complex and evolvable systems has led to a realization that architecture can play an important role in successful understanding and managing volatile systems [6]. Researchers and practitioners have also recognized that maintainability and evolution of systems are mainly constrained by the architecture [7] and this is why it is beneficial to investigate change and their impacts during architecture design. Different CIA methods have been developed to specifically address changes and their impacts at architecture[1] level [3, 4]. However, architecture design complexities and high cost of change are the fundamental problems and mostly occur in evolvable systems such as Web systems [8].

Web systems generally tend to evolve constantly, like a garden - continue to evolve, change, and grow [9]. Additionally, it has been reported that the development processes significantly influence the evolvability of a system [10]. Therefore, we conjecture that architecture change problems in Web systems are both due to differences in the architecture and development process of Web systems as well as less focused characterisation of these differences during architecture CIA. Additionally, there is little research on industrial perspectives of using current architecture CIA approaches during Web systems evolution. To increase our knowledge and to investigate the current state of practice, we pose three research questions (RQs) for the work presented in this paper:

RQ1. What are the current industrial practices of architecture CIA in Web systems?
RQ2. What are the deficiencies/inadequacies of the current architecture CIA approaches?
RQ3. What are the high level features/needs that an architecture CIA approach should address?

[1] We have used a common definition of architecture proposed by IEEE [5] as '*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.*'

We believe that an important source of information to address our research questions is from those who make decisions on architectural changes on a regular basis. Therefore, we have conducted an interview based study with web designers/architects who had experience in architecture design changes and CIA.

Based on the study results presented in this paper, the contributions of this paper are twofold. First is to report the industrial perspectives on using current architecture CIA approaches during Web systems evolution. Second is to elicit the high level needs that a CIA approaches should have and thus to provide guidance for future research. The rest of the paper is structured as follows. Section 2 discusses CIA and the important characteristics of Web systems. Section 3 and 4 present interview study setting and results respectively. Section 5 provides the classification of needs to guide future research. Section 6 presents the limitation of research. Conclusions and future work can be found in section 7.

2. Background

There is growing recognition in the research literature to analyses changes and their impacts in a cost-effective, timely and precise manner [11-13] in order to avoid system deterioration [1]. This section begins with a general discussion on CIA approaches and subsequently has a more precise focus on architecture CIA approaches. Additionally, we also describe the specific Web systems' characteristics considered important for architecture level CIA.

2.1 Change Impact Analysis

CIA approaches are used to assess the effect of a change and its resultant ripple effects on other entities of a system. In general, CIA approaches can be divided into two classes namely dependency analysis and traceability analysis [1]. Interestingly, most of CIA research has been carried out based on dependency analysis to assess the effect of changes at code level, such as call graph [1], static and dynamic program slicing [11-13]. Besides code level slicing, architectural slicing techniques [2, 4] are further introduced. However both program level slicing and architecture level slicing techniques require extensive relationships and dependency information between entities. In architectural slicing techniques, architecture slices consist only of a design component and do not attempt to preserve the semantics of design specification [4]. Semantics of design and design rationale [3] are considered important for architecture level CIA.

Previous research highlights that most CIA approaches are developed based on the understanding of both system development process and system architecture [1, 8, 18]. It is thus reasonable to assume that if the development process or system architecture is different, then the CIA approaches needs to address these differences accordingly. This is particularly relevant in the context of Web systems, given that the literature indicates that these systems often have a specific set of characteristics related to differences both in the architecture [19-22] and the process through which they are developed [14-17]. The paucity of techniques for handling specific characteristics of Web systems related to CIA suggests that current architecture CIA approaches may not be suitable to adopt in Web systems context. In the next section we will discuss the specific characteristics of Web development process and Web systems architecture.

2.2 Characteristics of Web systems and their relevance in CIA

There is a growing body of literature regarding the differences both in the development process and architecture of Web systems[2]. It has been reported that Web development process is potentially different in many ways in comparison to traditional software development process [14-16]. Specific aspects that get exaggerated in Web systems include co-evolution of business needs/processes and solution under development [17] and rapidly changing requirements [18]. Furthermore, the architecture of Web systems tend to be characterized by a tighter linkage between business process with both a highly component-based architecture and complex information architecture [19]. In the next section we will specifically focus on these two characteristics of Web systems i.e. (i) co-evolution of business needs/processes and solution under development, (ii) tighter linkage between business process and architecture.

2.2.1 Co-evolution of business needs/processes and solution under development. In traditional software systems, where architecture design is often preceded by requirement elicitation, developers may use both an iterative and an incremental approach to gain feedback from clients as to whether or not a particular solution addresses the business needs. Conversely, in Web systems, the iterative and incremental development

[2] When we use the term Web systems we are referring to those systems which utilise web technologies as an integral element of a functionally complex system which typically incorporates interfaces beyond the organisational boundaries.

approach is not intended to evaluate solutions against a known set of business needs but rather actually help the client to understand emerging business needs and let developers formulate an architecture design solution [20]. As a consequence, in Web systems, business needs and their architecture solution mostly co-evolve [21]. Typically, this characteristic is most noticeable in the development processes that are adopted in commercial Web development. Industry's best practice tend to be highly incremental and, in particular, often removes the distinction between requirements specifications and design specifications [19]. Much of the requirements are actually captured during the architecture design process [22]. The reliance on architecture solutions indicates that we need to support a high degree of cohesion between changing business needs and their architecture solution. Any flaw while analysing the effect of a change from business process to architecture designing solutions may result in derelict architecture design. We posit that, specifically for Web systems context, most of CIA approaches do not attempt to address the high degree of cohesion between changing business needs and their architecture solution.

Furthermore, during the joint exploration of business process and architecture solution, identification of change impact may become more complex while addressing other characteristics of Web systems such as tight coupling between business process and supporting architecture [19].

2.2.2. Tighter linkage between business process/ model and supporting architecture. To improve the quality and functionality that Web systems offer, relatively frequent design changes are taken in order to meet both technological/infrastructure requirements and changing business needs [16]. Consequently, the architecture of Web systems is likely to be highly inhibited by the technological and infrastructure constraints (e.g. limitation of web browsers, data and documents format-XML/DTD, security and availability constraints etc) [19]. This places a high degree of limitations on the form an architecture solution may take and it is much more directly related to the business needs being addressed [19]. These constraints and limitations result in a tight coupling between business processes/models and their supporting architecture. As a consequence, changes in business process/models often lead to fundamental changes in the solution space and can also potentially influence supporting architecture [16].

The nature of Web systems is that any change in business process/models may also impact on different views of technical architecture [19] such as information architecture (IA). The information architecture (which covers aspects such as the content viewpoints, navigational structures, information flows and its behavior [23]) is substantially more sophisticated in information rich Web systems than that of traditional software systems [19]. Mostly, Web systems are meant to be used by a vast, variable user community- a large number of anonymous users with varying requirement, expectation and skill set [9]. It may not be true in the context of traditional software systems that are generally developed for limited and targeted users. The result is an increased emphasis on the IA in Web systems and how it relates to user interaction. The above discussion implies that IA is a reflection of rapidly changing business needs being addressed or the resultant business process [19].

Given the tighter linkage between business needs, processes and the supporting IA, any latent changes may negatively affect the success of web-based business. This is true largely due to the late identification of architecture level change impacts at subsequent stages of Web systems development [8]. We assert that most architecture related problems in Web systems are potentially incurred due to inadequate support provided by current architecture CIA approaches [24]. As a consequence, most of the change impacts are not adequately identified as early as they could be, and the later these impacts are recognized and managed, the more expensive and difficult the corrections. Given the lack of support (for identification of impacts) provided by current architecture CIA approaches, we advocate that early identification of change impacts is obviously needed to maintain a sophisticated and up-to-date IA, this consequently leads to high degree of satisfied user interaction and thus a successful Web system.

Surprisingly, there is little research on architecture CIA approaches to address the previously discussed characteristics of Web systems, which substantially hinders the progress in the area of change impact identification. To investigate this concern, we have designed an interview based study to explore the current architecture CIA practices, potential concerns web designers have while carrying out CIA activity and the possible solutions to address these concerns during Web systems evolution.

3. The Study

Considering the research questions of our study (see section 1) and available resources, we decided to use interviews as a data collection method to understand the industrial perspectives on current architecture CIA approaches. From the collected interview data, we have elicited the high level needs/features for an architecture CIA approach.

3.1 Study settings

In our study, the interviews were intended not only to explore what Web designers/architects thought and experienced regarding the state of practice during identification of architecture change impact, but also what features they believed an architecture CIA approach should have for Web systems evolution. Therefore, we developed interview questions which kept a focus on CIA practices, potential concerns and their possible solutions. The interview comprised of 12 questions developed from reviewing the published literature on CIA and Web systems. Furthermore, 10 questions were developed on the demographics of the participants. Demographic questions were designed to screen participants and to refine the data sets relevant for final analysis. 60% of our participants were from Australia and 40% from Asia. The participants' experience in the Web systems development varies between 3 years and 5 years with a median of 4 years. A median score for working as a Web designer or architect is 3 years and a median score for working with one organization (current or previous) is 3 years. An average number of co-workers on the current (or last) Web systems project is 8 peoples. 95% of the respondents have an IT related tertiary qualification. These demographics give us confidence that we have gathered data from practitioners who are experienced in Web systems development, especially in architecture changes and their impact analysis.

The first author conducted all the interviews of duration 60-90 minutes each. Some interviews were face-to-face and some were conducted on phone. To execute the questionnaires with web designers/architects we had selected a sample size based on convenience sampling [25]. Participants were selected mainly based on recommendation and their availability from a range of organizations mostly ISO and CMMI certified and doing business in education, health, banking, stock-exchange, tele-communication and public sectors between 5 and 25 years. However, a few participants were approached based on their role in architecture CIA during Web systems development. Sixteen participants were contacted via e-mail to participate. 10 replies were received and all agreed to participate. In fact, strict criteria (including 3-5 years of experience in Web development, a variety of size and the types of systems developed and particularly experience in architecture level change impact analysis), reduced the number of participants, therefore only a limited number of interviews could be completed. Before conducting the interviews, a preliminary version of the questionnaire was piloted internally to determine any overlaps in the questions, relevancy to the desired topics, as well as consistency in phrasing and clarity of explanation. This was achieved by conducting face-to-face interviews using the questionnaire with 2 participants who were external to the research. The pilot study participants had the added advantage of having significant experience in interviewing in their research and being interviewed as web designers/architects for this study.

3.2. Data Collection

Interviews are recognized as the most appropriate data collection method to be used for the acquisition of knowledge and in-depth experience of practitioners [26]. Additionally, interviews provide the best possible solution in terms of the effort required for data collection and analysis [27]. Indeed interviews enabled us to question the web designers/architects directly about their thoughts and opinions. Similarly, it allowed the web designers/architects the freedom to describe and reflect in detail their views and beliefs relevant to the research goals of our study. The interviews were recorded and transcribed by the researcher. Once the needs were identified from the content analysis, they were sent to the participants for verification and some responses were received that were further incorporated in the final list of needs.

3.3. Data Analysis

We have used content and thematic analysis to analyse interview transcripts. It has been identified from interview data that the overall CIA was unstructured and unclear, and the specific impact analysis tasks, namely change identification, tracing the location of impacts and their analysis were not well supported in current architecture CIA approaches - specifically when these approaches were adopted in Web systems context.

3.3.1 Change impact analysis Needs. A list of 25 needs is developed for architecture level CIA during Web systems evolution. At a basic level, all these needs emerged as a result of coding and thematic analysis of interview data, and finally the generalisation of analysis results into a comprehensive list of 25 needs. We had neither provided any pre-defined list of needs, nor the respondents had the freedom of itemising any specific needs. Furthermore, we have re-organised these 25 needs by using a frequency distribution approach.

3.3.1 Frequency distribution of CIA Needs. Frequency distribution [28] is used to organize and summaries data in a form that allows further interpretation and analysis. It also helped us to see how the data clustered around central concepts and the degree of differences between analysed data [28]. The frequency of occurrence for each need was examined and we found that several needs can be grouped together as types or kinds of something. In that case we moved them together and put them under a major need, the kinds or types of which they all represent. Therefore the list of identified needs is reduced from 25 to 11 needs and we tagged each with prefix NE (NE1-NE11) as shown in table 1. Furthermore, numerical indexing is used to structure needs and their sub-needs. For example, need NE-2 has sub-needs as NE-21 and NE-22 etc. We have interpreted the clustering of the needs from 5-1 (5= highest, 1= lowest) based on the frequency. We have called these 11 needs intended needs for architecture CIA as it represents a more comprehensive set.

4. Results

It can be seen from table 1 that high-clustered needs are concerned with (i) NE-2: the early identification of change impacts on information architecture (IA) and (ii) NE-4: the execution of the analysis task at IA level, which arguably are the fundamental aspects of CIA process. The medium-priority needs are concerned with (i) NE-3: activity-orientation, structured support, guidelines and (ii) NE-5: management of CIA activities. These results clearly indicate the demand for process support, in order to facilitate CIA activity while addressing the characteristics of Web systems (as described in previous section). It has been discussed that a total number of 11 needs have been developed after the systematic re-organization of identified needs. We have found 5 needs elicited from the research literature, but these are already among the 11, which confirms that our list of needs is indeed relevant and provides a broader coverage. However, these 11 needs vary in term of their closeness to impact analysis. For example some needs are concerned with change requests, some concerned with actual CIA activity, some concerned with process detail and support, and some concerned with tracing the location of impacts. Moreover, since CIA approaches were not formalised, no tool support was available at the time of study.

5. Classifications of Needs for CIA

The set of needs derived from interview data provides guidance on the possible classification of these needs into different focus areas. This classification supports a more specific characterisation of these needs while carrying out CIA activity. The following five focus areas can be formulated to group all the needs (mentioned in table 1) and their mapping area as shown in table 2.

1. Traceability & dependency analysis
2. Process related support for CIA
3. Effectiveness of analysis results
4. Design information reuse
5. Understanding specific Web systems characteristics by practitioners

The main purpose of classifying these needs is to provide a structure which can be used for the identification of desired needs (to be supported) and possible shortcomings of a selected CIA approach. The following sections explain in depth each focus area and link the related set of needs. Furthermore, the relevance of these needs with respect to Web systems characteristics is also explained.

5.1. Traceability & dependency analysis

Traceability is related to establishing and maintaining traceability links between system artifacts and other related items [1]. Particularly, to address the co-evolution of business needs and architecture solutions, the traceability link from changing business needs to IA level design entities is highly desirable (NE-1). In turn this traceability supports adequate identification of change impacts on IA. Furthermore, the need to develop dependency links is also essential for maintaining the dependency among different abstraction[3] levels of IA (NE-21). This, in turn, supports the prediction of impacted design entities resulting from an implemented change (NE-25). Additionally as an aid for keeping the system consistent related to a change, establishing and maintaining inter-tracing between different abstraction levels of IA (NE-22, NE-23) and intra-tracing within a specific abstraction of IA (NE-24) were elicited from all the interviews. Indeed, the task of analysing the ripple

[3] By abstraction of architecture we mean different view of information architecture such as structure (what), behaviour (how), locations (where) that are quite align with the concepts provided in [33].

Table 1. Intended Needs for Change Impact Analysis

NE-1 link business needs/process and information architecture (IA) entities to maintain the traceability between them	
NE-2 Early impact analysis at IA level support for identifying how:	
	NE-21 Architecture changes affect other part of architecture design
	NE-22 Changes made at a higher abstraction level affect components at a lower abstraction level
	NE-23 Changes made at a lower abstraction level affect components at a higher abstraction level
	NE-24 Changes made at any level affect components at the same abstraction level
	NE-25 To determine which design entities are necessary to modify after making a change
NE-3 Process development support for:	
	NE-31 Activity descriptions (steps)
	NE-32 Better and more robust analyses to perform by introducing support such as checklist
	NE-33 Activity (from NE-31) should be better guided and instructed (guidelines)
NE-4 Process working support for:	
	NE-41 Managing multi-level information architecture changes (structure, behavior, location etc)
	NE-42 Generating cross-reference(intra-level) IA information (structure to behavior to location)
	NE-43 Extracting information related to interconnections of IA components (intra-level)
	NE-44 Navigation between architecture knowledge and impacted design entities
	NE-45 Representation of IA intra-level (e.g. structure to behavior to location) and inter-level (e.g. structure to structure) dependencies between design entities
NE-5 Process management support for:	
	NE-51 Instantiating company level methods and procedures for systems development
	NE-52 Reporting to the validation of the process employed
	NE-53 A consistent and systematic way to identify architecture level change impacts
NE-6 Avoid overestimation (unnecessary change) and underestimation (overlooking change)	
NE-7 Support to employ design information and old impact analysis results as a knowledge base	
NE-8 Support to employ IA design decisions logs	
NE-9 Feedback mechanism used to evaluate the results of impact set after retesting affected system	
NE-10 Analyse impacts early (before detail designing) so that there can be many changes during systems evolution	
NE-11 Identify change impacts on IA resulting from changing business needs/process	

effect on different abstraction levels of IA (NE-2) and achieving an understanding of the nature of change in Web systems (NE-13) were reported to be among the most problematic tasks during CIA. Typically, these tasks were believed to stem directly and indirectly from inadequate traceability and dependency analysis. We believe that inadequate traceability and dependency analysis, in Web systems, makes the identification of change impacts more complex mainly due to tighter-linkage between business process and architecture.

5.2. Process related support for CIA

An important finding from the interview data was that a commonly agreed process for architecture CIA does not exist. It has also been observed that normally analysis of change impacts was left to individuals and was considered mostly ad hoc. As a result, during information architecture design (at early development stage), change impacts were not adequately analysed, processed and managed by the web designers/architects. Consequently, the demand for a CIA process to be employed at the early stage of Web systems development is highly desirable. Few other needs also emerged such as CIA process should be supported by a set of guidelines, be lean and not fixed.

Table 2. Mapping between CIA intended needs and area of classification

Classified Areas	Relevant needs
Traceability and dependency analysis	NE-1, NE-21, NE-22, NE-23, NE-24, NE-25
Process related support	NE-31, NE-32, NE-33, NE-41, NE-42, NE-43, NE-44, NE-45, NE-51, NE-52, NE-53
Effectiveness of analysis	NE-06
Design information reuse	NE-7, NE-08, NE-09
Understanding specific Web systems characteristics	NE-10, NE-11

The CIA process was described by the web designers/architects as “ad-hoc”, “haphazard”, “unsystematic”, and “flailing around” etc. Likewise, 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.

It has also been identified that there were certain activities that used to perform in almost every selected CIA approach, therefore developing a process is important 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.

Mostly, deficiencies in formalising CIA approaches were identified from all interview data and can be depicted by process related support (NE-2, NE-3, NE-4) as a major need. However, in few organisations official procedures were there for reporting issues to validate a particular change, but these procedures were not developed to address change impacts at architecture level (such as IA). 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 (NE-53) in order to support impact identification at architecture level. Surprisingly, approaches reported in research literature for CIA [2-4] were not part of Web systems development in most of the interview study. Consequently, the web designers /architects adopted their own approaches. Additionally, since the architecture 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 and checklist support (NE-31, NE-32), process guidelines (NE-33), information tracking and its representation (NE-41, NE-42, NE-43, NE-44, NE-45), process instantiation (NE-51, NE-53) and process validation (NE-52).

As discussed above there were certain activities used to perform in almost every selected CIA approach. Typically, these CIA activities mostly consist of high level tasks such as “examine and review change request”, “inspect associated documents”, “consult and discuss change with the stakeholders for identification of possible impacts”, “implement the identified change”, and “verify the change and their impacts”. Indeed, these tasks are common to the activities of generic CIA process model proposed by Bohner and Arnold [1]. Bohner and Arnold have not explicitly mentioned the importance of validating the impact analysis results. From the interview data, however, we have derived the need for process validation (NE-9) and posit this need as an improvement over Bohner and Arnold’s process model.

5.3 Effectiveness of analysis results

Mostly, it is desirable to perform the CIA in a way that all the relevant change impacts can be identified within the constraints which are reasonable from a system development point of view [1,29]. Effectiveness refers not only to identifying all the relevant impacts, but also to avoiding the problem of impacts overestimation or underestimation. Overestimation of impacts results in false impacts and may cause redundant work. Similarly in a worst-case scenario of underestimation (i.e. overlooked impacts that are due to be identified when a change is being made), these false impacts may cause an unexpected behavior of changed system. For example, one of the participants during the interview described:

…mostly we underestimate the impacts and therefore underestimating the consequence of making a change that 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 suggests that CIA approaches adopted in a Web systems context must support effective analysis results in order to reduce unwanted impacts and thus avoid expensive and complex modifications afterwards (NE-06). Furthermore, it also places a high emphasis on the reliability of impact set for a highly sophisticated IA, whilst keeping in view the tight-coupling between business processes and their supporting IA.

5.4. Design information reuse

Information related to design decisions is considered as important architectural knowledge. This information can play a crucial role when identifying change propagation, gauging the scope of impacts and exploring the dependencies between architecture design entities [30]. Typically, this design information consist of referenced architecture information, previous

CIA results, design decisions taken and the solution to implement those decisions [31].

During Web systems evolution, at the stage when business needs/processes and their supporting IA solutions begin to emerge, one of the participants described:

…as business requirements (specifically related to information architecture) get change, 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 methods do not adequately support impact identification at information architecture level.

The above implies that change impacts on IA level can not be predicted solely based on change requests specifications. Therefore, the design decision information (leveraged by both the co-evolution and the tight-coupling of business process and IA) is important to employ during CIA activity (NE-7).

It has been reported that current architecture CIA approaches mainly focus on changes to the architectural component and connectors [3, 4], and do not employ design decisions and other design information. Indeed, design decisions information mostly emerges during architecture design process [20] while addressing the tighter-linkage between business process and their supporting architecture. Typically, employing design decision information (NE-8) and sharing this information across different development groups can be very helpful for a well-informed decision making. Additionally, previous impacts results are recognised as important feedback for validating the estimated impacts (NE-9).

5.5 Understanding specific Web systems characteristics by practitioners

Interview data revealed that designers/architects faced a few problems during CIA such as understanding the system domain and more importantly understanding the specific characteristics of Web systems in relation to CIA. Two main characteristics contributing to the difficulties for systems understanding were identified.

- Co-evolution of business needs/processes and their architecture solutions are not well understood. The deficiency in understanding the co-evolution is partially due to the lack of considering the difference in the architecture and development process of Web systems. Furthermore, this situation leads to pose a major hindrance while analysing the roots of a change and its impacts.
- A single change in business process may typically lead to fundamental changes in supporting IA and it is mainly so due to tight-coupling between business process and system architecture. Architectural level tight-coupling in Web systems may complicate the impact analysis further, as the designers/architects have to track several combinations of design entities to analyse the effect of a change in a significantly complex and multi-level architecture.

In support of architectural tight-coupling in Web-based systems, one of the participants expressed

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

The need for system understanding deals primarily with an in depth understanding of the co-evolution of business process and architecture (such as IA) and this substantially supports to identify impact on IA (NE-11). Consequently, a better and adequate system understanding leads to an early identification of architecture level change impacts (NE-10). We have observed that other identified needs also tend to link with system understanding needs such as early identification of impacts reported as desirable by utilising design decisions and other design information (NE-44). Additionally in Web systems, an understanding of architectural level tight-coupling can provide an adequate visualization of related design entities (NE-45) and dependency relationships among them in a multi-level IA (NE-41, NE-42, NE-43). Both co-evolution and tight-coupling between business process and their supporting architecture are the two distinct characteristics of Web systems in relation to CIA as discussed in section 2.2. Overall we have concluded from interview data that the specific Web systems characteristics need to be recognised in order to address them adequately by CIA approaches.

6. Limitations

Like most interview based studies in software engineering, our study also faces reliability and validity threats. Following the guidelines provided in [32] and [27], we put certain measures in place to address validity and reliability issues. For example, our research instrument underwent rigorous evaluation by experienced researchers and practitioners. All the questions were tested in 2 pilot studies. The participants were assured of anonymity and confidentiality. However, completely eliminating the possibility of bias is difficult.

The participants were mainly from Australasian region. This may introduce another limitation as the study findings may not be generalisable globally. Furthermore, our study also suffers from the non-existence of a proven argument for an architecture CIA process specifically for Web systems evolution. Hence we consider this research as an investigative effort to draw some key conclusions that can help to establish some empirical support for future research directions and to develop and validate such endeavor.

7. Conclusions and Future work

In this paper we have presented the results of an interview based study. Typically, the findings provide adequate insights of current architecture CIA practices during Web systems evolution to address RQ1. Additionally, this study has gathered new information both about the potential concerns that web designers have whilst using current CIA approaches (RQ2) and designers' expectations towards an architecture CIA approach for Web systems (RQ3). Furthermore, we have elicited a list of high level features/needs deemed essential for a CIA approach. We have also presented the description and classification of each CIA needs accordingly. Indeed, this classification assisted us to relate a set of needs to each focus area, and to justify their relevance with the specific characteristics of Web systems where required.

In understanding the current practices (i.e. RQ1), we observed that a systematic CIA process or approach is largely not part of Web systems development, so web designers/architects tend to adopt their own approaches. However, to address RQ2, we reported that overall CIA activity for the identification of impacts at architecture (such as IA) was unstructured, inconsistent and unclear. This typically tends to reflect that CIA processes/approaches developed for traditional software system domains may not be suitable to adopt in other system domains such as Web systems [24]. Key sentences from previous sections are worth repeating here, given their significance, "Since the CIA approaches were not formalised, no tool support was available at the time of study. Therefore instead of tool support major focus was on the basic process functions such as descriptions of CIA activities and checklist support, traceability and dependency information tracking, design information and its representation, process guidelines, process instantiation and process validation". The major focus on different process related aspects mainly addresses RQ3 and suggests that an extension of existing CIA processes [1,3] can be beneficial both to address the specific characteristics of Web systems and thus to suitably adopt (extended CIA processes) during Web system evolution.

Findings of this study reveal focus areas to better guide our future efforts in the same line of research. We believe that the important aspects for future research in the same area are:

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

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 gap identified from the review of related research literature and architecture CIA approaches [24]. More importantly a number of major CIA needs, from our interview-based study, advocates that current CIA methods can be extended to address the specific characteristics of Web systems and thus to architecture evolution. Given the paucity of research focus on handling the characteristics of Web systems related to CIA, and based on industrial perspectives of current architecture CIA approaches as reported in this paper, we will develop a process model to improve current CIA practices. Our CIA process model can also adopt appropriate method to consistently perform some routine impact analysis tasks.

Acknowledgements

We thank all our interview participants who took time from their work days to be interviewed and provide feedback.

8. References

[1] S. A. Bohner and R. S. Arnold, *Software Change Impact Analysis* Los Alamitos, CA, USA: IEEE Computer Society Press, 1996.

[2] K. v. d. Berg, "Change Impact Analysis of Crosscutting in Software Architectural Design," in *ACE 2006 in collaboration with 20th European Conference on Object-Oriented Programming-ECOOP 2006,* Nantes, France, 2006.

[3] A. Tang, A. Nicholson, Y. Jin, and J. Han, "Using Bayesian belief networks for change impact analysis in architecture design," *Journal of Software Maintenance,* vol. 80, pp. 127-148, 2007.

[4] J. Zhao, H. Yang, L. Xiang, and B. Xu, "Change impact analysis to support architectural evolution " *Journal of Software Maintenance* vol. 14, pp. 317-333, 2002

[5] IEEE, "IEEE Recommended Practice for Architecture Description of Software-Intensive System," IEEE Computer Society, New York, 2000.

[6] P. Clements, R. Kazman, and M. Klein, *Evaluating Software Architectures: Methods and Case Studies*: Addison-Wesley, 2002.

[7] P. Bengtsson, "Towards Maintainability Metrics on Software Architecture: An Adaptation of Object-Oriented Metrics" in *First Nordic Workshop on Software Architecture (NOSA'98),* Ronneby, 1998.

[8] Z. Xiaoyu and C. Mei-Hwa, "Maintaining Multi-Tier Web Applications," in *IEEE International Conference on Software Maintenance*, Paris, France: IEEE, 2007.

[9] S. Murugesan and A. Ginge, "Web Engineering: Introduction and Perspectives " in *Web Engineering: Principles and Techniques*. vol. 1, W. Suh, Ed. PA, USA: Idea Group Inc. Publishing, 2008.

[10] S. Cook, R. Harrison, M. M. Lehman, and P. Wernick, "Evolution in software systems: foundations of the SPE classification scheme," *Journal of Software Maintenance and Evolution,* vol. 18, pp. 1-35, 2006.

[11] G. Canfora and L. Cerulo, "Fine grained indexing of software repositories to support impact analysis" in *Proceedings of the international workshop on Mining software repositories,* Shanghai, China, ACM, 2006

[12] K. B. Gallagher and J. R. Lyle, "Using Program Slicing in Software Maintenance" *IEEE Transactions on Software Engineering,* vol. 17, pp. 751-761, 1991.

[13] L. Huang and Y.-T. Song, "Dynamic Impact Analysis Using Execution Profile Tracing," *in Fourth International Conference on Software Engineering Research, Management and Applications,* Washington, USA: IEEE Computer Society, 2006.

[14] A. McDonald and R. Welland, "Web Engineering in Practice," in *Fourth WWW10 Workshop on Web Engineering,* Hong Kong: ACM Press, 2001.

[15] R. S. Pressman and D. Lowe, *Web Engineering- A practitioner's Approach,* New York: McGraw Hill, 2008.

[16] D. Lowe and J. Eklund, "Development issues in specification of Web systems," in *6th Australian Workshop on Requirement Engineering,* Sydney, Australia, 2001.

[17] N. Yusop, D. Lowe, and D. Zowghi, "Impacts of Web Systems on their Domain," *Journal of Web Engineering,* vol. 4, pp. 313-338, 2005.

[18] D. Lowe and B. Henderson-Sellers, "Impacts on the development process of differences between web systems and conventional software systems," in *Int. Conf. on Adv. in Infrastructure for Electronic Business, Science, and Edu. on the Internet,* L'Aquila, Italy, 2001.

[19] D. Lowe and B. Henderson-Sellers, "Characterising Web Systems: Merging Information and Functional Architectures," in *Architectural Issues of Web-Enabled Electronic Business*, N. S. Shi and V. K. Murthy, Eds. London: Idea Group Publishing, 2003, pp. 227-293.

[20] D. Lowe and J. Eklund, "Client Needs and the Design Process in Web Projects," *Journal of Web Engineering,* vol. 1, pp. 23-36, 2002.

[21] G. Sinha, "Build a Component Architecture for E-Commerce," in *e-Business Advisor,* 1999.

[22] L. Gates, "Analysis and Design: Critical yet Complicated," in *Application Development Trends,* 2001.

[23] L. Rosenfeld and P. Morville, *Information Architecture for the World Wide Web*, 3 ed. CA: O'Reilly Media Inc, 2006.

[24] Z. Mehboob, D. Zowghi, and D. Lowe, "An Approach for Comparison of Architecture Level Change Impact Analysis Methods and their relevance in Web Systems Evolution," in *20th Australian Software Engineering Conference,* Gold Coast, Australia: IEEE Computer Society, 2009.

[25] C. Robson, *Real world research: a resource for social scientists and practitioner-researchers,* Oxford: Blackwell Publishers Ltd, 2002.

[26] R. Agarwal and M. R. Tanniru, "Knowledge acquisition using structured interviewing: an empirical investigation," *Journal of Management Information Systems,* vol. 7, pp. 123-40, 1990.

[27] S. Kvale, *Interviews: An Introduction to Qualitative Research Interviewing,* London: Sage Publication Inc, 1996.

[28] L. Cohen and M. Holliday, *Practical Statistics for Students*, 1st ed. London: SAGA Publications Company, 1996.

[29] H. Kagdi and J. I. Maletic, "Software-Change Prediction: Estimated + Actual," in *Second Int. IEEE Workshop on Soft. Evolvability (SE'06),* Philadelphia, Pennsylvania, USA: IEEE Computer Society, 2006.

[30] J. Bosch, "Software Architecture: The Next Step " in *First European Workshop on Software Architecture-EWSA'04*, St Andrews, UK, 2004, pp. 194-199.

[31] P. Avgeriou, P. Kruchten, P. Lago, P. Grisham, and D. Perry, "Architectural knowledge and rationale: issues, trends, challenges," *SIGSOFT Softw. Eng. Notes,* vol. 32, pp. 41-46, 2007.

[32] R. K. Yin, *Case Study Research: Design and Methods,* 2nd ed. Thousand Oaks, CA: Sage Publication, 1994.

[33] X. Kong, L. Liu, and D. Lowe, " Separation of Concerns: a Web Application Architecture Framework", *Journal of Digital Information*, vol. 6, no. 2, 2005.