Transactions of the Korean Nuclear Society Autumn Meeting
Gyeongju, Korea, November 2-3, 2006

Development of a Web-based CANDU Core Management Procedure Automation System

Sanghoon Lee, Eunggon Kim, Daeyou Park, Hyungbum Suh*, Sungmin Kim*, Choongsub Yeom
Institute for Advanced Engineering, Yongin P.O.Box 25, Gyeonggi-Do 449-863, shoon@iae.re.kr
** Wolsong Nuclear Power Division, Korea Hydro & Nuclear Power*

1. Introduction

CANDU reactor core needs efficient core management to increase safety, stability, high performance as well as to decrease operational cost.

The most characteristic feature of CANDU is so called "on-power refueling" i.e., there is no shutdown during refueling in opposition to that of PWR. Although this on-power refueling increases the efficiency of the plant, it requires heavy operational task and difficulties in real time operation such as regulating power distribution, burnup distribution, LZC statistics, the position of control devices and so on.

To enhance the CANDU core management, there are several approaches to help operator and reduce difficulties, one of them is the COMOS (CANDU Core On-line Monitoring System). It has developed as an on-line core surveillance system based on the standard in-cre instrumentation and the numerical analysis codes such as RFSP (Reactor Fueling Simulation Program) [1].

As the procedure is getting more complex and the number of programs is increased, it is required that integrated and cooperative system. So, KHNP and IAE have been developing a new web-based system which can support effective and accurate reactor operational environment called COMPAS that means CANDU cOre Management Procedure Automation System.

To ensure development of successful system, several steps of identifying requirements have been performed and Software Requirement Specification (SRS) document was developed [2]. In this paper we emphasis on the how to keep consistency between the requirements and system products by applying requirement traceability methodology [3].

2. Development of the System

We started analysis and gathering requirements of the target system using object oriented programming (OOP) methodology which is widely popular in large-scale software engineering. Furthermore, OOP is easier to learn and the OOP approach is often simpler to develop and to maintain. We have several steps to accomplish the system by using concept of OOP and each step has its output or product as illustrated in Table 1.

Table 1 Development Step of the System

Step	Activities	Products
Analysis	Define requirements	SRS, Usecase
Design	Define Software & Hardware Architecture	Class Diagram, Deployment model
	Data Modeling	DB shcema
Implement ation	Coding Testing	Applications
Test	Verify the acceptability of the system	Target Systems

2.1 Analysis

Requirements are identified by interviewing members who have responsibility of the reactor management procedures and then wrote SRS document using IEEE 830 template at the beginning of the development [2]. Table 2 shows the skeleton of the document.

Table 2. Skeleton of SRS

Chap 1. Introduction	1.1 Purpose 1.2 Scope 1.3 Definitions, Acronyms and Abbreviations 1.4 References 1.5 Overview
Chap. 2 General description	2.1 System Interface and Operation 2.2 System Function 2.3 System Architecture 2.4 Constraints 2.5 Assumptions and Dependencies
Chap. 3. Specific Requirements	3.1 Fuel Management 3.2 Reactor Safety Management 3.3 Instrument Management 3.4 Supporting Functions
Chap. 4 Performance and security	4.1 System Performance 4.2 Reliability 4.3 Security
Chap. 5 Glossary and Abbreviation	
Chap.6 Appendices	

After making document, we have several meetings and discussion on the requirements to confirm the scope of the system and validate if there were some missing items or not. This kind of process is called configuration management which is also playing important role in software development process and has close relationship with requirement traceability.

2.2 Design

Effective and accurate design is main key to success the development. In the view of OOP, the most important design is class diagram. The system is composed of the several components and effective interactions between the components are very important to make a good system. The class diagram of the system is illustrated in Figure 1.

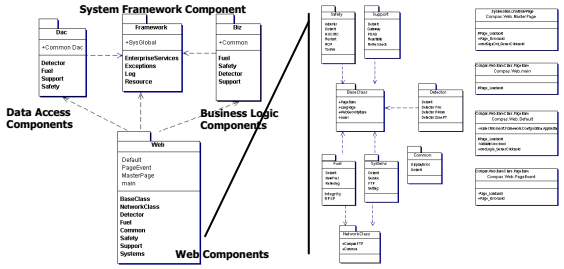

Figure 1. Class Diagram of the System

To implement class design, it is needed to determine software and hardware architecture. These architectures make a role of blue print of the system and define how to interact the components, where to deploy the components and what is the most effective hardware configuration. The software and hardware architecture of the systems are illustrated in Figure 2 and Figure 3 respectively.

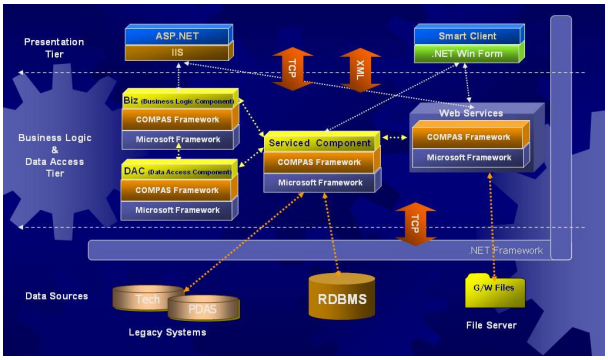

Figure 2. Software Architecture of the System

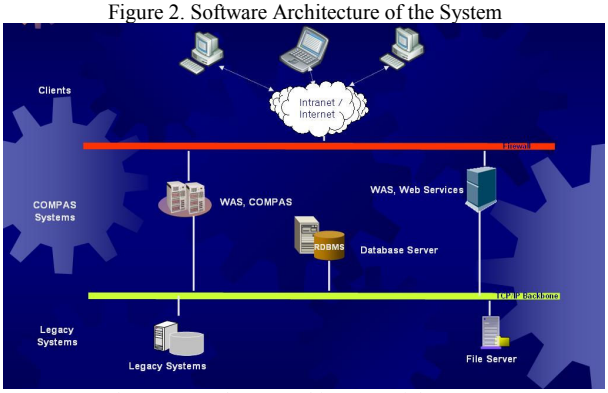

Figure 3. Hardware Architecture of the System

2.3 Implementation

Implementation is making each component mentioned in the class diagram and requirement spec. as well as presenting idea in the possible way. Although there are many tools and environment to realize it, we choose MS .NET framework, ASP.NET and C# to increase integration of legacy systems and performance of development. As shown in the software architecture, main components are implemented by C# and presentation materials are implemented by ASP.NET. Integration of the each system is done by web services, XML and so on.

In addition to ASP.NET, to provide flexible and plentiful user interface, smart client technology is adopted and it makes possible to complex user interface in the web-based system. The examples of web page and smart client are shown in the Figure 4 and Figure 5.

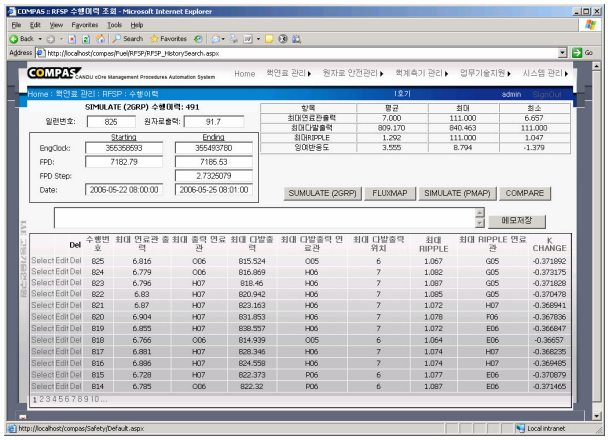

Figure 4. Example of web page

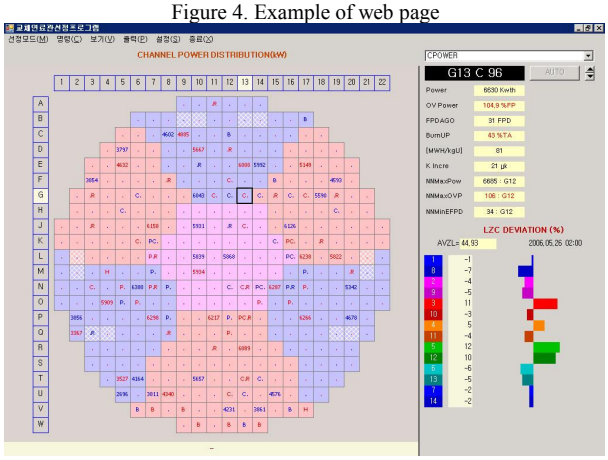

Figure 5. Example of smart client

3. Conclusion

In this paper, we show how to develop a web-based CANDU core management system in the view of OOP methodology. Several steps were applied successfully, such as analysis, design, and implementation.

To make successful system, there are essential points to be considered, such as how to gather requirements, to determine which method will be the best with respect to who implements the system.

REFERENCES

[1] Daeyu Park, Sanghoon Lee, Choongsub Yeom, Yunho Kim, Yongbae Kim, Seongmin Kim, Development of CANDU Core Operation Support System Based on Core Monitoring System, KNS Conference, Spring, 2005

[2] Sanghoon Lee, Daeyou Park, Hyungbum Suh, Sungmin Kim, Choongsub Yeom, A Case Study on the Development Process of the Software Requirement Specification of the COMPAS, KNS Conference, Fall, 2005

[3] Sanghoon Lee, Daeyou Park, Hyungbum Suh, Sungmin Kim, Choongsub Yeom, Study on Requirement Traceability in Development of CANDU Core Management Procedure Automation System, KNS Conference, Spring, 2006