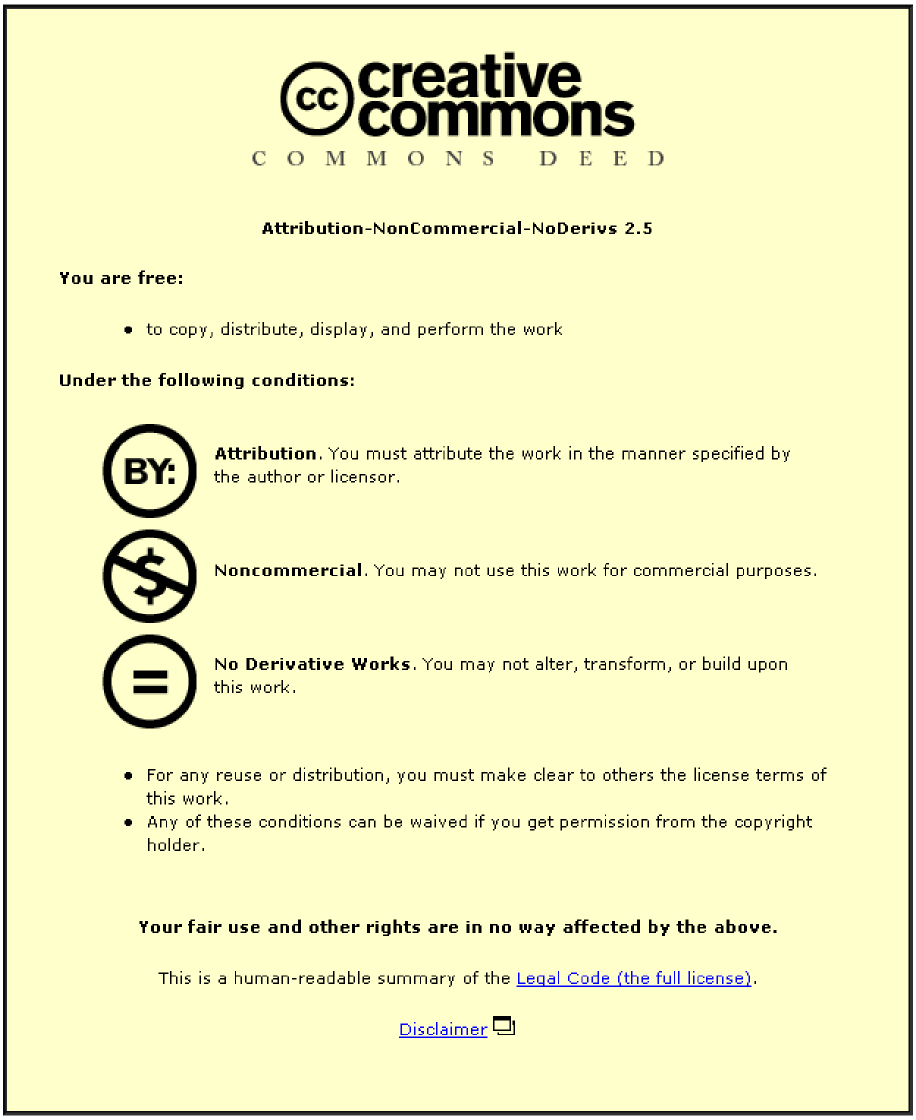


An Integrated Requirements Management System for Construction Projects

A.K. Jallow, P. Demian & A.N. Baldwin
Civil and Building Engineering Department, Loughborough University, UK

C.J. Anumba
Department of Architectural Engineering, The Pennsylvania State University, USA

Abstract

Within the AEC/FM Industry, the current paper-based system used to manage client requirements information and the change request process lacks efficiency and effectiveness. Traditionally, requirements management has principally been focused at the early stages of a construction project where elicited client requirements information is used as the basis for design and does not extend to later phases. Links between corresponding requirements at different phases do not exist which makes traceability difficult. There are no methods to keep track of client requirements and the changes in a satisfactory way that take a whole lifecycle approach. Construction organisations, like other engineering related businesses, are turning to computer systems (to replace current manual and paper intensive processes) in their quest for practical ways to facilitate requirements information management. This paper presents a prototype system for client requirements information management taking a whole lifecycle approach. It comprises of integrated components made up of a repository and change management system (CMS). The prototype was developed following case studies of construction projects and interviews with construction experts in order to fully understand the need for such a system. The applicability and effectiveness of the prototype will be validated using a focus group made up of various construction stakeholders and academic practitioners.

Keywords: Requirements information, Requirements management, Changes, Change management, Construction Projects.

1 Introduction

Initial client requirements are generated early in a construction project, during the preparation phase. During 'briefing', requirements are documented. Briefing is regarded as an on-going process in construction projects (Worthington, 2000) and initial requirements are later transformed into different levels of detail and content at various phases of the project. Traditionally, the brief is not stored in a centralised repository which provides collaborative access by project teams. Instead, each team holds their own copy of the brief. As the project progresses through the life phases, changes to the brief occur by the addition of new requirements or amendments of existing ones. This could be as a result of client's change of mind or other factors such as design decisions, materials, soil and other environmental conditions, regulatory issues, etc. At completion of a project, a substantial number of changes might have occurred to the brief. At that stage, clients usually assess the quality of the building against their requirements incorporating *all* the changes they have requested throughout the construction project. As a result of the evolutions of the client requirements, comparing the completed building with the initial brief may not be too accurate. It is thus important that these requirements are sufficiently recorded and tracked along the project lifecycle and during operation. Similarly, Kiviniemi (2005) highlighted that, because of inadequate documentation of the requirements changes, it is difficult to find the approved requirements changes, and the end-users may assess the building against the original, out-dated requirements. Thus it becomes paramount not only to document the initial brief but also all additions and amendments. This documentation should be done in such a way that it captures in detail the requirements defined at each phase of the project according to a standardised structure and format. Traceability between the requirements is important and "even a simple active link between the client requirements and design tools can increase the use of requirements documentation throughout the design and construction process and facilitate necessary updates of the client requirements" (Kiviniemi, 2005). Individual requirements that are related but are represented at different phases should be mapped together to provide links between them to enable traceability. The storage of the documented requirements information should facilitate shared access since construction projects are collaborative in nature with different teams coming together to fulfill a project. The storage mechanism should also provide for amendments and additions to the requirements. Many people are involved in the change process and the iteration that a change request goes through may be complex and cumbersome to coordinate and control. Requirements information generated as a result of the changes should be reflected to the original brief in order to provide continuous capture and maintenance of the requirements information. Historical information of client requirements and their transformation from one project phase to another should be documented and made available through-life of a building.

Comprehensive case studies of construction projects focusing on client requirements management were conducted through semi-participant observations. Interviews with selected stakeholders were held and a focus group comprising of construction experts and academics was also convened. The results indicated the need for a requirements management system that takes a lifecycle approach. As a result, a prototype system was developed based on the Enterprise Requirements Information Management Framework [eRIM] (Jallow, et al., 2010). It started with a needs analysis and the gathering of the business requirements. Soft Systems Methodology (SSM) was applied and a rich picture produced to help in the description of the problem situation and define a possible solution. Unified Modelling Language (UML) was also applied and a Use Case, Activity and a Class diagrams were produced in order to design and develop the prototype system.

2 Related Work

2.1 Client requirements information management

Construction projects produce large amounts of data and information from different sources. This is used at various phases of the project. The Industry is known to have '*silos*', computer systems used to manage such information. Most of these systems are disparate and operate on their own processes, data and users, making information sharing and exchange difficult. Rujirayanyong and Shi (2006) indicate that a great amount of operational data are generated by construction projects distributed across various functional systems which could be potentially useful for future projects but not widely collected and centrally stored. Requirements information needs to be centrally stored to facilitate shared access.

Information management including centralised storage, sharing and exchange is an important element of requirements management in construction projects. It is vital that real-time requirements information is passed to those who need it. A few computer systems have been deployed within the construction industry to help management of client requirements information management but none has taken an information-centric lifecycle approach with a combination of a requirements change management process as an integrated system. Many commercial requirements management software tools exist such as: DOORS from Telelogic, UGS Teamcenter, Slate and Caliber-RM. These are generic and have been developed for use across different industry sectors. However, none have become widely adopted in the construction industry. Traceability of requirements and their changes should be managed carefully to enable dissemination and correlation with the original requirements. Current state-of-art doesn't support this. Traceability is important to facilitate impact analysis of changes. We argue that a properly developed and maintained lifecycle requirements information management system will support collaboration and shared access to the requirements, traceability, impact analysis, and change process management to contribute to delivering successful projects.

2.2 Business process management (BPM)

According to TIBCO (2006), "BPM is a change management and system implementation methodology that enables the continuous comprehension and management of business processes that interact with people and systems, both within and across organizations." BPM is both a management method and computer systems defined as sets of methods and tools that can be used to coordinate operational business processes between people and systems (Garimella et al., 2008; Tanrikorur, 2007). Oracle (2008) states that "BPM emerged as both a management principle and a suite of software products focused on bridging diverse systems, people, and processes." A Business Process Management Suite/System (BPMS) is an integrated collection of software technologies that enables the control, management and necessary continuous improvement of business processes. BPM Suites (BPMS) are software components that coordinate tasks and synchronize data across existing systems and also help coordinate human process activities, streamlining tasks, triggers, and timelines related to a business process, assuring they are completed as defined by a process model. In the context of the client requirements change process, which is process-driven management activity (Jallow et al., 2010), BPM can make the process more efficient, agile, and visible by ensuring that the change authorisation process is explicitly defined, monitored over time, and optimized for maximum productivity. Process models/maps define the workflow of business processes through which business activities are executed. Business Process Modelling Notation (BPMN), a graphical representation language of business processes is an industry standard used in BPMS to produce process maps for subsequent implementation. Business processes define the order of work to be carried out and relate to the efficiency and effectiveness of a company; the better the processes are defined and managed, the more efficiently a company can operate (Juric and Pant, 2008). Most organisations' business processes are performed manually through interaction of people and resources and BPM can help automate these activities for better efficient and effective performance. BPM provides flexibility and agility of business processes. Business processes in most cases are performed both internal and external of organisations and should include collaborative capabilities and to connect with the information needed to get things done (Rittinghoue and Ransome, 2010).

2.3 Service oriented architecture (SOA) and web services

History has shown that integrating distributed heterogeneous computer systems to enable them to share and exchange information is difficult to achieve. This is as a result of the different programming languages and IT platforms. An integration middleware such as the Common Object Request Broker Architecture (CORBA) needs to be implemented to enable integration and interoperability between such systems. SOA philosophy has emerged as a promising solution to system-to-system integration problems. It takes the approach of services where separate system functionalities can be developed as a distinct service which can be invoked and consumed independently by other external systems (Sabri, et al., 2007). This enables *loose coupling* where heterogeneous systems can function and operate independently but can share and exchange information using these services.

Web services are described and presented in a Web Services Description Language (WSDL) document which is based on an Extensible Markup Language (XML) document style. According to Erl (2004), WSDL is a W3C specification providing the foremost language for the description of web services definition. WSDL describes a web service, listing its methods, the data types used for all parameters and return values and the supported methods of communication (Ferrara and MacDonald, 2002). Simple Object Access Protocol (SOAP) is a technology originally conceived to bridge the gap between disparate and distributed systems and has evolved into the most widely used and supported messaging format and protocol for use with web services (Erl, 2004). SOAP uses XML style to specify its messaging format and uses Hypertext Transmission Protocol (HTTP) for message transmission between a service requester and provider (Sabri et al., 2007).

3 Prototype System

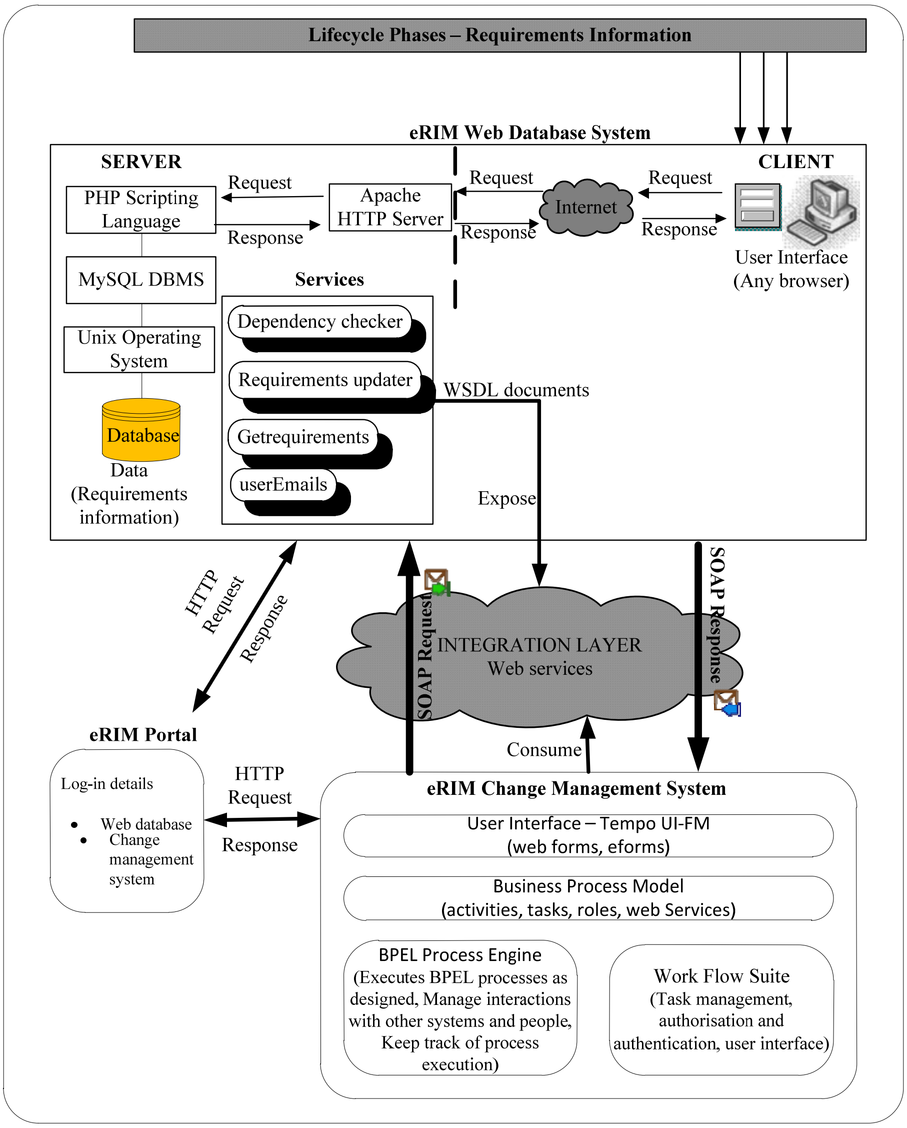
A prototype integrated requirements management system based on eRIM was developed. A pre-contract requirements management scenario of one of the case studies was used to demonstrate the potential of the system. The scenario is that client requirements are elicited and documented in a text document using Microsoft Word as a brief. This document is distributed to various parties (the Project Management Board, Consultant Project Manager, Internal Project Manager, and the Architect) in different formats and media. The Architect commences design and periodically, design meetings are held to review progress and checks that the drawings fulfill the client requirements. During this process, suggestions are made by the Architect which could result in additions of new requirements or amendments of existing ones. These become changes to the requirements which have to go through a change order process. The client or any member of the team can request a change by filling in a *request for change (RFC)* form. The form details information such as the proposed changes, reason for change, effect on time and budget amongst others. It is sent to the *consulting firm* that manages the project on behalf of the client with all other supporting documents. This is done by posting the hardcopy or emailing the electronic files. After receiving the request, a master RFC number is allocated. The Quantity Surveyor (QS) and the Consultant Project Manager review the request and provide cost and time implications where necessary. This review however, cannot be completed without checking for dependencies and related requirements. This is carried out by manually checking through the brief and other related documents stored in various systems. No matter what the implications (negative or positive impact) are, the change request would then proceed to the client for approval and the change initiator is informed of the decision. Situations were observed when a request is not approved in the first iteration and would require more information from the initiator before final approval is granted.

3.1 The system architecture

Based on the scenario above, analysis indicated that a centralised storage of client requirements information and a change management system were needed. Open source software and technologies were used for the development of the prototype. A web database system was designed and developed to serve as a central repository of client requirements and to enable shared access through the Internet. MySQL was used as the *Relational Database Management System (RDBMS)*, *Apache* as the S*erver* to provide Hypertext Transmission Protocol (HTTP) services to enable access through the Internet and Hypertext Preprocessor (PHP) as the web scripting language to develop the web interface.

A change management system was developed to provide cohesive coordination and control of the change request process. *Intalio Works BPMS*, comprising a *Designer/modeller* & *Server* was used as the BPMS to develop the CMS. The designer/mpdeller was used to produce the process model which was subsequently transformed into a BPEL process based on the BPMN notation. The BPEL process was deployed and executed on *Intalio Server* which provides process engine and workflow services.

To facilitate integration and interoperation between the two systems (*web database and change management systems*), web services (*'dependency checker'*, *'requirements updater'*, *'Getrequirements'* and *'useremails'*) were developed as application services that perform specific individual tasks within the database. The WSDL documents of these web services were published to the internet and made it possible to reuse them as software components or services in other systems. A MySQL database connection was established to enable the invoked web services to interact with the web database system.



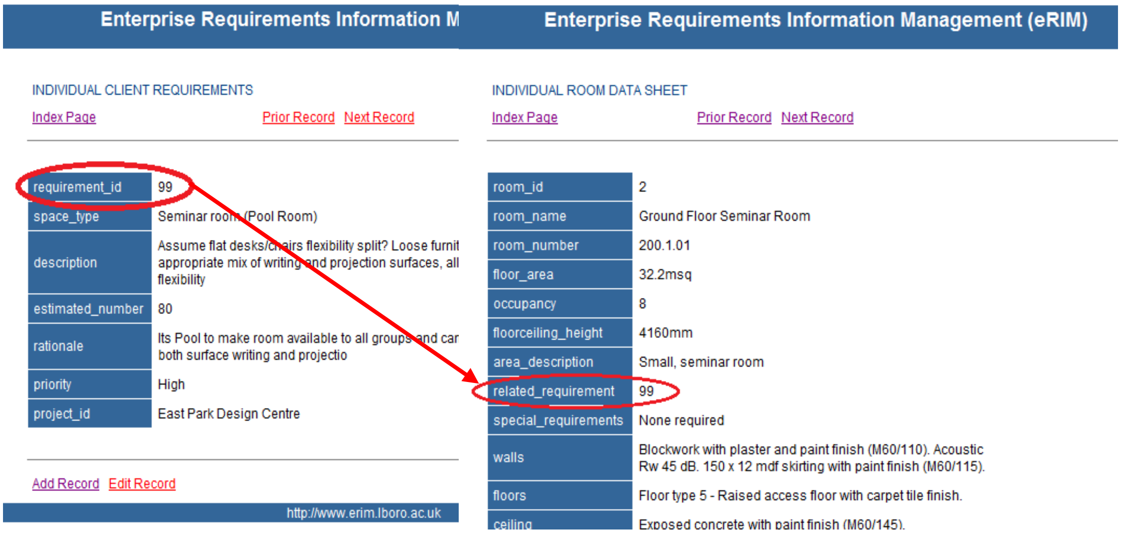
Lifecycle Phases – Requirements Information
eRIM Web Database System
SERVER
CLIENT
PHP Scripting Language
Request
Response
Apache HTTP Server
Request
Response
Internet
Request
Response
User Interface (Any browser)
MySQL DBMS
Services
Unix Operating System
Dependency checker
Requirements updater
WSDL documents
Getrequirements
userEmails
Database
Data (Requirements information)
Expose
HTTP Request
Response
SOAP Request
SOAP Response
INTEGRATION LAYER
Web services
Consume
eRIM Portal
Log-in details
Web database
Change management system
HTTP Request
Response
eRIM Change Management System
User Interface – Tempo UI-FM (web forms, eforms)
Business Process Model (activities, tasks, roles, web Services)
BPEL Process Engine (Executes BPEL processes as designed, Manage interactions with other systems and people, Keep track of process execution)
Work Flow Suite (Task management, authorisation and authentication, user interface)

Figure 1: Architecture of the prototype requirements management system

A web-based portal (eRIM portal) was developed to serve as the *'central access point'* to both systems, eliminating the need to switch between different systems. Both systems can be accessed using a single log-in for authentication. Liferay, an open source enterprise portal and social collaboration software was used to develop the portal. Integration between the two was achieved through the use of web services. Information flow (share and exchange) was achieved through SOAP message request and response mechanisms. Figure 1 shows a graphical presentation of the system architecture.

3.2 *Repository*

As discussed earlier, a MySQL database was developed to serve as the requirements repository. Client requirements rules for business management were analysed to produce the database schema. For the purpose of this prototype, only the tables and views necessary to process requirements information at preparation and design phases of a construction project are implemented and demonstrated. A table (‘*requirements*’) was created to hold the initial brief information at preparation phase and another table (‘*roomdatasheet*’) to hold room data sheet information at design phase. In order to provide traceability, a link was created between the two tables. This is a catalyst for creating *‘joins’* between related requirements. Accordingly, a unique identifier was used as a ‘*mapper*’ linking each requirement to a set of related room data sheets. The mapper is a *‘primary key’* which uniquely identifies a requirement. In this way, both forward and backward traceability can be provided between requirements information at each project phase. Figure 2 shows a view of a requirement and room data sheet to demonstrate how the two are linked. All changes to the requirements are stored in a separate table (‘*changes*’) which is also linked to the ‘*requirements*’ table.



Enterprise Requirements Information M

INDIVIDUAL CLIENT REQUIREMENTS

Index Page Prior Record Next Record

requirement_id	99
space_type	Seminar room (Pool Room)
description	Assume flat desks/chairs flexibility split? Loose furnit appropriate mix of writing and projection surfaces, all flexibility
estimated_number	80
rationale	Its Pool to make room available to all groups and car both surface writing and projectio
priority	High
project_id	East Park Design Centre

Add Record Edit Record

http://www.erim.lboro.ac.uk

Enterprise Requirements Information Management (eRIM)

INDIVIDUAL ROOM DATA SHEET

Index Page Prior Record Next Record

room_id	2
room_name	Ground Floor Seminar Room
room_number	200.1.01
floor_area	32.2msq
occupancy	8
floorceiling_height	4160mm
area_description	Small, seminar room
related_requirement	99
special_requirements	None required
walls	Blockwork with plaster and paint finish (M60/110). Acoustic Rw 45 dB. 150 x 12 mdf skirting with paint finish (M60/115).
floors	Floor type 5 - Raised access floor with carpet tile finish.
ceiling	Exposed concrete with paint finish (M60/145).

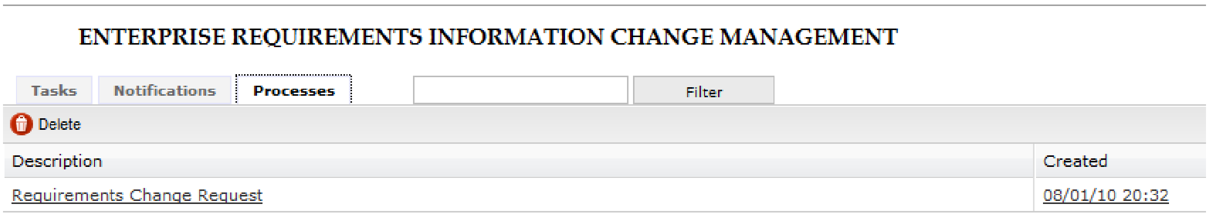
Figure 2: Traceability link between a requirement and a room data sheet

Web based forms were developed in the user interface to enable users to *add*, *view* and *update* information to any table. Password protection was used to control access. Each user is granted access rights depending on roles which determine how an individual interacts with the system and their responsibilities in the project.

3.3 *Change management system (CMS)*

The change management system was developed using Intalio Works BPMS. The development started by modelling the change request process using the case study described in this paper. The inbuilt process designer within the BPMS was used as the modelling tool which is based on BPMN. The process map described the activities, tasks, steps, resources, participants/actors (both humans and system) and information flow of the change request process. The process map was then orchestrated and transformed into an executable BPEL process. This transformation was achieved using Business Process Executable Language (BPEL) and implemented through the process engine/server.

Deployed business processes can be run through the user interface by completing web forms. In principle; users interact and perform tasks assigned to them through the user interface; access process notifications and can initiate a process. Three key requirements for an effective and efficient requirements change management are: the ability to perform traceability and checking for dependencies; updating requirements information after the approval process; and communicating that information as email notification to all those concerned. These requirements have been incorporated and implemented as features within the CMS. A user logs into the user interface and clicks on the *'processes'* link which opens a window displaying the *'Requirements Change Request'*. By clinking on this link, the user is presented with the change request form which is used to initiate a change request process. This form is filled and routed to the next activity in the process to be carried out either by a human or automatically. If the activity is a human workflow, the task will be listed under the *'task list'* when the person logs in. The process continues and before changes can be authorized, dependency of the proposed change has to be evaluated as a decision making task. This requires checking the database for all related requirements. As a result, the *'Dependency checker'*, which is a SQL operation service, was invoked and consumed by the *'check dependency'* activity of the change process. The *'dependency checker'* as a web service then sends a SOAP message based on the SQL query parameters to the database. The database is queried and the results are returned as SOAP response to the process. The process continues until a decision on the change is made. Following this, the relevant team members receive email notifications containing details of the changes. The *'send emails'* task of the change process invokes a built-in web service within the server and consumes the *'send'* operation to distribute email notifications. Simultaneously, the database is updated with the changes using the *'update database'* task of the process. This task invokes the *'insert'* web service which has been generated as a SQL command to insert data into a specific database table. At the same time, the change request initiator is also notified of the decision on the change on the process user interface. Figure 3 shows the change management system user interface with links *to tasks, notifications* and *processes*. Under the processes, the *'Requirements Change Request'* link which as user clicks to initiate a change request is also shown.



ENTERPRISE REQUIREMENTS INFORMATION CHANGE MANAGEMENT

Description	Created
Requirements Change Request	08/01/10 20:32

Figure 3: Change Management System User Interface

4 Discussion and Conclusion

The traditional approach to managing client requirements concentrates hugely on production of the initial brief. Once a design is produced, the brief is put to one side and all subsequent client wishes and changes are centered on the design. The original requirements information within the brief is rarely consulted or updated. Requirements are not centrally stored to facilitate shared access and as a result, duplicates exist which opens up the possibility of errors. Relationships and dependencies between requirements are not explicit. Client requirements change management is a laborious, manual process. It is also virtually impossible to effectively track the status of a request once it is in the change process through the manual procedure. The request would become invisible to the initiator. Business Process Management (BPM) has emerged as a technology that can be utilized by organisations to help coordinate and control the change process. Web services address the difficulties associated with checking for dependencies, updating change information and communicating the information to all stakeholders. The combination of SOA with web services and BPM can contribute towards integration and information sharing between disparate computer systems such as the web database and change management systems.

It is anticipated that when fully implemented the eRIM prototype will contribute immensely towards a better client requirements management in construction projects. The final stage of this research will be to evaluate the prototype through user tests and focus groups, to validate this assertion.

References

ERL, T., 2004. Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services. Upper Saddle River, NJ: Prentice Hall.

FERRARA, A. and MACDONALD, M., 2002. Programming .NET Web Services: Building Web Services with ASP.NET and C#. Sebastopol, California: O'Reilly.

GARIMELLA, K., Lees, M. & Williams, B. 2008. *BPM for Dummies,* Wiley Publishing, Inc. Indianapolis, Indiana.

JALLOW, A.K., DEMIAN, P., BALDWIN, A.N. and ANUMBA, C.J., 2010. Development of an Innovative Framework for Clients' Requirements Information Management in Construction Projects, In: *Proceedings of the 6th International Conference on Innovation in Architecture, Engineering and Construction (AEC),* 9-11 June 2010 (paper accepted).

JURIC, M.B., PANT K, 2008. Business Process Driven SOA using BPMN and BPEL: From Business Process Modeling to Orchestration and Service Oriented Architecture. Birmingham: Packt Publishing.

KIVINIEMI, A., 2005. *Requirements Management Interface to Building Product Models.* #161. Stanford University: CIFE.

ORACLE CORPORATION. 2008. *State of the Business Process Management Market*, Oracle Corporation, Redwood Shores, CA.

RITTINGHOUSE, J.W. and RANSOME, J.F., 2010. Cloud Computing: Implementation, Management and Security. Boca Raton, FL: CRC Press.

RUJIRAYANYONG, T. and SHI, J.J., 2006. A project-oriented data warehouse for construction. *Automation in Construction,* 15(6), 800-807.

SABRI, E.H., GUPTA, A.P. and BEITLER, M.A., 2007. Purchase Order Management Best Practices: Process, Technology, and Change Management. Fort Lauderdale: J. Ross Publishing.

TANRIKORUR, T. March 2007. *Business Process Management 101: The Basics of BPM and How to Choose the Right Suite*, CMP Media LLC, NY.

TIBCO. 2006. *The Case for Business Process Management*, TIBCO Software Inc., Palo Alto, CA.

WORTHINGTON, J., 2000. The Briefing and Design Process, *Overhead Material Seminar,* February 02 2000, Chalmers University of Technology.