Project-Oriented Software Configuration Management System

Chan Lee[1], Cheong Youn[2]

[1]*Department of Computer Engineering, Chungnam National University,*
Daejeon, South Korea
[2]*(Corresponding author) Department of Computer Engineering,*
Chungnam National University, Daejeon, South Korea

Abstract: The SCM (Software Configuration Management) is an activity, which configures the form of a system (documents, programs, and hardware) and systematically manages and controls modifications used to compile plans, development, and operations resulting from software development and maintenance. This process improves the quality of the software and is highly correlated with the development of reliable software. This paper presents the analysis, methods, and techniques of embodiment of the configuration management system design method, which is based on the systematic management of documents based on projects and activities. Since this system is designed based on the foundation of the web, it provides the user with high accessibility, maximizing the efficiency in carrying out tasks

Keywords: Configuration Management, Modification Management, Traceability, Version

I. INTRODUCTION

Like many other application fields, the integration of computer technology in the nuclear industry is imperative to the demands of the changing era. However, the primary reason why integration of computers in the nuclear power field was due to the difficulty in finding errors when using software to operate systems, compared to the use of hardware. Especially in the case of safety requiring systems, there has been significant resistance in the use of software. According to digital operating systems, issues in acquiring safety and reliability are being highly disputed. In order to solve these problems, research on quality assurance activities required in nuclear power plant software have been progressive. As verification and validation work as important measures in guaranteeing the quality of the software, part of the quality management process is able to systematically manage documents extracted as a result of advancement management.

When mass amounts of documents involving software development processes, such as nuclear power and/or various fields of study, are produced the systematic management of these documents is able to improve the quality of the software and is imperative to the development of reliable software. In such cases where documents are not properly managed, developers may have difficulty and experience confusion, thus causing considerable distress in future maintenance. In guaranteeing high quality in the software development phase and producing reliable products, it is important to control and govern documents.

Software quality management should be seriously valued in both the development phase as well as in the modification and maintenance phase. Even while operating the software, requests in modification continue to be received, so in order to confront these requests specific corresponding plans should be established. If modification requests are not properly processed in the software maintenance phase, this will result in deterioration in quality and declination in the life of the software. Especially in systems where safety is seriously valued, the chance of accidents due to the software may increase. Because of this, when recent research institutes and companies carry out projects, they are making attempts to automate systematic management of various documents containing information which will satisfy high quality and reliability.

This paper introduces the configuration management method and techniques of embodiment based on the systematic management of documents centered on projects in developing software for the nuclear power plant. This system is embodied based on the web foundation and supports distributed environment. The 2nd chapter provides an explanation for software configuration management, the 3rd chapter provides an explanation for related research about web application design model, the 4th chapter provides an explanation on the analysis and embodiment for software configuration management system centered on projects, and the 5th chapter states our conclusion.

II. SOFTWARE CONFIGURATION MANAGEMENT

The SCM is a process used to compile plans, development, and operation data resulting from software development and maintenance, which then configures the form of a system and systematically manages and controls modifications. Maintenance refers to the modification management made after the software has been delivered to the customer. Configuration activities, however, refer to the entire life cycle of the software, which thus includes the modification management of the software both before and after delivery.

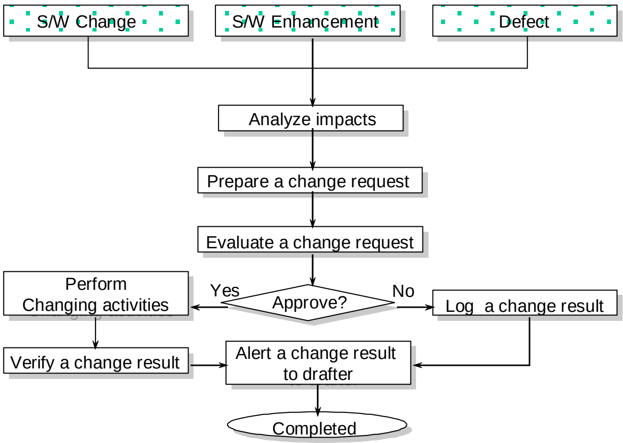


Figure 1. General modification process

IEEE decides the standards in "Software Configuration Management" [IEEE std 828-1990] and provides an explanation on items necessary for the planning process in making the SCM software. In addition, the configuration management activity is divided into four parts, Configuration Identification, Configuration Control, Configuration Status Accounting, and Configuration Audits and Reviews. As we can see, the various activities of SCM are related to the baseline. Since SCM includes the development process baselines presented at various levels of software engineering, such as demand item specification resulting from demand item analysis process, design specification resulting from design process, and original code resulting from the embodiment process, etc, are items that are managed by the SCM system.

Configuration Identification understands the software system's structure, uniquely identifies each of the composing points, and allows access in any form, thus throughout it's whole lifespan offers tracking related to software or software products. After analysis, Configuration Identification decides whether it will approval or disapprove modification requests on configuration. As management of direct modification of software works as a core function in Configuration Identification, it records accurately so that modification request evaluation and management procedures can be tracked down. Configuration Status Report consists of number of modifications in projects, versions of new software items, release identification personnel, number of releases, comparison of releases, etc, and is established as a criteria which not only preserves condition records of every item, but also provides a tracking system of all the modifications in criteria during the software's lifespan. Configuration Audits verify whether or not the software was made correctly according to demand items, standards, and contract matters. It also verifies whether or not the software product was produced with correct identification and manuals, and whether or not all modification requests were resolved[7].

After the Configuration Control analyzes the configuration of the modification request and evaluates it, decisions on weather the modification request will be approved or returned will be made. The Configuration Control's core function is to later on manage direct changes in the software.

III. WEB APPLICATION MODEL

Recently, since most software systems are compatible regardless of location and users are able to easily approach it, they are being developed based on the advantageous Web (World Wide Web). The software configuration management system for the nuclear power plant, as presented in this paper, was also designed and embodied using the web. From the views of the developer, this application was designed into three levels of View, Controller, and Model, and this division method is called the Model-View-Controller (MVC) model.

The View is the output section you see in the client in HTML, SML, or JAVA applet. The view only functions in showing information and is not concerned wit how the data was formed or where the data was retrieved. This is called the expression layer.

The Controller receives the input of the user from the view and hands it over to the model, which functions in sending the results of service provided by the model to the View. This division governing the application flow is called the control layer.

The Model, which executes tasks that are actually needed in application, is considered the core of the application. The model reads data from DB, and then uses it to perform tasks such as computation or execution. This is called the logic layer.

Web applications, when designed by using the MVC model, have the advantage of being made with solid and decisive structure.

IV. PROJECT CENTERED CONFIGURATION MANAGEMENT SYSTEM ANALYSIS AND DESIGN

The SCM system designed in this paper has been designed based on project. Project centered configuration management reflect upon present corporations' process, functions at each level, and the performance and the distinctive quality of corporation task of group system which allows unified management and operation of the organization of the corporation, project, and project related documents.

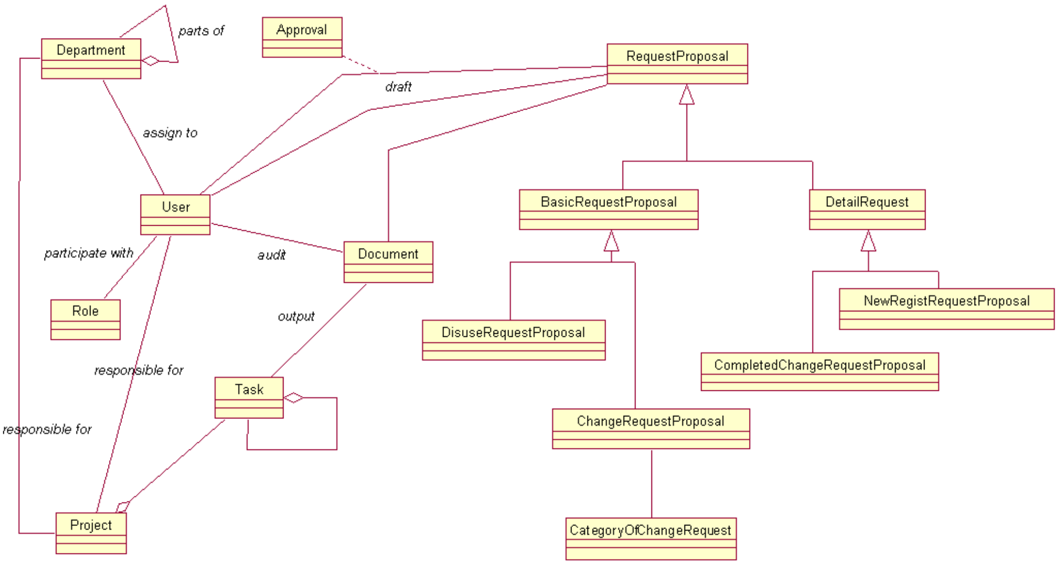
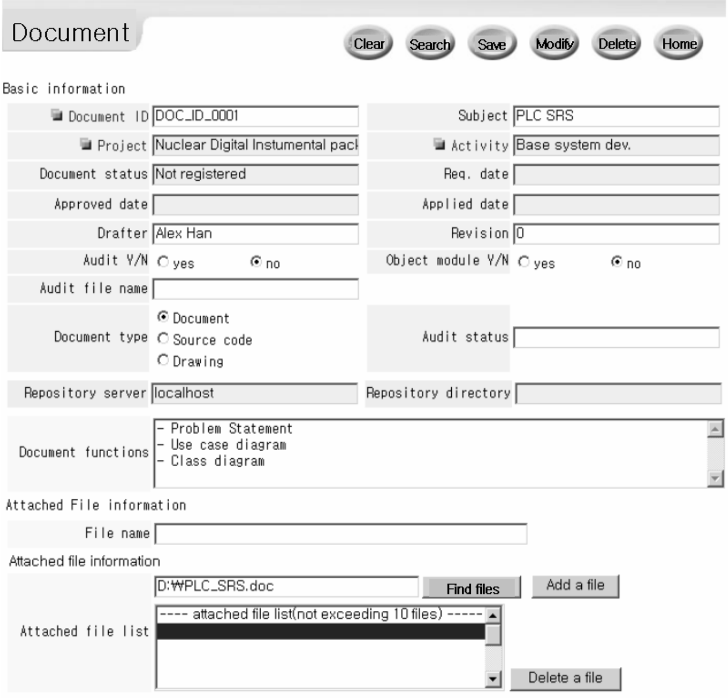


Figure 2. Class diagram

In addition, in order to efficiently manage project and documents, versions are used. The formation of versions is to be used as a kind of tool, which allows us to check the evident step-by-step development situation and function improvement of the project and documents. Generally, separate documents produced by the user exist in various activities within the project.

The class diagram of the system classified as UML is in Fig.2.

Project is a set of processes in which each process has a series of steps required in producing a software product. The project advances according to the process and the project process occurs in the order of the steps. Activities that should be executed in each step and results expected from each step are defined by the documents. Activities are once again consisted of documents yielded from lower activities and produced activities.



Document

Basic information

Document ID	DOC_ID_0001	Subject	PLC SRS
Project	Nuclear Digital Instumental pacl	Activity	Base system dev.
Document status	Not registered	Req. date	
Approved date		Applied date	
Drafter	Alex Han	Revision	0
Audit Y/N	○ yes ● no	Object module Y/N	○ yes ● no
Audit file name			
Document type	● Document ○ Source code ○ Drawing	Audit status	
Repository server	localhost	Repository directory	
Document functions	- Problem Statement - Use case diagram - Class diagram		

Attached File information

File name	

Attached file information

	D:₩PLC_SRS.doc
Attached file list	---- attached file list(not exceeding 10 files) -----

Figure 3. Document management screen

Activities are unit services, which are examples of various flow operations (tasks) composing project. Flow operation refers to the flow between activities. The flow between projects provides the base for efficient division of various tasks and levels shown in the project accomplishment process, which allows efficient and systematic project management.

In the SCM system for nuclear power operations, prototype concept level, requisite, design, realization, unification, inspection, final product concept, operation and maintenance, recognition steps can be considered as higher-level activity.

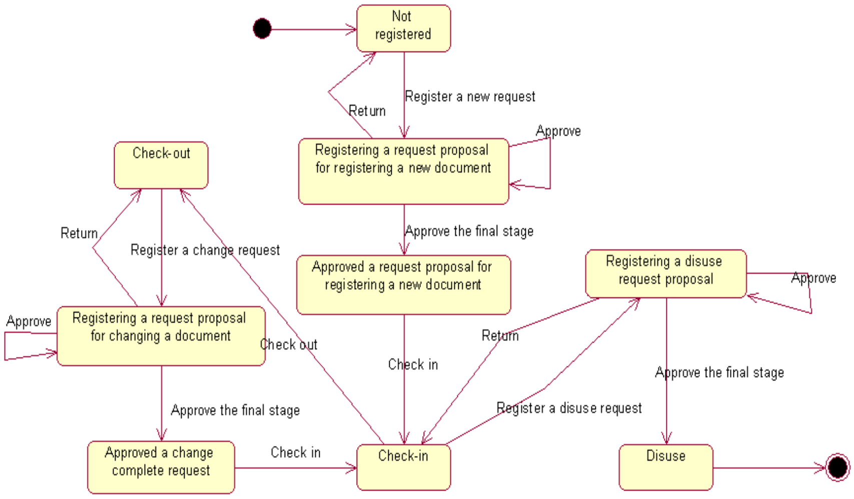
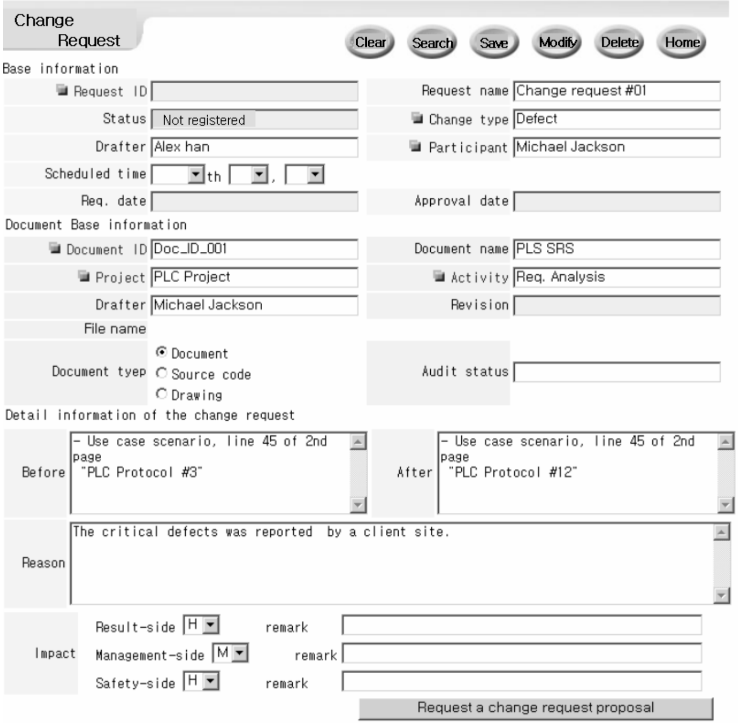

Not registered
Register a new request
Return
Approve
Registering a request proposal for registering a new document
Check-out
Return
Register a change request
Approve the final stage
Approved a request proposal for registering a new document
Registering a disuse request proposal
Approve
Approve
Registering a request proposal for changing a document
Check out
Return
Check in
Approve the final stage
Register a disuse request
Approve the final stage
Approved a change complete request
Check in
Check-in
Disuse

Figure 4. Condition changes in document

Documents from SCM must be essentially managed as the configuration item as shown in Fig.3, and as the project advances the basic units of task results in similar condition changes like that in Fig.4 in data on screen.

In SCM, the modification of the configuration item is inevitable. The request form needed when modifying a configuration item or making a new registration is a method, which allows the user with the document written with a specific goal to be approved and to be uploaded to the management server. The request form is the separated into a detailed request form and a standard request form depending on whether or not the information on the document needs modification or addition. A detailed request form is consisted of new registration request form and modification completion form, and a standard request form is consisted of discard request form and modification request form.

In the request form process, the approval makers will choose an approval line. According to the approval line, the request forms will automatically notify those in charge of the approval. These days, the definition and practice of approval line exists in an intensive parallel form, and this present system deals with the approval line that occurs continually. Fig.5 and Fig.6 show a definition screen for modification request draw up and the modification request draw up process, respectively.



Base information

Request ID		Request name	Change request #01
Status	Not registered	Change type	Defect
Drafter	Alex han	Participant	Michael Jackson
Scheduled time	th ,		
Req. date		Approval date	

Document Base information

Document ID	Doc_ID_001	Document name	PLS SRS
Project	PLC Project	Activity	Req. Analysis
Drafter	Michael Jackson	Revision	
File name			
Document tyep	● Document ○ Source code ○ Drawing	Audit status	

Detail information of the change request

Before: - Use case scenario, line 45 of 2nd page "PLC Protocol #3"

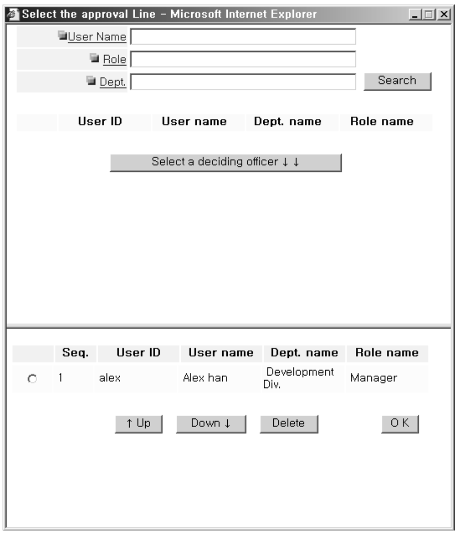
After: - Use case scenario, line 45 of 2nd page "PLC Protocol #12"

Reason: The critical defects was reported by a client site.

Impact

Result-side	H	remark	
Management-side	M	remark	
Safety-side	H	remark	

Figure 5.Modification request form


Select the approval Line - Microsoft Internet Explorer
User Name
Role
Dept.
Search
Select a deciding officer ↓ ↓
↑ Up
Down ↓
Delete
O K

User ID	User name	Dept. name	Role name

	Seq.	User ID	User name	Dept. name	Role name
○	1	alex	Alex han	Development Div.	Manager

Figure 6. Definition screen

This system is analyzed as previously explained and applies the MVC model in its design as seen in Fig.7.

View refers to the screen shown to the final user of the SCM system. The function of Business Logic is to manage the flow of SCM. For instance, for the SCM manager, it approaches models related to user and occupation information management, modify division, etc, and shows the suitable View. For the SCM user, it approaches models related to project, request form, write up, modification, and retrieval of documents and shows the suitable view. The Model represents the data used in SCM and manages tasks such as data creation, storage, and execution. Data Access Object works as an approach control function of the database. For example, it improves the efficiency of database management, such as assignment and cancellation whenever requesting connection while managing the connection pool.

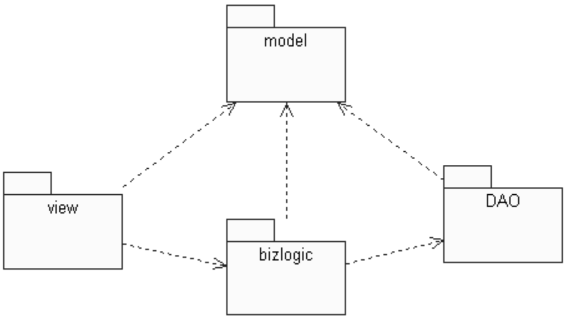


Figure 7. Design Structure

The class diagram for the document managed in the SCM system is shown in Fig.8. By designing with the MVC pattern, modules now have less affects on each other, which improves modification, expansion potential.

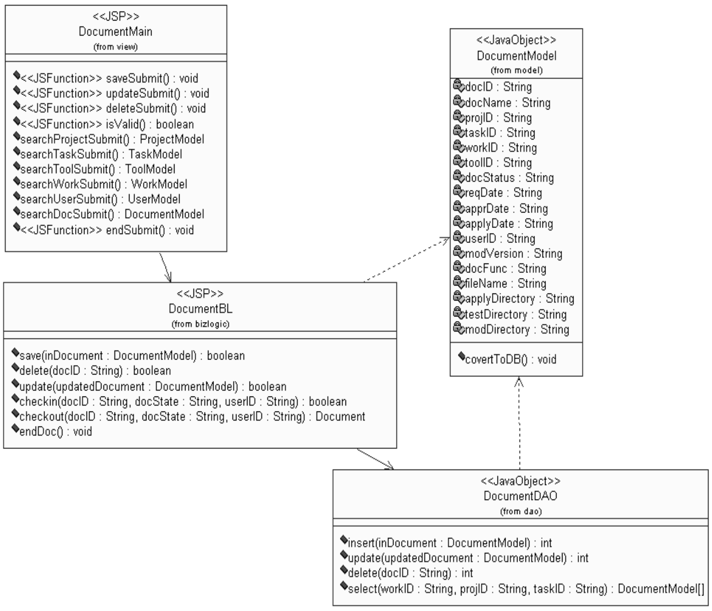

<<JSP>>
DocumentMain
(from view)
<<JSFunction>> saveSubmit() : void
<<JSFunction>> updateSubmit() : void
<<JSFunction>> deleteSubmit() : void
<<JSFunction>> isValid() : boolean
searchProjectSubmit() : ProjectModel
searchTaskSubmit() : TaskModel
searchToolSubmit() : ToolModel
searchWorkSubmit() : WorkModel
searchUserSubmit() : UserModel
searchDocSubmit() : DocumentModel
<<JSFunction>> endSubmit() : void
<<JavaObject>>
DocumentModel
(from model)
docID : String
docName : String
projID : String
taskID : String
workID : String
toolID : String
docStatus : String
reqDate : String
apprDate : String
applyDate : String
userID : String
modVersion : String
docFunc : String
fileName : String
applyDirectory : String
testDirectory : String
modDirectory : String
covertToDB() : void
<<JSP>>
DocumentBL
(from bizlogic)
save(inDocument : DocumentModel) : boolean
delete(docID : String) : boolean
update(updatedDocument : DocumentModel) : boolean
checkin(docID : String, docState : String, userID : String) : boolean
checkout(docID : String, docState : String, userID : String) : Document
endDoc() : void
<<JavaObject>>
DocumentDAO
(from dao)
insert(inDocument : DocumentModel) : int
update(updatedDocument : DocumentModel) : int
delete(docID : String) : int
select(workID : String, projID : String, taskID : String) : DocumentModel[]

Figure 8. Documents

V. PROJECT CENTERED EMBODIMENT OF CONFIGURATION MANAGEMENT SYSTEM

This system is a configuration management system centered on project, based on the developing software for nuclear power operation, which manages configuration management system based on project-oriented approach, which emphasizes the efficiency of document management and systematic task flow management. View and Control Stages were developed using a web application managementtechnician JSP (Java Server Pages), and Model Stage was organized using Java. A public software called Jakarta Tomcat 4.0.4 was used for the JSP application drive for web container.

For efficient data management, Microsoft's RDBMS, MS SQL Server 2000, was applied. Because java standard JDBC was used in database connection techniques, according to occasional demands of future clients porting of other RDBMS thru support of standard SQL is made possible.

VI. CONCLUSION

The SCM presented in this paper supports the web environment through the Internet which constructs the infra of present knowledge information sharing. Taken into fact that the typical company structure shares all information through the Internet, the imperative support of the use of the web in configuration management, modification management, and problem tracking is a basic and essential component. This type of web environment support provides a whole and global user work environment for the user, which maximizes the efficiency of work. In addition, by sufficiently reflecting the real time aspect in work, we can minimize resource waste and reduce unnecessary delay in the entire work time.

However, when documenting, the SCM system stated in this paper does not take parallel processing into consideration, which is when users simultaneously approach the same document in attempt to modify it. In contrast to other documents, numerous programmers must work together in developmental code documents so parallel processing is considered important. Further research on parallel processing is required for the development of a better SCM system.

REFERENCES

[1] Cheong Youn, *Software Engineering*, (Sangnung Publisher Co., 1999)

[2] IEEE Standards Board, *IEEE Standard for Software Configuration Management Plans*, IEEE Std 828-1990

[3] IEEE Standards Board, *IEEE Guide to Software Configuration Management*, ANSI/IEEE Std 1042-1987

[4] Donghoon Lee, Bumkyun Choi, *JSP Professional*, (Game Publisher Co., 2002)

[5] James J.Hunt, *Distributed Configuration Management via Java and the World Wide Web*, *ICSE SCM*, 1997

[6] Alexis Leon, *A Guide to Software Configuration Management*, (Artech House, 2000)

[7] Kiwon Jung, Changsup Youn, Taehyun Kim, *Software Quality and Process*, (Hongnumg Science Publisher Co., 1997)