

Supporting Integrated Dependency Model for Change Impact Analysis in Web systems

Zafar Mehboob [\[HREF1\]](#), PhD Research Student, Faculty of Information Technology [\[HREF2\]](#), PO Box 123, University of Technology, Sydney [\[HREF3\]](#), NSW, 2007. Zafar@it.uts.edu.au

Didar Zowghi [\[HREF4\]](#), Associate Professor, Faculty of Information Technology [\[HREF2\]](#), PO Box 123, University of Technology, Sydney [\[HREF3\]](#), NSW, 2007. Didar@it.uts.edu.au

David Lowe [\[HREF5\]](#), Professor, Faculty of Engineering [\[HREF6\]](#), PO Box 123, University of Technology, Sydney [\[HREF3\]](#), NSW, 2007. David.Lowe@uts.edu.au

Abstract

Change Impact Analysis (CIA) is typically used to identify the consequences of making a software change by utilizing the dependency and traceability information with regard to that change. Previous research highlights that CIA techniques are developed by both an understanding of interdependencies (between system components) as well as interaction of architectural modules. Given that Web systems (WS) are usually developed as a multi-tier and components based system, it is postulated that their multi-tier interaction and components interdependencies require different CIA methods. In this paper we propose extending current CIA techniques with the concept of an integrated dependency model to accommodate multi-tier and interoperability dependencies in WS.

Introduction

Change Impact Analysis (CIA) is a crucial part of change management process that mainly helps to reduce the cost and risk ([Sneed, 2001](#)) caused by unanticipated impacts due to a change. Without taking into account integrated system interdependencies, developer may miss some of the impact and thus incurs high cost and error-prone software. Cost of unidentified impacts may increase with the number of frequent and release change in a multi-tier and integrated application such as WS ([Mookerjee, 2005](#)). This is largely true as much of the impacts go unnoticed due to inadequacy of existing CIA techniques ([Sneed, 2001](#)). Intuitively in WS, changes in one tier can affect the other tier due to inter-tier interactions, which exacerbate the issue of identifying change propagations. Furthermore, WS often endure frequent changes including usability enhancement at Graphical User Interface (GUIs), components upgrading at logic level, and configuration and databases updates. To cater frequent changes and its propagations effectively in multi-tier architecture depends on a thorough and cross-tier identification of dependencies. Previous research highlights that CIA techniques are developed by an understanding of architecture interdependencies ([Jacyntho et al., 2002](#)). Thus, it is beneficial to augment Current CIA techniques with an integrated dependency model, given that the architecture of WS is fundamentally characterized by complex interdependencies and multi-tier interactions ([Xiaoyu and Mei-Hwa, 2007](#)).

Web systems and Change Impact Analysis

WS are a typical example of multi-tier application. After a change is made to WS, the resulting change propagation can affect other portions of the system across different tiers. To this end, a slicing technique has been introduced to analyse components' dependency ([Pan et al., 2005](#)) at logic tier. Additionally, other approaches such as ([Galliers et al., 1999](#)) and ([Xiaoyu, 2006](#)) are also developed for presentation and logic tier respectively. A recent multi-tier systems analysis approach has been

proposed but it is limited to regression testing ([Xiaoyu and Mei-Hwa, 2007](#)) with an associated high cost for updating dependency information. None of these techniques are comparatively suitable for Web specific CIA, where multi-tier interaction and interoperability dependencies are quite intrinsic. Therefore, the next natural step will be to explore how relevant dependency information can be identified to cater for multi-tier interaction and interoperability issues, thus to facilitates CIA in WS context.

Dependency information in Web systems

WS typically use a client server architecture that generally consists of multi-tier namely presentation tier (user interface), logic tier (functional process logic and business rules), and data tier (data storage and data access).

The presentation tier contains graphical user interface (GUI) code along with GUI operation. Web pages can be represented by GUI object to cater for the interaction and operation of GUI events. A GUI object model consists of GUI object as a node and edges to model HTML structure, web objects and their attributes. A node is linked to other node(s) through three types of dependency relationship including control, data and link dependencies ([Xiaoyu and Mei-Hwa, 2007](#)). This GUI object model directly provides a prospect to identify and analyze the dependencies among GUI objects that can be potentially useful for CIA

The logic tier uses various programming languages and COTS components to implement the business and computation logic. The structural dependency and interaction at logic tier can be represented by system dependence graph (SDG) ([Pan et al., 2005](#)). Additionally, SDG also helps to depict dependency information about call returns, control of GUI event and event loop. In general SDG structure consists of nodes and edges and thus facilitates the study of various types of dependencies including control, data and parameter in/out between two entities at logic tier.

The data tier usually consists of database servers and data repositories. External data persists in external devices and can be shared among multiple application or across different web pages ([Abiteboul et al., 2000](#)), thus create a network of data dependency which is not catered by data models ([Xiaoyu and Mei-Hwa, 2007](#)). Data model are normally represented by database schema for database table and XML schema for XML files. Each piece of data from these data models can be treated as a data object and thus can be represented by a data object dependency model to cater data, control dependencies and links/actions by the external data.

Additionally, due to interoperability dependencies, introduction and replacement of Web services and middleware components may create a range of issues such as incompatibility, incorrect exchange of data, execution mismatch and non-compliance to the technology standards ([Bohner, 2003](#)). However, the formalism provided in web services representation in the form of extensible mark-up language (XML) can be used to extract and analyse interoperability dependencies ([Bohner, 2001](#)). Moreover, Simple Object Access Protocol (SOAP) and Universal Description-Discovery-Integration (UDDI) along with Web service stake offer a wealth of relevant interoperability dependency information (such as service and operation bindings between client and server, data exchange and semantic alignment) that can be analyzed to provide better visibility into WS ([Schlimmer, 2002](#)).

Integrated dependency model

In this section, we give an overview of inter-tier interaction and referred to it as an integrated dependency model as shown in figure 1. This dependency model shows a comprehensive approach to represent intra-tier (as discussed in previous section) and inter-tier dependence relationships in a multi-tier structure.

Any GUI object from presentation tier can call logic tier component for business computation or to execute functionality. Similarly, any components at logic tier may direct users to a web page (at the presentation tier) as a result of invoking page or page return. Similarly, parameter in/out dependency is possible if any data is passed as input from presentation tier to logic tier and any output data (result of computation or business logic) is transmitted from logic tier to presentation tier. Finally, a link dependency persists in case where any attribute or property of GUI object leads to the logical functionality at the logic layer (for example as a result of GUI event). Thus, three types of dependency relationships namely call, parameter in/out and link are possible between presentation and logic tier as illustrated in figure 1.

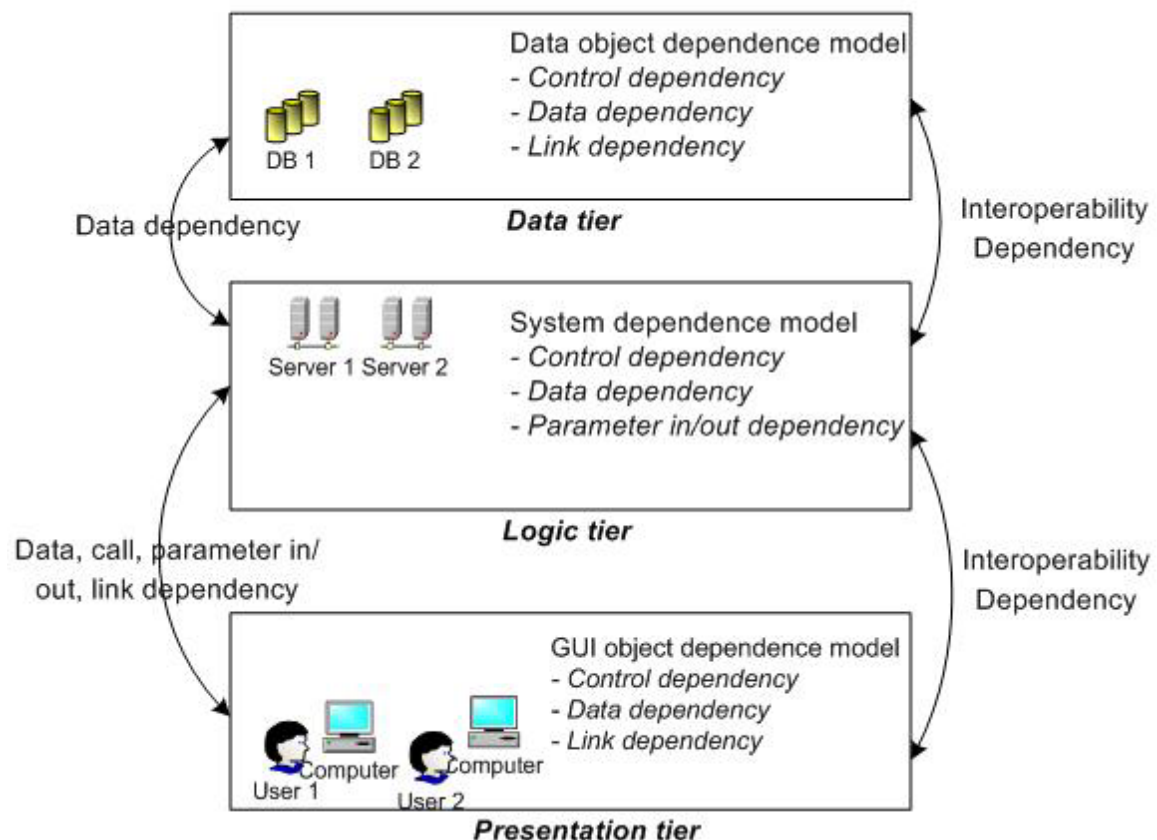


figure 1. Integrated dependency model for multi-tier Web systems

Furthermore, our integrated dependency model shows that there exist data dependencies between data tier and logic tier. As a GUI object can query some information, processed by logic layer, requested from data base and then query result is fetch by logic tier and finally send to presentation tier. Data dependency represents data fetching, passing and processing from data tier to logic tier and vice versa. Figure 1 illustrates an integrated dependency model in multi-tier WS.

Beside the dependencies as discussed above, there are interoperability dependencies from one tier to another tier(s) as illustrated by our proposed model. For example, components at logic tier can be distributed to different operating systems and run with different web servers. Moreover at presentation tier, different versions of web browsers can be run. All of this heterogeneity makes it harder for Web systems' components that are compatible with one another and that inter-operate easily and correctly.

Conclusions and Future Work

We have presented the idea of integrated dependency model for WS. This integrated and comprehensive dependency model will facilitates to depict different types of dependency relationship at multi-tier WS. This model also provides a broad and better visibility of dependencies in WS, thus reveals change impacts and will assist in reducing overlooked impacts. In our future work, we will improve this integrated dependence model in accordance with specific characteristics of WS and further develop a framework with lightweight instrumentation.

References

Abiteboul, J., P. Buneman, and D. Suciu (2000). *Data on the Web:From Relations to Semistructured Data and XML*. Morgan Kaufmann, San Francisco, California.

Bohner S. A.(2003). "Extending Software Change Impact Analysis into COTS Components,"in *Proceedings of the 27th Annual NASA Goddard/IEEE Software Engineering Workshop (SEW-27'02)*, p.175-182.

Bohner S. A.(2001). "DFAS on the XML Road to Better Interoperability," *White paper for Defense and Accounting Service*, Crystal City,VA.

Galliers, J., A. Sutcliffe, and S. Minocha (1999). "An impact analysis method for safety-critical user interface " in *ACM Transactions on Computer-Human Interaction (TOCHI)* v.6 n.1 p.341-369.

Jacyntho, M. D., D. Schwabe, and G. Rossi (2002). "A Software Architecture for Structuring Complex Web Applications" in *Journal of Web Engineering* v.1 n.1 p.37-60.

Mookerjee, R. (2005). "Maintaining Enterprise Software Applications" in *Communications of the ACM* v.48 n.1 p.75-79.

Pan, Y., D. Pan, and M.-H. Chen (2005). "Slicing Component-Based Systems" in *10th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'05)* Shanghai, China.

Sneed, H. M.. (2001). "Impact Analysis of Maintenance Tasks for a Distributed Object-oriented system" in *IEEE International Conference on Software Maintenance (ICSM'01)* Florence,Italy, 2001.

Schlimmer, J. C. (2002). "Web Services Description Requirements" Web Services Description Working Group report. Available online [\[HREF7\]](#) .

Xiaoyu, Z., and C. Mei-Hwa (2007). "Maintaining Multi-Tier Web Applications" in *IEEE International Conference on Software Maintenance- ICSM 2007* Paris, France.

Xiaoyu, Z. (2006). "Modeling Multi-tier web Applications for testing and maintenance" Ph.D. thesis, State University of New York, Albany, New York, USA.

Hypertext References

HREF1
<http://it.uts.edu.au/contact/view/zafar>

HREF2
<http://www.it.uts.edu.au/>

HREF3
<http://www.uts.edu.au/>

HREF4

<http://it.uts.edu.au/contact/view/didar>
HREF5
<http://services.eng.uts.edu.au/~dbi/index.htm>
HREF6
<http://www.eng.uts.edu.au/>
HREF7
<http://www.w3.org/TR/ws-desc-regs/>

Copyright

Zafar Mehboob, Didar Zowghi and David Lowe, © 2008. The authors assign to Southern Cross University and other educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to Southern Cross University to publish this document in full on the World Wide Web and on CD-ROM and in printed form with the conference papers and for the document to be published on mirrors on the World Wide Web.