Development of a Web-Based CANDU Core Management Procedures Automation System

Sanghoon Lee[1], Daeyu Park[1], Choongsub Yeom[1] and Hyungbum Suh[2]
[1] Institute for Advanced Engineering (IAE), Yongin, Korea
[2] Korea Hydro & Nuclear Power (KHNP), Wolsong, Korea

Abstract

Introduce CANDU core management procedures automation system (COMPAS) – A web-based application which semi-automates several CANDU core management tasks. It provides various functionalities including selection and evaluation of refueling channel, detector calibration, coolant flow estimation and thermal power calculation through automated interfacing with analysis codes (RFSP, NUCIRC, etc.) and plant data. It also utilizes brand new *.NET* computing technology such as *ASP.NET, smart client, web services* and so on. Since almost all functions are abstracted from the previous experiences of the current working members of the *Wolsong Nuclear Power Plant* (NPP), it will lead to an efficient and safe operation of CANDU plants.

1. Introduction

A CANDU plant needs efficient core management to increase safety, stability, and high performance as well as to decrease operational cost as like as other type of nuclear power plants.
Besides of above similarity, 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 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 system called *COMOS* (CANDU core On-line Monitoring System) and developed by IAE in 2005. It provides an on-line core surveillance system based on the standard in-core instrumentation and numerical code such as RFSP (Reactor Fueling Simulation Program) [1].

As management procedures are getting more complex and the number of programs is increased, it is required that integrated and collaborative system to get high performance and accuracy of operation as well as prevent human error. In addition to, historical knowledge and experience of seniors should be transferred to junior members seamlessly. So, KHNP and IAE have developed a new web-based system which can support effective and accurate reactor operational environment called COMPAS that is abbreviated of CANDU cOre Management Procedures Automation System.

To ensure successful development of system, we adopt progressive elaboration approach from identifying to tracing user requirement as detail as possible. At first, we prepared *Software Requirement Specification* (SRS) document as a result of discussion and interview between stakeholders then we applied requirement traceability methodology to keep consistency between requirements and products [2][3].

We also considered how to integrate and upgrade from the legacy system of Wolsong NPP, because new system should support error-free integration and migration as well as ensuring high user acceptance. To meet these requirements, we chose *Microsoft .Net framework* environment. Through the .Net technology, it makes possible to build more user-friendly interfaces and manageable web application.

2. Development of the System

2.1 Scope of the System

The target system is designed to support semi-automated CANDU core management of Wolsong NPP unit 1 & 2. The definition of *semi-automated* in the paper is that it performs designed task automatically but it required user intervention at every decision to confirm. The system integrates data from various source such as Digital Control Computer (DCC), human input, Plant Data Acquisition System (PDAS) and so on and processes them using computing algorithm based on plant operational procedures. Consequently, the processed data will be displayed in the web browser of users.

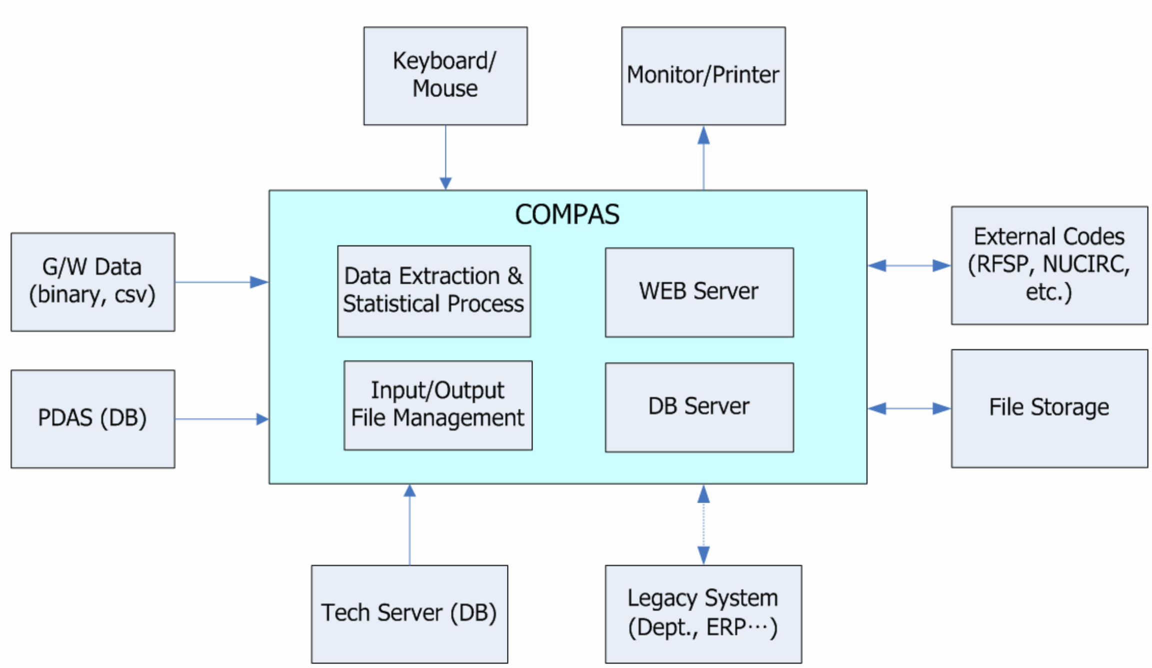


Figure 1 Scope and Interface of the System

The starting point of design was identifying the scope of systems and their interfaces. As the result, the scope of system is identified as above:

To clarify the scope, it is useful to summarize into table with respect to functional requirements, input, output and tracking code as Table 1.

Function Name	Functional Requirements	Input	Output	Tracking Code
RFSP Run	Reactor analysis Manage run history	DAF, G/W data files	Output files Running transaction	SRS-3.1.1 Sul 0-8-205
Refueling	Select refueling channel Manage fueling history	G/W, RFSP	Refueling request	SRS-3.1.2 Sul 0-8-212
Fuel Inventory Management	New fuel inventory management Taking over, Shipping, Storing & Stock	Printed documents	Taking over test result, Shipping confirmation	SRS-3.1.3 Sul 0-1-211
Fuel Integrity Check	Detect & Analysis of defected fuel Check GFP system	G/W data, DN Scan result	DN operation results DN analysis chart GFP check chart	SRS-3.1.4 Sul 0-8-209
ROP Calibration	Calculate ROP detector calibration factor	G/W, RFSP, Poison density	Nominal DC	SRS-3.2.1 Sul 0-8-207
Check Coolant Flow Rate	Calculate CTM System RTD Bias Estimate design coolant flow	G/W data PDAS data	RTD bias Design coolant flow	SRS-3.2.2 Sul 0-8-210
Check Thermal Power	Monitoring and Calibration of Thermal Power	G/W	Results and reports	SRS-3.2.3 Sul 0-8-201
Low Power and Restart	Check reactor status and reactivity under low power and restart	G/W, RFSP, Reactivity calculation code	Calculated reactivity Reactivity balance	SRS-3.2.4 Sul 0-8-204
Adjuster Withdrawal	Monitoring transient reactor	G/W, RFSP	ROPT cal. factor Power limit	SRS-3.2.5 Sul 0-8-206
Pt. Detector Calibration	Determine Pt. detector calibration factor	G/W, user input	Request value and results report	SRS-3.3.1 Sul 0-8-203
Nominal Neutron Flux	Estimation of nominal neutron flux difference	G/W, user input	Request change value Results and reports	SRS-3.3.2 Sul 0-8-208
Vd. Detector Calibration	Calculation S-factor of Vd. detector	G/W, RFSP, user input	S-factor, Kj-factor	SRS-3.3.3 Sul 0-8-202
Reactivity	Calculate reactivity	G/W, user input	Reactivity	SRS-3.4.1
Check Refueling	Check refueling status			SRS-3.4.2
G/W Data Processing	DCC data processing	DCC outputs	DCC values	SRS-3.4.3

Table 1 Scope of Reactor Management Procedures

Once the scope of system is determined, we can design and specify functionality and work breakdown structures of our target system. As shown in the Figure 2, it is comprised of major four parts, i.e., Nuclear Fuel Management, Reactor Safety Management, Detector Management and Technical Support. All of them are very important and critical components of the system.

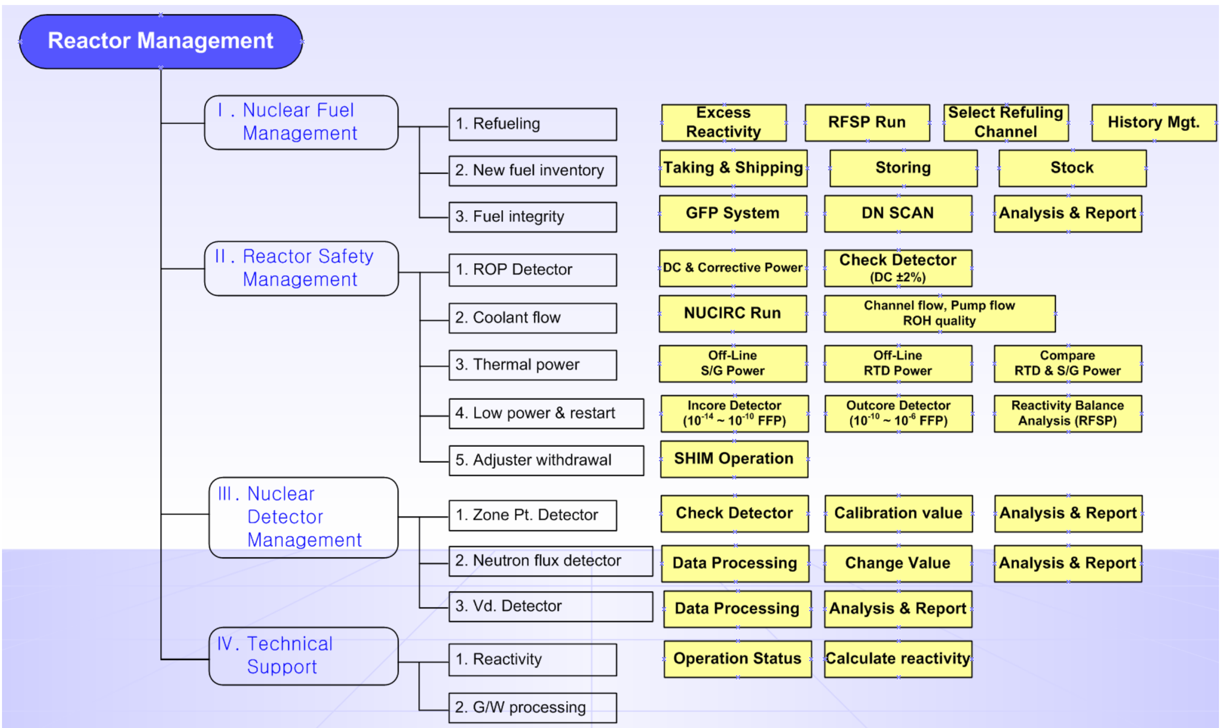


Figure 2 Scope and Interface of the System

2.2 Analysis and Writing SRS

In the beginning of development, requirements were identified by interviewing members who have responsibility of the reactor management procedures and then SRS document was written by using IEEE 830 template [2]. Table 2 shows the skeleton of the document.

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 Nuclear Fuel Management 3.2 Reactor Safety Management 3.3 Detector Management 3.4 Technical Support Functions
Chap. 4. Performance and security	4.1 System Performance 4.2 Reliability 4.3 Security
Chap. 5. Glossary and Abbreviation	
Chap. 6. Appendices	

Table 2 Skeleton of SRS

After making document, we have had 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 an important role in the software development process and has close relationship with requirement traceability.

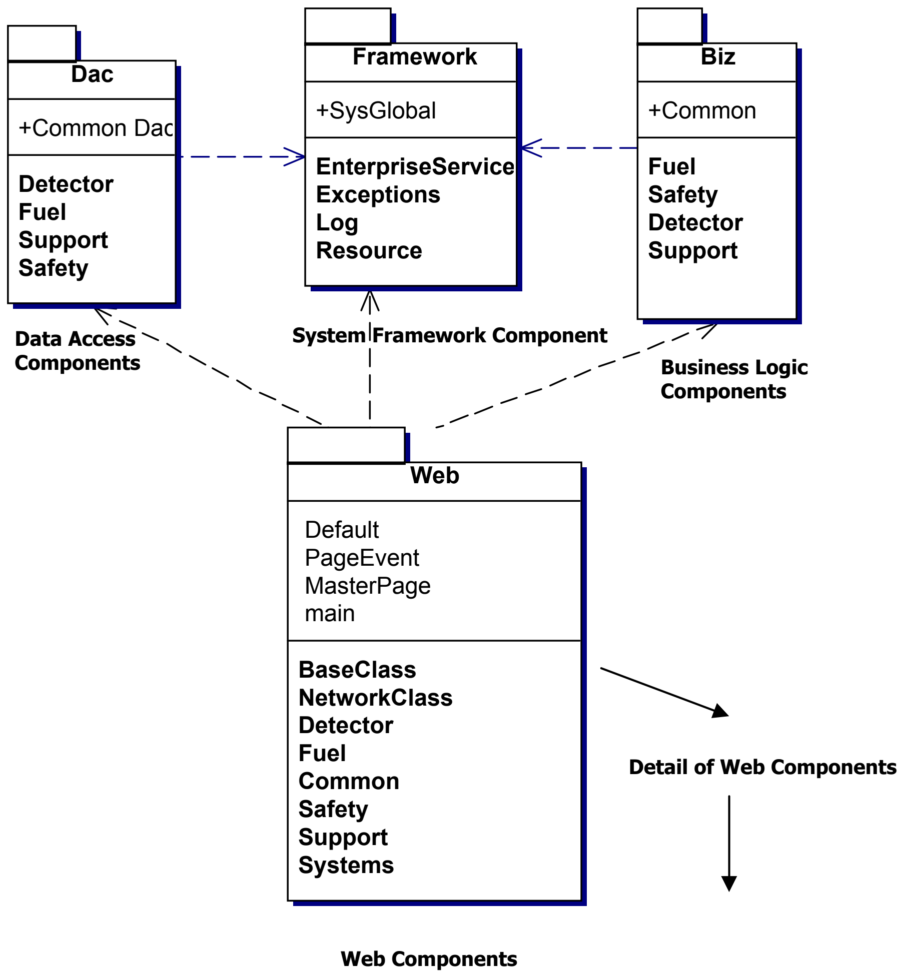
2.3 Object Oriented Design

We adopted *Object Oriented Programming* (OOP) methodology to analyze and retrieve requirements of the target system. OOP is widely popular in large-scale software development, easier to learn and is often simpler to develop and to maintain. To accomplish required purpose, you have to define activities for each step i.e., analysis, design, implementation, and test. To close each step, you have to complete planned deliverables (output or product) as illustrated in Table 3.

Step	Activities	Deliverables
Analysis	Define requirements	SRS, Use case
Design	Define Software & Hardware Architecture Data Modeling	Class diagram, Deployment model DB modeling
Implementation	Coding Testing	Applications
Test	Verify the acceptability of the system	Target system

Table 3 Main Activities and Deliverables

The primary success point of information system is how to design it effectively and systemize requirements. In the view of OOP, the most important design is class diagram. Since the system is composed of the several components, effective interactions between the classes are very important to make a good system. The class diagram of the system is illustrated in Figure 3.



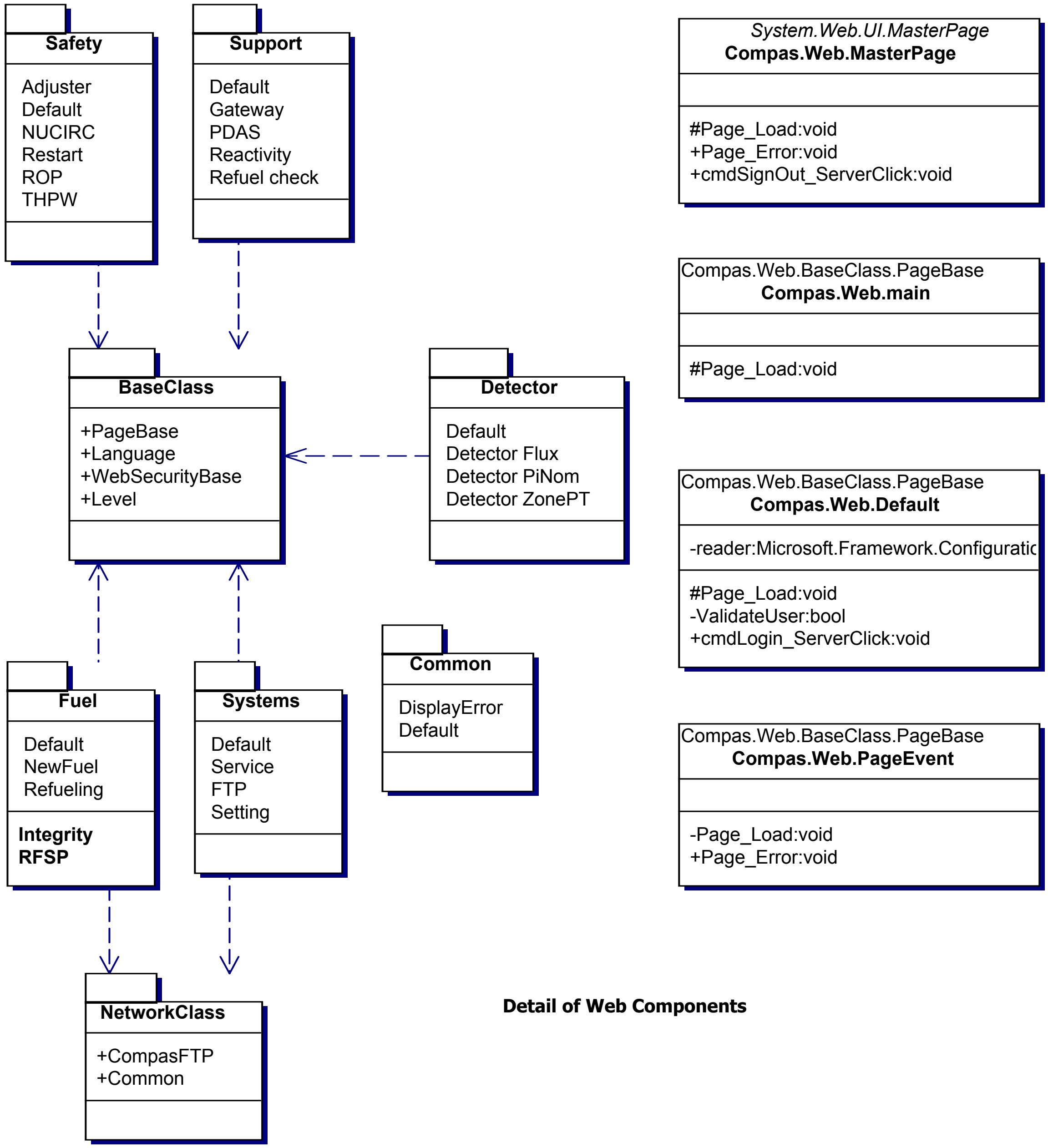


Figure 3 Class Diagram of the System

To implement the design into real application, it is needed to determine and consider software and hardware architecture. These architectures make an important role as a blue print of the system. Software architecture shows interaction between element components, whereas hardware architecture is used to determine where to deploy the components and what is the most effective hardware configuration. The software and hardware architecture of the system are illustrated in Figure 4 and Figure 5 respectively.

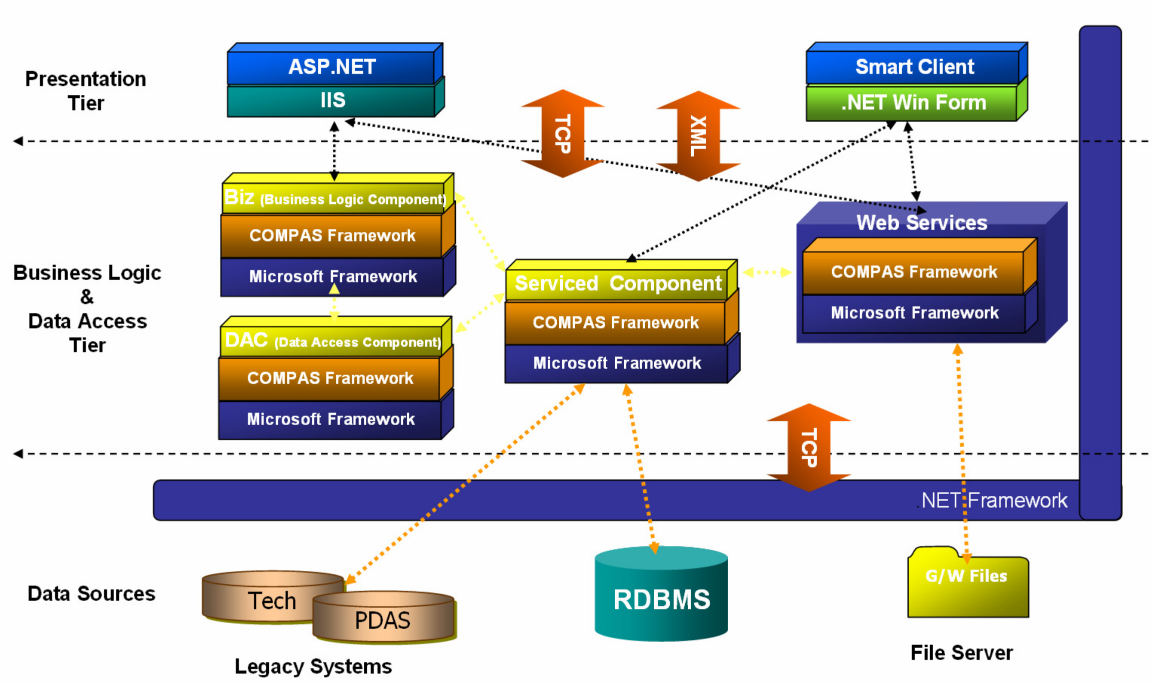


Figure 4 Software Architecture of the System

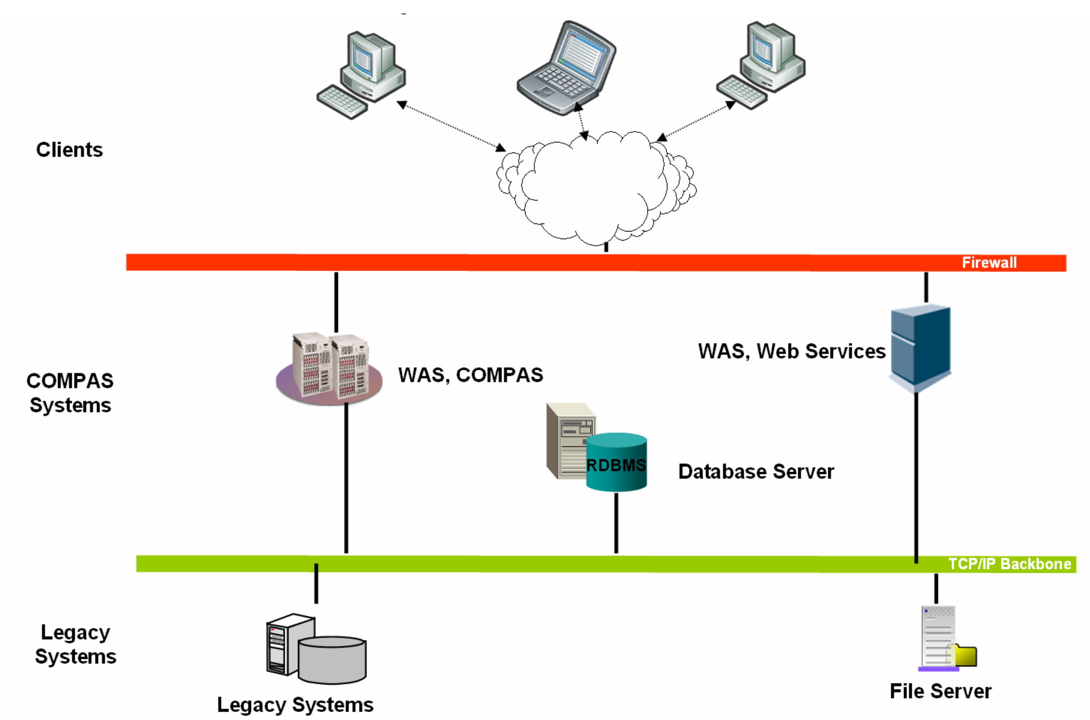


Figure 5 Hardware Architecture of the System

2.4 Implementation of the System

An implementation is making components mentioned in the design and requirement specification to present idea in the possible and tangible way. Among the many tools and environments to realize it, we chose *Microsoft .NET framework, ASP.NET* and *C#* to increase integration level of legacy systems and performance of development. As shown in the software architecture, main back-end components are implemented by C# and front-end presentations are implemented by ASP.NET. Integration of the each system is done by web services using XML and dataset.

On the contrary of previous approach to import and use DCC data, we adopted *web services* to use the data as a universal data source around all applications and platform independently. The web services convert raw binary file data into XML dataset and work as a provider as like as a remote web component [4].

In addition to ASP.NET, to provide flexible and plentiful user interface, *smart client* technology is adopted and makes it possible to enable complex user interface in the web-based application. The examples of web page and smart client are shown in the following figures.

As stated in the system scope statements, COMPAS has four major components. To show the results of each component implementation, a few screens of web browser and smart client are captured and if needed, an explanation is added.

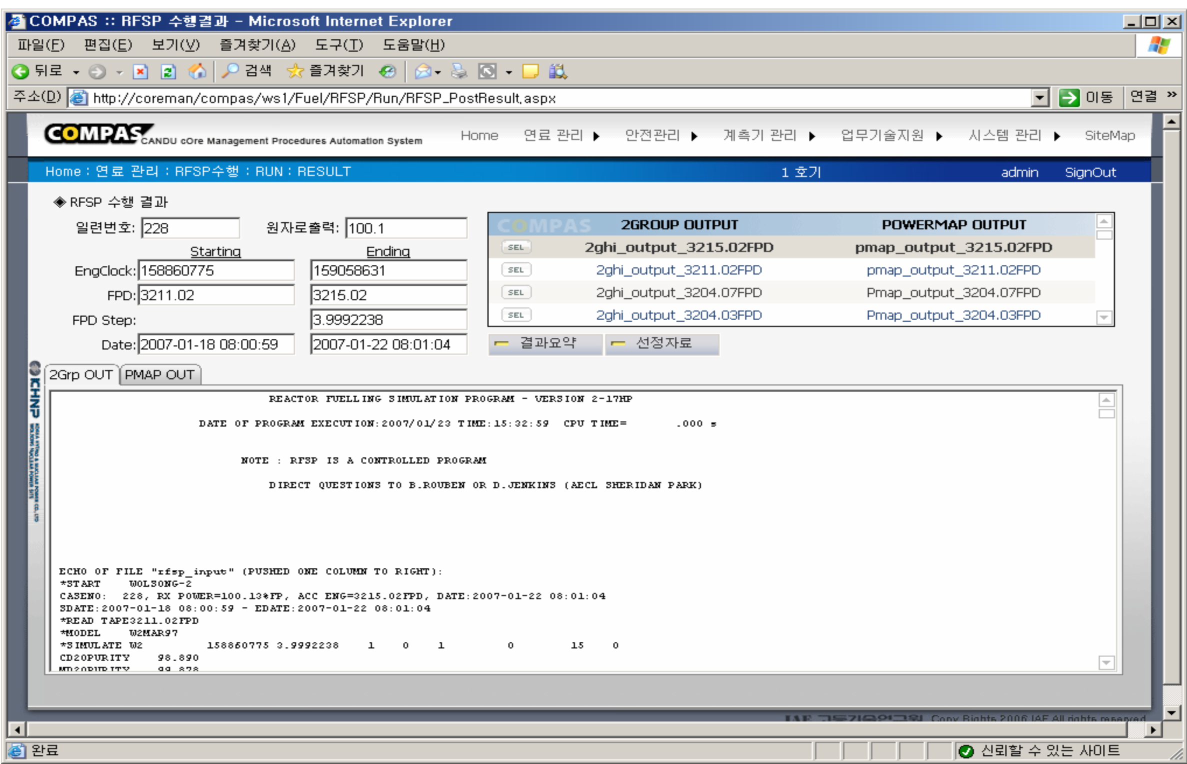


Figure 6 RFSP Code Interface

The RFSP is one of the most important codes in managing CANDU core and it requires to be executed efficiently and accurately in any case. It shows the result of code execution in the Figure 6 and this result is stored into database and then used in the smart client version of channel selection support interface as in the Figure 7. It is comprised of channel map with selected channel (grayed square) and its detail properties (upper right) and derived LZC deviation as the consequence of the selection (lower right). Using this intuitive information, operator can easily select refueling channel. It is also useful to show all channel information including detector position in one screen as Figure 8.

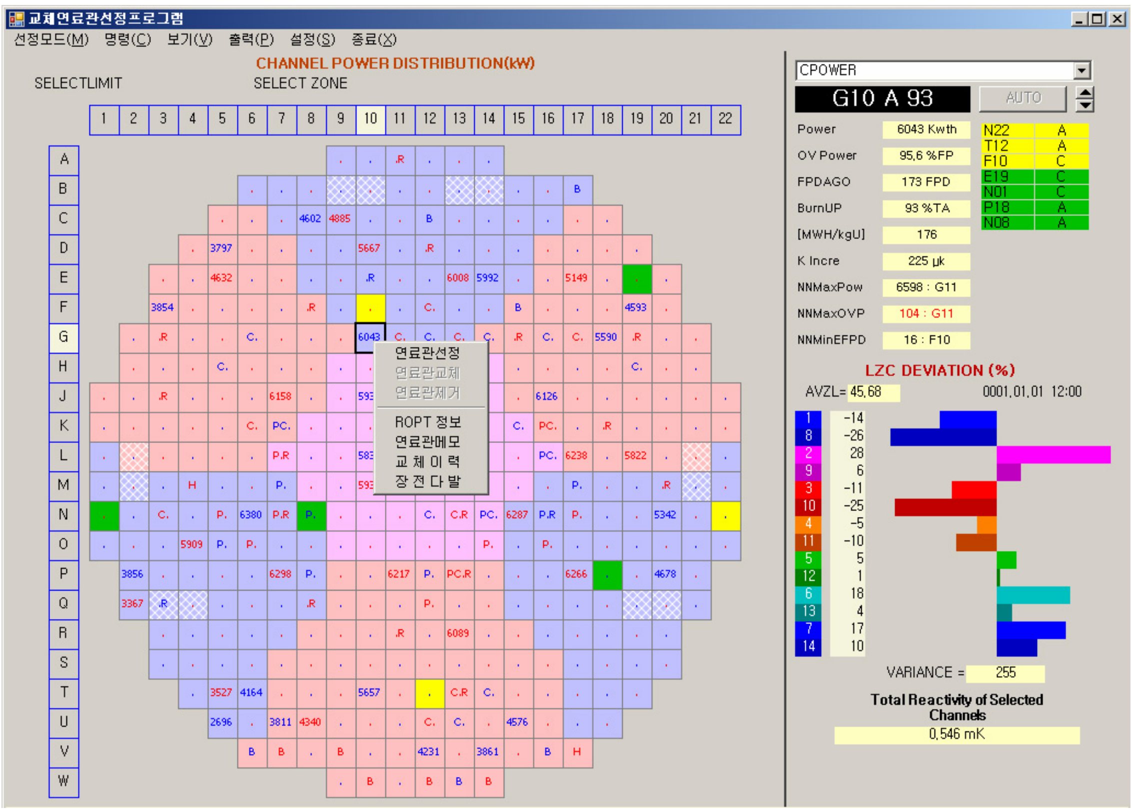


Figure 7 Refueling Channel Selection Interface

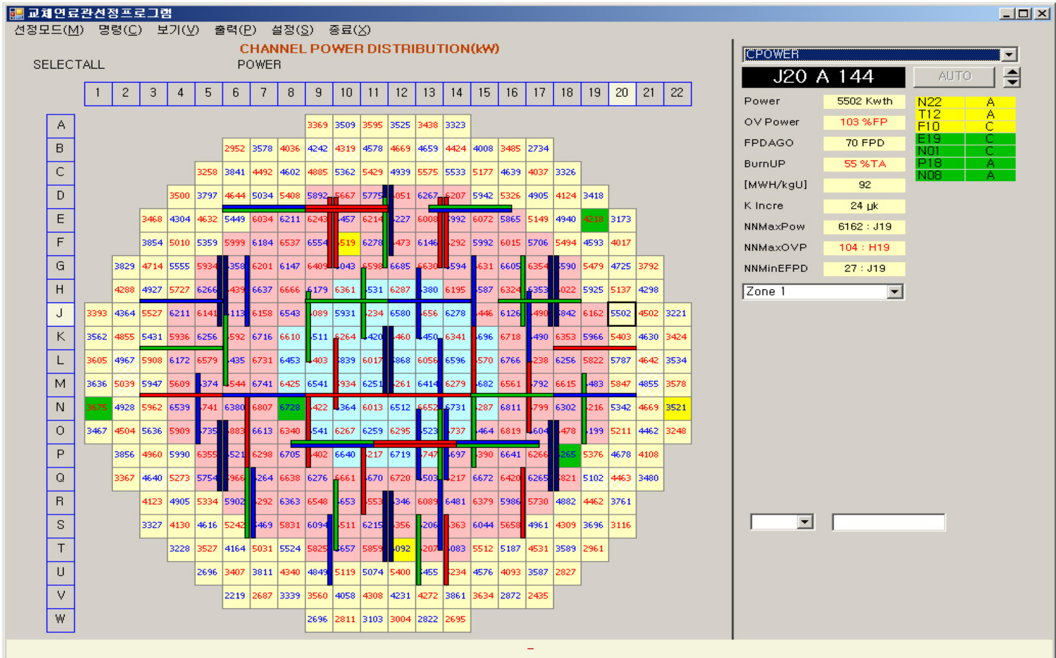


Figure 8 CANDU Channel Map with Detectors

In the view of CANDU safety, it is important to know current thermal power and coolant flow rate. To meet these requirements, COMPAS provides coolant flow estimation using NUCIRC calculation (Figure 9), and calculate thermal power (Figure 10) and ROP detector calibration factor (Figure 11).

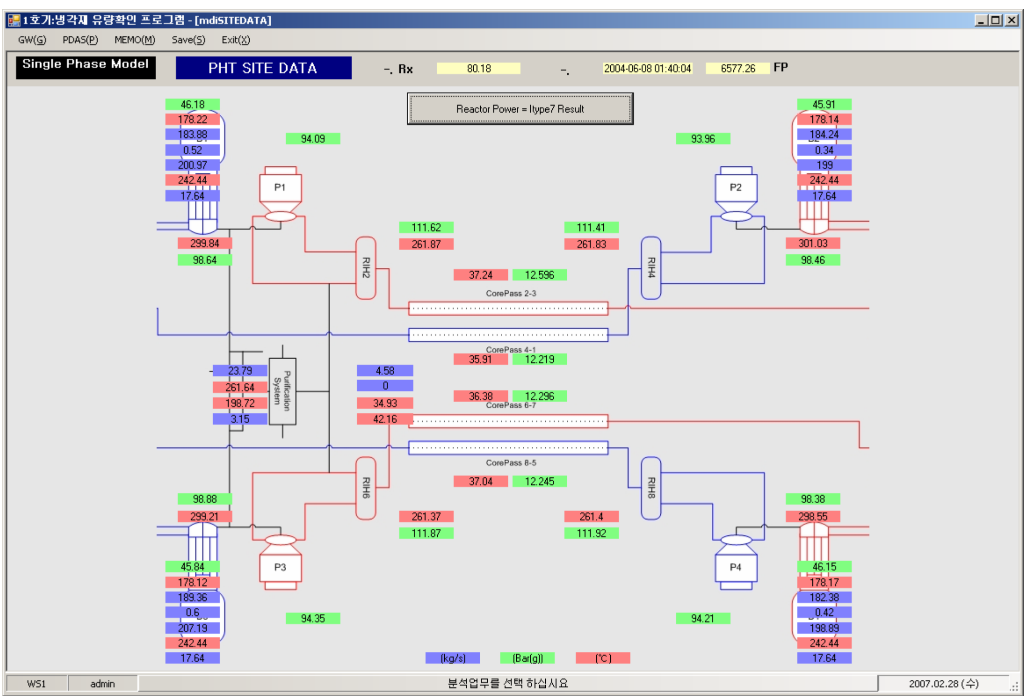
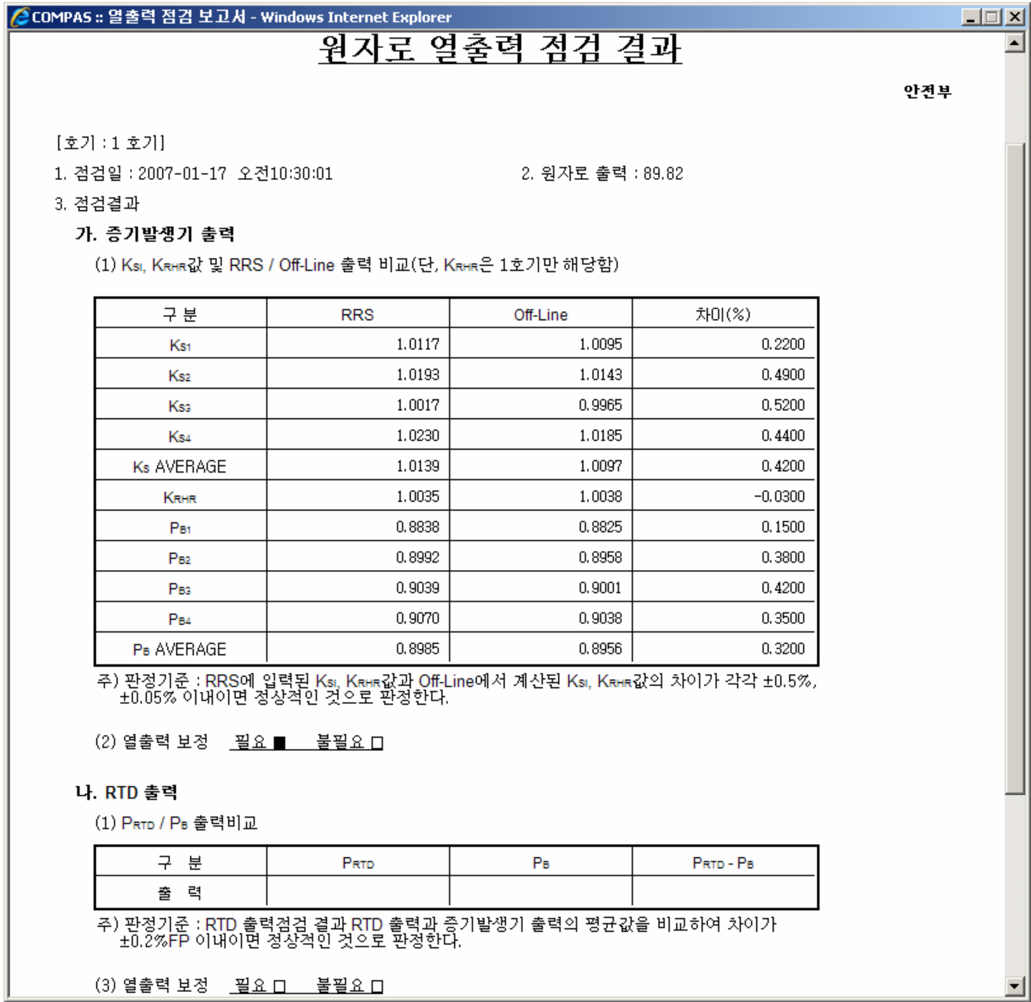


Figure 9 Estimation of Coolant Flow Rate Using NUCIRC



COMPAS :: 열출력 점검 보고서 - Windows Internet Explorer

원자로 열출력 점검 결과

안전부

[호기 : 1 호기]

1. 점검일 : 2007-01-17 오전10:30:01   2. 원자로 출력 : 89.82

3. 점검결과

가. 증기발생기 출력

(1) K_{SI}, K_{RHR}값 및 RRS / Off-Line 출력 비교(단, K_{RHR}은 1호기만 해당함)

구 분	RRS	Off-Line	차이(%)
K_{S1}	1.0117	1.0095	0.2200
K_{S2}	1.0193	1.0143	0.4900
K_{S3}	1.0017	0.9965	0.5200
K_{S4}	1.0230	1.0185	0.4400
K_S AVERAGE	1.0139	1.0097	0.4200
K_{RHR}	1.0035	1.0038	-0.0300
P_{B1}	0.8838	0.8825	0.1500
P_{B2}	0.8992	0.8958	0.3800
P_{B3}	0.9039	0.9001	0.4200
P_{B4}	0.9070	0.9038	0.3500
P_B AVERAGE	0.8985	0.8956	0.3200

주) 판정기준 : RRS에 입력된 K_{SI}, K_{RHR}값과 Off-Line에서 계산된 K_{SI}, K_{RHR}값의 차이가 각각 ±0.5%, ±0.05% 이내이면 정상적인 것으로 판정한다.

(2) 열출력 보정 필요 ■ 불필요 □

나. RTD 출력

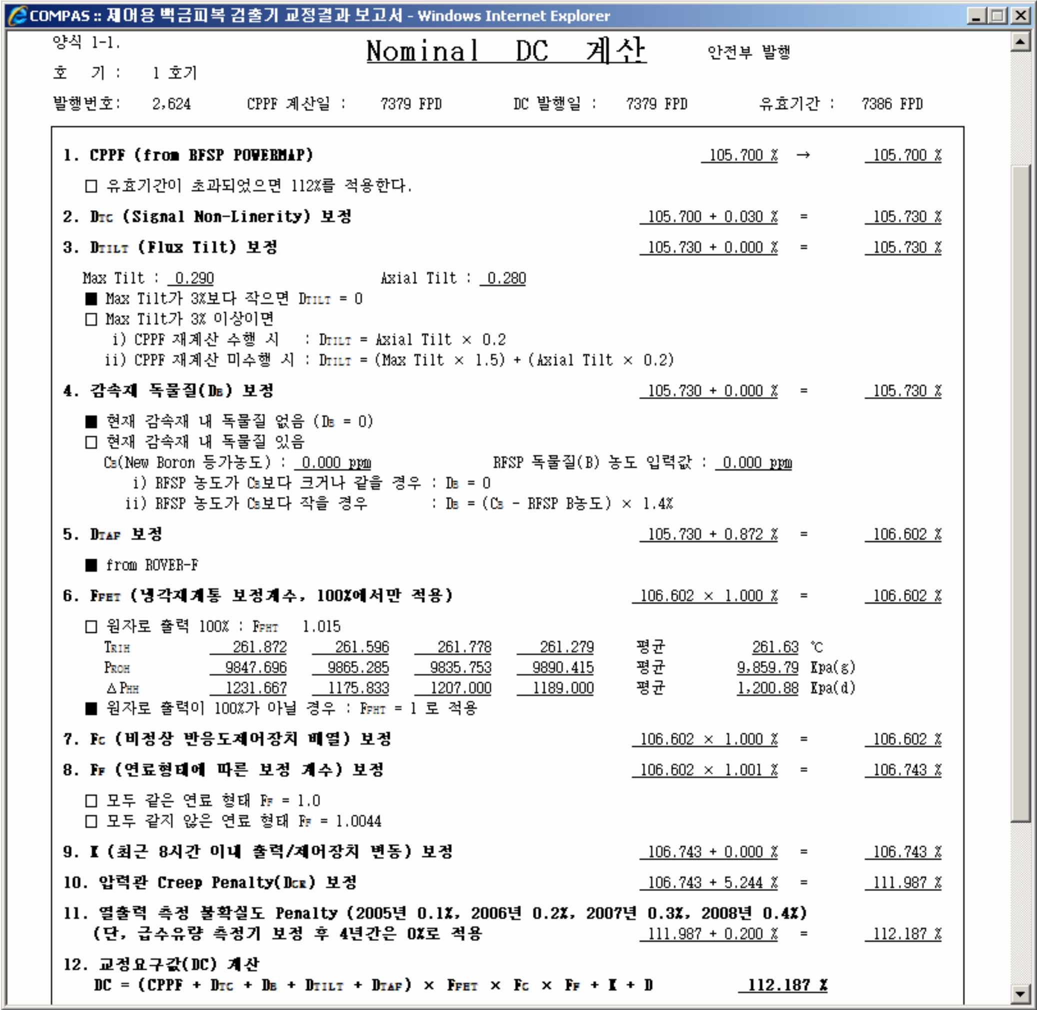
(1) P_{RTD} / P_B 출력비교

구 분	P_{RTD}	P_B	P_{RTD} - P_B
출 력			

주) 판정기준 : RTD 출력점검 결과 RTD 출력과 증기발생기 출력의 평균값을 비교하여 차이가 ±0.2%FP 이내이면 정상적인 것으로 판정한다.

(3) 열출력 보정 필요 □ 불필요 □

Figure 10 Thermal Power Calculation



COMPAS :: 제어용 백금피복 검출기 교정결과 보고서 - Windows Internet Explorer

양식 1-1.

Nominal DC 계산 안전부 발행

호 기 : 1 호기

발행번호: 2,624 CPPF 계산일 : 7379 FPD DC 발행일 : 7379 FPD 유효기간 : 7386 FPD

1. CPPF (from RFSP POWERMAP) 105.700 % → 105.700 %

☐ 유효기간이 초과되었으면 112%를 적용한다.

2. D_{TC} (Signal Non-Linerity) 보정 105.700 + 0.030 % = 105.730 %

3. D_{TILT} (Flux Tilt) 보정 105.730 + 0.000 % = 105.730 %

Max Tilt : 0.290 Axial Tilt : 0.280
■ Max Tilt가 3%보다 작으면 D_{TILT} = 0
☐ Max Tilt가 3% 이상이면
i) CPPF 재계산 수행 시 : D_{TILT} = Axial Tilt × 0.2
ii) CPPF 재계산 미수행 시 : D_{TILT} = (Max Tilt × 1.5) + (Axial Tilt × 0.2)

4. 감속재 독물질(D_B) 보정 105.730 + 0.000 % = 105.730 %

■ 현재 감속재 내 독물질 없음 (D_B = 0)
☐ 현재 감속재 내 독물질 있음
C_B(New Boron 등가농도) : 0.000 ppm RFSP 독물질(B) 농도 입력값 : 0.000 ppm
i) RFSP 농도가 C_B보다 크거나 같을 경우 : D_B = 0
ii) RFSP 농도가 C_B보다 작을 경우 : D_B = (C_B - RFSP B농도) × 1.4%

5. D_{TAF} 보정 105.730 + 0.872 % = 106.602 %

■ from ROVER-F

6. F_{FHT} (냉각재계통 보정계수, 100%에서만 적용) 106.602 × 1.000 % = 106.602 %

☐ 원자로 출력 100% : F_{FHT} 1.015

T_{RIH}	261.872	261.596	261.778	261.279	평균	261.63	℃
P_{ROH}	9847.696	9865.285	9835.753	9890.415	평균	9,859.79	Kpa(g)
ΔP_{HH}	1231.667	1175.833	1207.000	1189.000	평균	1,200.88	Kpa(d)

■ 원자로 출력이 100%가 아닐 경우 : F_{FHT} = 1 로 적용

7. F_C (비정상 반응도제어장치 배열) 보정 106.602 × 1.000 % = 106.602 %

8. F_F (연료형태에 따른 보정 계수) 보정 106.602 × 1.001 % = 106.743 %

☐ 모두 같은 연료 형태 F_F = 1.0
☐ 모두 같지 않은 연료 형태 F_F = 1.0044

9. K (최근 8시간 이내 출력/제어장치 변동) 보정 106.743 + 0.000 % = 106.743 %

10. 압력관 Creep Penalty(D_{CR}) 보정 106.743 + 5.244 % = 111.987 %

11. 열출력 측정 불확실도 Penalty (2005년 0.1%, 2006년 0.2%, 2007년 0.3%, 2008년 0.4%)
(단, 급수유량 측정기 보정 후 4년간은 0%로 적용 111.987 + 0.200 % = 112.187 %

12. 교정요구값(DC) 계산
DC = (CPPF + D_{TC} + D_B + D_{TILT} + D_{TAF}) × F_{FHT} × F_C × F_F + K + D **112.187 %**

Figure 11 ROP Calibration Factor Calculation

One of the major purposes of COMPAS is reducing routine jobs. For example, calibration of detectors such as Zone Pt., Vanadium, and Neutron flux detectors. Generally, the process of detector calibration is done step by step procedure that is import required DCC data, process it, then make a report. Procedures of Zone Pt. detector calibration are depicted in the Figure 12 and 13.

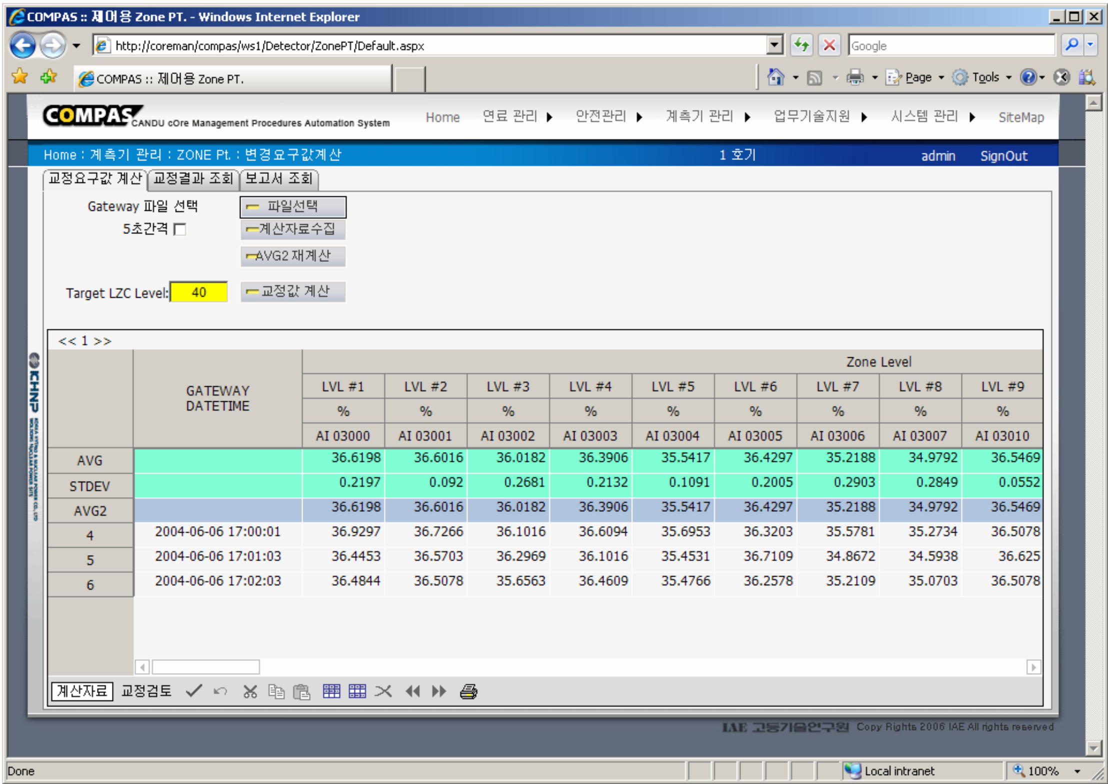


Figure 12 Import DCC Data for Zone Pt. calibration

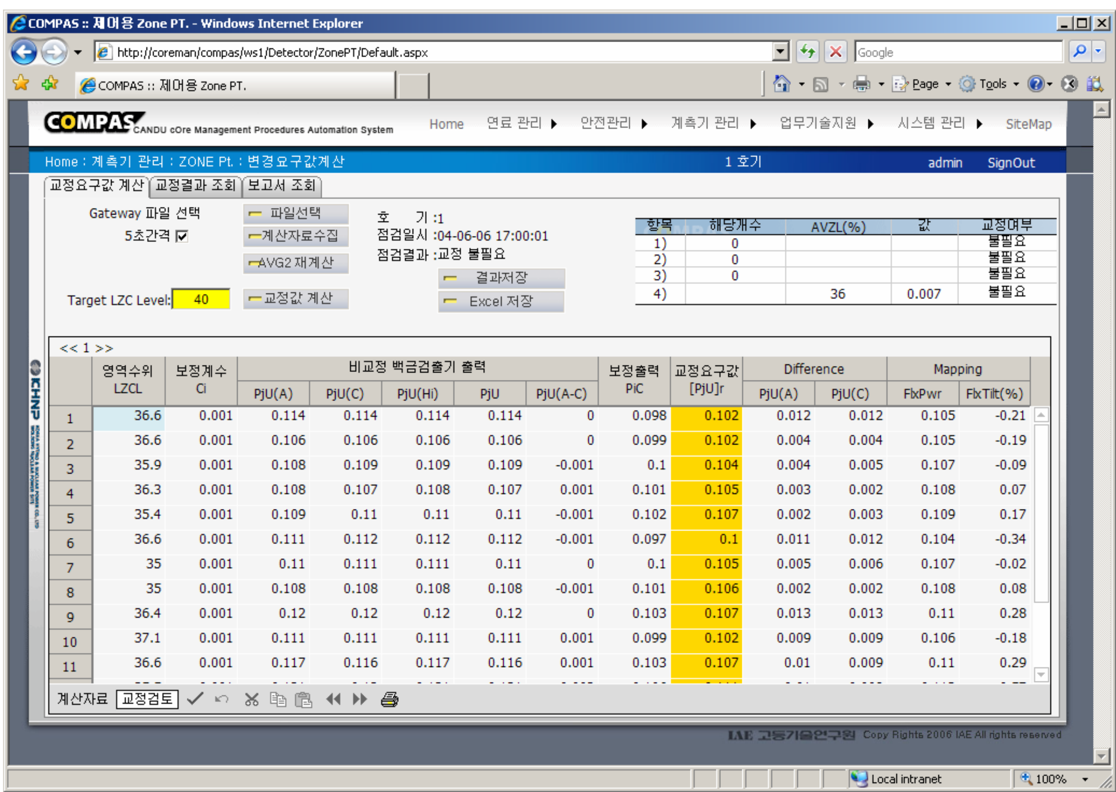


Figure 13 Determine Requirement of Calibration

In addition to major component, it is very useful to know current and future reactivity to ensure stable operation of the rector. Therefore, reactivity estimation module was also developed and integrated the COMPAS system. To calculate reactivity, several parameters should be known such as, Temperature Coefficient, Moderator Poison, Reactivity Device Position, Fission Product and so on [5]. Using these inputs and numerical method you can calculate reactivity at any time as depicted in the Figure 14.

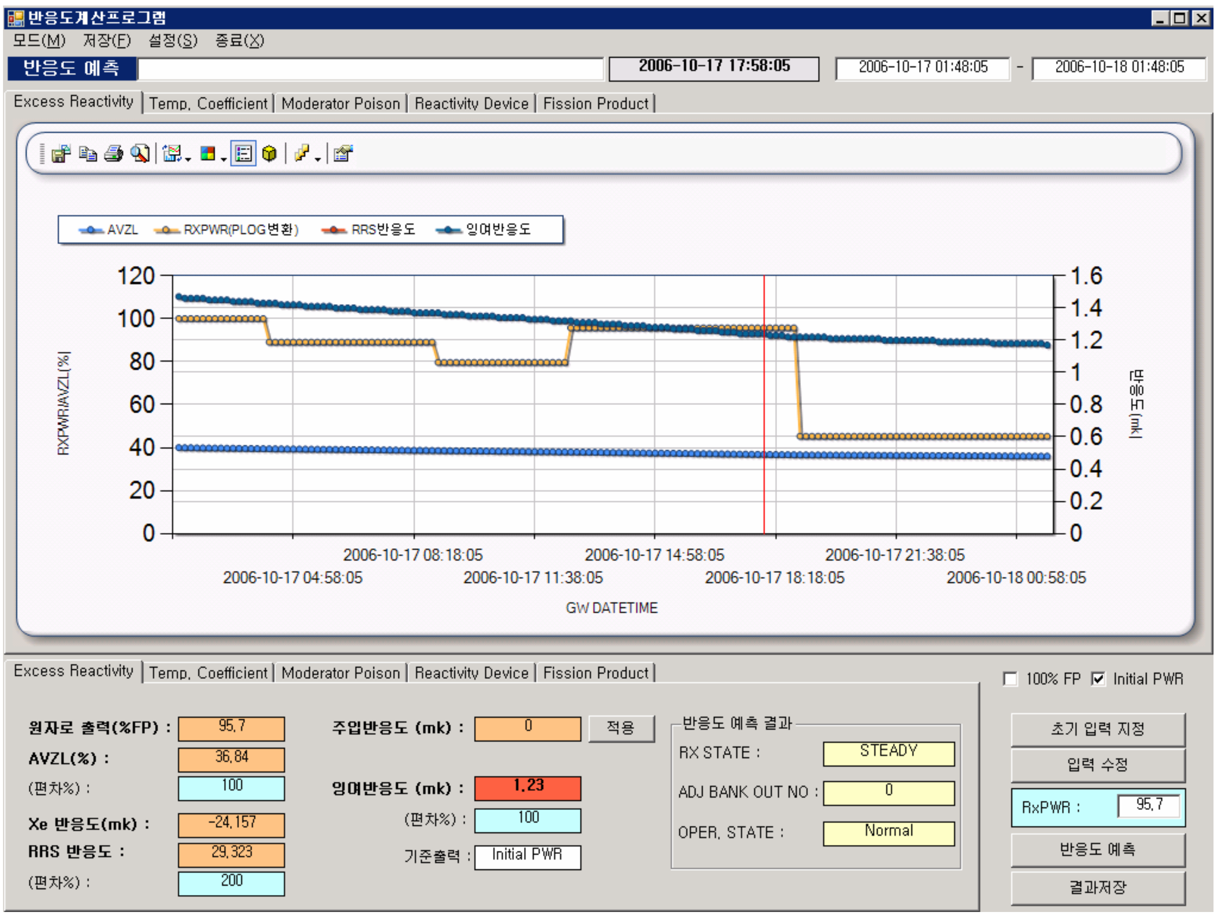


Figure 14 Estimation of Reactivity

3. Conclusion

We developed web-based CANDU core management procedures automation system not only to increase operational performance and efficiency but also to reduce human error, inconsistency of several co-work environments. It also makes and records historical knowledge and work database. Since the system minimizes routine jobs, operators can focus on the more productive tasks as well.

In this paper, we also show how to develop a web-based CANDU core management system in the view of OOP approach. Several steps were applied successfully including analysis, design, and implementation. Additionally, to overcome inherent constraint of web that is limited user interfaces, we introduced smart client technology. It can provide plentiful and flexible graphical user interface of classical client server application to web clients. Another new feature of the system is using web services to import DCC data. These features and functions are integrated under .NET framework to make ASP.NET web application.

As a conclusion, an automation system to support effective operation of CANDU is developed successfully. Currently, it is under testing, verification and validation phase by the member of Wolsong NPP unit 1 & 2. It is planned to deploy in real task in the end of this year and expected to increase the performance of the management. Moreover, we plan to migrate it into Wolsong unit 3 & 4 in the near future but the exact time is not determined yet. We expect the modification and migration the system without difficulties because it is portable, modular, localized by nature. It is a sort of convergence between information technology and nuclear engineering will make more prospect future of CANDU.

4. 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

[4] Sanghoon Lee, Eung Gon Kim, O Whan Kwon, and Choong Sub Yeum, A Study on the Development of Web Service based Operational Data Transfer System of CANDU Nuclear Reactor, Korea Society of IT Services, May 2006

[5] Daeyu Park, Sanghoon Lee, Ohwan Kwon, Seongmin Kim, Hyungbum Suh , Jinil Kim, Development of the CANDU Reactivity Module of COMPAS, KNS Conference, Fall, 2006