

문서중심 및 웹기반 핵설계 자동화 시스템의 설계 및 구현

박 용 수* · 김 종 경**

요 약

본 논문에서는 모델링과 전산코드 실행 등에 시간과 노력이 많이 드는 핵설계 업무를 자동화하기 위하여 IDP™(Innovative Design Processor)를 개발하였다. IDP의 기본 원리는 문서중심 설계와 웹기반 설계이다. 문서중심 설계란 프로그래머가 아닌 일반 설계자가 동적문서(active document)라는 문서를 작성하여 이를 특수한 프로그램이 과실행하도록 하면 해석결과와 표 및 그림 등이 담긴 완전한 설계문서를 자동적으로 얻게 됨을 말한다. 동적문서는 일반 HTML 또는 XML 편집기를 이용하여 작성될 수 있고 웹에서 또한 작성이 가능하다. LAMP(Linux/Apache/MySQL/PHP)의 웹기반 설계 환경하에서 설계마법사 형태의 서버 및 클라이언트 프로그램을 통해 설계 초보자도 쉽게 설계를 할 수 있게 되어 있다.

Design and Implementation of a Document-Oriented and Web-Based Nuclear Design Automation System

Yong Soo Park* · Jong Kyung Kim**

ABSTRACT

To automate nuclear design works which are time-consuming and man-power intensive, Innovative Design Processor (IDP™) is being developed. Two basic principles of IDP are the document-oriented design and the web-based design. The document-oriented design is that, if the designer writes a design document called active document and feeds it to a special program which has a robust parser, the final document with complete analysis, table and plots is made automatically. The active documents can be written with ordinary HTML/XML editors or created automatically on the web, which is another framework of IDP. Using the proper mix-up of server side and client side programming under the LAMP (Linux/Apache/MySQL/PHP) environment, the design process on the web is modeled as a design wizard style so that even a novice designer makes the design document easily.

키워드 : 자동화(Automation), 문서중심 설계(Document-Oriented Design), 웹기반 설계(Web-Based Design), 동적문서(Active Document)

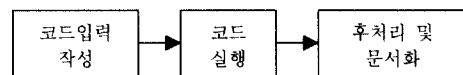
1. 서 론

원자력발전소 노심의 핵물리적 특성을 분석하는 핵설계 업무를 수행하기 위해서는 복잡한 3차원 모델링과 많은 종류의 전산코드가 필요하다. 즉, 정확한 모델링과 전산코드의 실행 및 데이터의 처리 등에 많은 시간과 노력이 들게 된다. 따라서 설계자의 수작업을 줄이고 설계품질과 생산성을 향상시키기 위한 많은 노력이 진행되어 왔다. 보편적으로 유닉스의 X윈도우 시스템 안에서 자동화 프로그램을 개발하는 것인데 모티프(Motif)와 C++를 이용한 것이 대표적이다.

어떤 언어를 사용하던지 간에 주된 개념은 항상 코드중심이다. 코드중심의 설계란 설계자가 전산코드의 입력을 마련하면 전산코드의 수행과 후처리(post-processing) 및 문서화

는 자동화 프로그램에 의해 정해진 양식에 따라 이루어짐을 말한다(그림 1).

이와 같은 접근 방식은 설계 절차에 큰 변화가 없는 경우에 적합할 수 있지만 설계자로 하여금 설계절차의 변화에 능동적으로 대처할 수 없도록 하는 것 또한 사실이다. 즉, 설계절차의 변경이 있을 경우 설계자는 자동화 프로그램을 프로그래머가 변경할 때까지 기다려야 하는 것이다.



(그림 1) 코드중심 설계

컴퓨팅 환경 또한 매우 급격히 변화하기 때문에 플랫폼 독립적인 소프트웨어는 필수 사항이 되고 있다. 따라서 이와 같은 요구를 만족할 수 있는 새로운 패러다임이 필요하다.

본 논문에서는 IDP™(Innovative Design Processor)라는

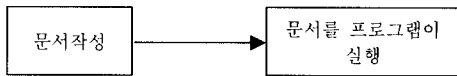
* 정 회 원 : 한전원자력연료(주) 설계기술원 선임연구원
 ** 정 회 원 : 한양대학교 원자력공학과 교수
 논문접수 : 2004년 4월 3일, 심사완료 : 2004년 7월 28일

진보된 설계자동화 시스템을 설계 및 구현한다. IDP는 설계자로 하여금 모든 설계 작업을 쉽게 할 수 있도록 도와주는데, 문서중심 설계(document-oriented design)와 웹기반 설계(web-based design) 라는 두 가지 기본 원리를 바탕으로 하고 있다.

2. 문서중심 설계

문서중심 또는 문서지향 설계란 설계자가 전산코드의 입력보다는 설계문서 작성에 초점을 맞추는 것이다. 즉, 설계자가 원하는 설계문서의 초안을 작성하여 이를 특수 프로그램이 실행하도록 하면 설계관련 일체의 결과값과 표와 그림 등이 담긴 최종 설계문서를 자동적으로 얻게 되는 것이다(그림 2).

모든 설계의 최종 결과는 문서화되어야 하므로 이와 같은 접근 방법은 설계자의 문서화 작업 부담을 크게 덜어준다. 또한 설계자는 문서를 편집할 수 있기 때문에 프로그래머의 도움없이 설계절차의 변경에 맞게 문서를 수정할 수 있다. 이 문서는 일반적인 형태의 문서가 아니라 동적 문서(active document)[1]라고 한다.



(그림 2) 문서중심 설계

2.1 동적문서

정보 기술의 발전은 문서에 대한 우리의 개념을 바꾸어왔다. 동적문서가 그 중의 하나이다. 동적문서는 컴퓨팅 환경과 문서편집 작업에 따라 문서 자신 또는 컴퓨팅 환경을 변화시킬 수 있는 문서라고 정의할 수 있다. 동적 문서에 포함된 응용프로그램을 동적문서 응용프로그램(active document applications)이라고 한다.

이 같은 응용프로그램은 일반적인 응용프로그램 보다 쉽게 구축이 가능한데 그 이유는 문서 편집기 기능의 이점을 활용할 수 있기 때문이다[2].

문서 편집기로서 인터리프(Interleaf)[3]는 동적문서의 디자인과 구현 및 실행에 좋은 플랫폼이라 할 수 있다. 인터리프의 중심구조는 내용 뿐 아니라 스타일정보를 포함하고 있는 컴포넌트(component)로 되어 있다. 예를 들어, 이 논문을 인터리프로 작성한다면 이 단락(paragraph)은 para라는 컴포넌트가 된다. 인터리프로 동적문서를 작성할 수 있기 때문에 문서만을 가지고 완전한 어플리케이션을 작성할 수 있는 것이다.

1990년대 중반 이후에 미국의 웨스팅하우스사는 핵설계평가 점검표(Physics Assessment Checklist, PAC)[4]라는 개선된 핵설계 방법론을 개발하였는데, PAC계산을 쉽게 수행하기 위해 인터리프를 이용하여 Active Procedure Toolkit (APtk)[5]이라는 핵설계 자동화 프로그램도 함께 개발하였

다. 인터리프로 작성된 동적문서에는 설계절차에 맞게 전산 코드 입력, 코드 실행과 표 및 그림이 담긴 결과물 요약 등을 담당하는 동적 컴포넌트(active component)가 포함되어 있다. 이 동적문서를 APtk가 처리하도록 실행시키면 APtk의 인터리프 파서(parser)가 동적 컴포넌트를 해석, 관련된 작업이 순차적으로 실행되어 최종 설계문서가 완성되는 것이다.

인터리프를 이용한 APtk는 매우 효율적인 설계자동화 도구이지만 다음과 같은 문제점이 있다.

- 동적문서가 인터리프에서만 편집이 가능한 인터리프 고유의 ASCII형태로 되어 있다.
- 인터리프는 오래되어 최신 컴퓨팅환경에 맞지 않는 이미 단종된 프로그램이다.
- APtk은 유닉스 C-shell과 awk 스크립트로 작성되었는데 이들 언어로는 다른 일반적인 형태의 동적문서를 파싱할 파서를 작성하기가 어렵다.

따라서 APtk의 동적문서 개념을 기본으로 새로운 동적문서 처리 시스템을 개발할 필요가 있다.

2.2 HTML

하이퍼텍스트 마크업 언어(Hypertext Markup Language, HTML)[6]는 인쇄물에 그리 적합한 언어가 아니지만 캐스캐이딩 스타일 시트(Cascading Style Sheet, CSS)[7]와 함께 쓰이면 문서를 잘 표현할 수 있다. CSS를 통해 문서 내용은 스타일과 분리될 수 있으며 동적 컴포넌트는 클래스(class)의 개념을 통해 구현될 수 있다. 따라서 동적문서는 HTML과 CSS로 작성이 가능하며 (그림 3)에 동적문서의 한 예가 있다.

```

<html>
<head>
<title>IDP Test</title>
<meta name = "keywords" content = "IDP, final parameter" />
<meta name = "description" content = "IDP - Final Parameter Cal." />
<link rel = "File-List" href = "/IDPtest.files/filelist.xml" />
<link rel = "stylesheet" type = "text/css" href = "http://koyu/css/IDP.css" />
</head>
<body>
<pre class = "APformula-hide">
docnumber = "foo"
num = 5
</pre>
<p class = "MsoNormal">Appendix to #docnumber#</p>
<br class = "PageBreak" clear = "all" />
<pre class = "APif"> num > 3 </pre>
<pre class = "APfor"> INDEX = 1 ; INDEX<= num ; INDEX++</pre>
<pre class = "APsubmit-hide">gnuplot < plot.inp </pre>
<pre class = "APimage">plot.eps 1 2.3 4 4.2 A4 </pre>
<br class = "PageBreak" clear = "all" />
<pre class = "APendfor">End of APfor loop</pre>
<pre class = "APendif">Bottom of APif Block</pre>
</body>
</html>
  
```

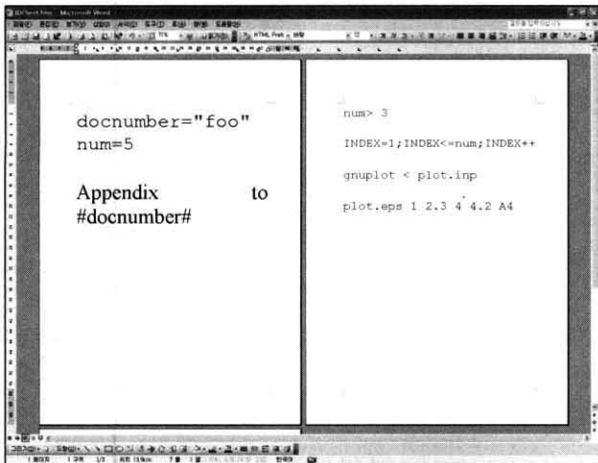
(그림 3) HTML형태의 동적문서

(그림 3)과 같은 구조화된 HTML문서는 awk의 정규표현식(regular expression)등을 이용하여 비교적 쉽게 파싱이 가능하나 (그림 4)와 같이 비구조화된(non-structured) 문서는 이벤트 기반(event-driven) 또는 DOM(Document Object Model) 기반의 강력한(robust) 파서가 필요하다. 본 논문에서는Perl언어의 HTML::Parser 라이브러리를 이용하고 적절한 콜백 함수(callback function)를 작성하여 비구조화된 문서에 대한 파서를 만들었다.

```
<html><head><title>IDP Test</title><meta name = "keywords"
content = "IDP, final parameter" /><meta name = "description"
content = "IDP - Final Parameter Cal." /><link rel = "File - List"
href = "/IDPtest.files/filelist.xml" /><link rel = "stylesheet"
type = "text/css" href = "http://koyu/css/IDP.css" /></head>
<body><pre class = "APformula-hide">docnumber = "foo"
num=5</pre><p class = "MsoNormal">Appendix to #docnumber# </p>
<br class = "PageBreak" clear = "all" /><pre class = "APif"> num >
3</pre><pre class = "APfor">INDEX = 1 ; INDEX<= num ; INDEX
++</pre>
<pre class = "APsubmit - hide"> gnuplot < plot.inp </pre>
<pre class = "APimage"> plot.eps 1 2.3 4 4.2 A4 </pre><br class
= "PageBreak" clear = "all" /><pre class = "APendfor"> End of APfor
loop</pre><pre class = "APendif"> Bottom of APif Block</pre>
</body></html>
```

(그림 4) 비구조화된 HTML 문서

동적문서는 간단한 노트패드나 HTML편집기를 이용하여 작성이 가능하다. MS 워드는 비록 전문 HTML 편집기는 아니지만 역시 이용이 가능하다. MS워드를 사용하면 사용자는 위치위(WYSIWYG) 환경에서 동적문서를 쉽게 작성할 수 있으며 MS 워드의 스타일 기능을 이용하여 동적 컴포넌트를 삽입할 수 있다. (그림 5)는 MS워드에서 동적문서를 연 모습을 보여주고 있다.

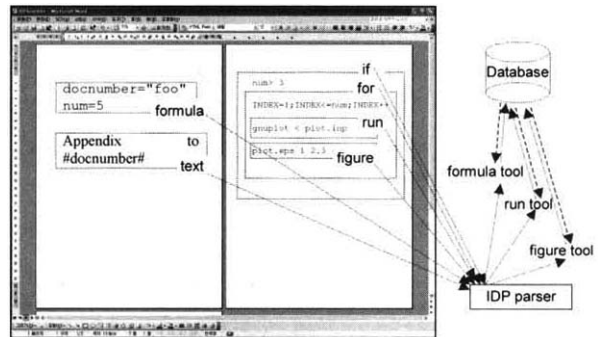


(그림 5) MS 워드에서 연 동적 HTML 문서

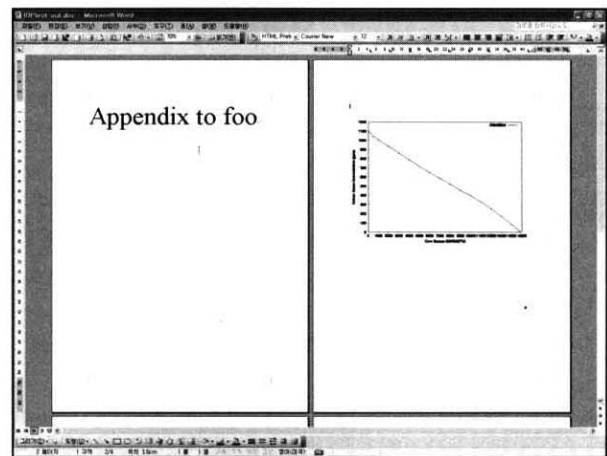
작성된 HTML문서는 IDP의 HTML 파서에 의해 해석되고 처리된다. 위 예의 경우 APformula-hide 컴포넌트에 의

해 변수할당이 이루어지고, #문자에 둘러싸인 변수가 변수 값으로 대체되며, num이라는 변수가 APif 컴포넌트에 의해 체크되면 APfor 컴포넌트에 의해 루프가 구성되어 그래프 작성 프로그램인 gnuplot이 루프가 끝날 때까지 수행된다.

(그림 6)과 같이 파서가 처리하는 과정에서 얻어지는 변수들은 모두 데이터베이스에 저장된다. 데이터베이스는 변수를 해쉬(hash)형태의 배열로 awk에서 읽을 수 있도록 하기 위한 파일기반의 데이터베이스이다. (그림 7)은 최종 처리된 동적문서를 MS 워드로 연 것이다.



(그림 6) HTML 문서처리 과정



(그림 7) 최종 HTML 문서

2.3 XML

전술한 바와 같이HTML은 인쇄보다는 디스플레이에 목적을 두고 고안되었기 때문에 전자출판에 최적의 솔루션은 아니다. 이 때문에 확장형 마크업 언어(Extensible Markup Language, XML)[8]가 개발된 것이다.

IDP에서는 XML을 주 마크업 언어로 선택하였다. 그러나 아직 HTML이 XML 보다는 더 범용적이고 많은 사용자층을 갖고 있는 현 상황을 고려하여 HTML형태의 동적문서 역시 지원하여야 한다. HTML형태의 동적 컴포넌트도 유지하면서 모든 동적 컴포넌트를 XML로 다시 정의할 수 있다 <표 1>.

XML태그는 자체 설명적(self-explanatory)이므로 이를 사용하면 동적문서는 더욱 논리적이고 간결하며 읽기 쉽게 된다. (그림 8)은 (그림 3)과 동등한 XML 동적문서를 보여주고 있다. 확장형 스타일 시트(Extensible Stylesheet Language, XSL)[9]가 XML에 일반적으로 사용되지만 XSL대신 CSS를 사용하는 것도 가능하다. (그림 8)에서 XML 태그와 일반적인 HTML태그가 동적문서에 혼용되어 있는 것을 볼 수 있다. 따라서 동적문서는 언제든지 순수한 XML문서로 쉽게 전환이 가능하다.

XML문서를 파싱하기 위해 Perl의 XML::Parser 또는 XML::LibXML 라이브러리를 이용할 수 있으나 (그림 9)과 같이 HTML::Parser 라이브러리의 XML mode를 이용하면 XML 문서의 파싱이 가능하다. 원래 XML mode는 XHTML문서를 파싱하기 위해 도입되었으나 일반적인 잘 구성된(well-formed) XML문서의 파싱에도 훌륭하게 작동한다.

<표 1> IDP의 XML 동적 컴포넌트

Name	Functions
cdrom	전산코드 결과로부터 cdrom 목록 작성
edit	변수값을 표 또는 지도형태로 표시
extract	전산코드 출력물에서 자료를 추출하여 변수로 저장
figure	이미지 처리(Postscript 또는 PNG)
for	C의 for-loop과 유사한 루프 구성
formula	awk의 연산식을 이용하여 변수를 할당 또는 계산
if	if 조건문과 같은 분기 구성
paste	간단한 텍스트 파일을 출력 문서에 삽입
read	파일 또는 텍스트 데이터베이스로부터 데이터를 읽음
Replace	외부 ASCII파일에 변수 치환
run	코드 작업을 포함한 쉘 스크립트 실행
?	PHP 스크립트 실행

```
<?xml version = "1.0"?>
<doc>
<head>
<title>IDP Test</title>
<keywords>IDP, final parameter</keywords>
<descrip>IDP - Final Parameter Cal.</descrip>
<link rel = "File-List" href = "/IDPtest.files/filelist.xml" />
<?xml-stYLESHEET type = "text/css" href = "http://koyu/css/IDP.css"?>
</head>
<body>
<formula type = "hidden">
docnumber = "foo"
num = 5
</formula>
<p class = "MsoNormal">Appendix to #docnumber#</p>
<pagebreak />
<if condition = "num > 3">
<for condition = "INDEX = 1; INDEX<= num; INDEX++">
<run type = "hidden"> gnuplot < plot.inp </run>
<figure src = "plot.eps" margin-left = "1" margin-right = "2.3"
margin-top = "4" margin-bottom = "4.2" paper = "A4" />
<pagebreak />
</for>
</if>
</body>
</doc>
```

(그림 8) XML 형태의 동적문서

```
#!/usr/bin/perl -w
use strict ;
use HTML::Parser ;
my $p = HTML::Parser -> new(api_version => 3,
    xml_mode => 1,
    declaration_h => [\&declaration, "text"],
    start_h => [\&start, "tagname, attr, attrseq, text"],
    text_h => [\&text, "text"],
    comment_h => [ "" ],
    end_h => [\&end, "tagname, text"],
    process_h => [\&process, "tagname"],
    default_h => [sub { print shift }, "text"],
);
```

(그림 9) HTML::Parser라이브러리 이용 예

의문부호(?)는 PHP(PHP : Hypertext Preprocessor)[10]를 처리하는 특별한 동적 컴포넌트이다. PHP는 요즘 각광을 받는 범용 웹개발 언어의 하나로서 HTML에 삽입될 수 있다. 서버쪽 스크립트뿐 아니라 클라이언트쪽 GUI 스크립트 및 심지어 수치계산[11]에도 사용이 가능하다.

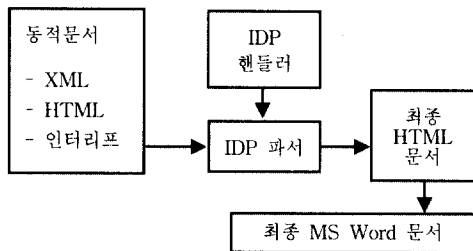
```
<?xml version = "1.0"?>
<doc>
<head>
<link rel = "File-List" href = "/IDPtest.files/filelist.xml" />
<?xml-stYLESHEET type = "text/css" href = "http://koyu/css/IDP.css"?>
</head>
<body>
<formula>PHPTableRequired = 1</formula>
<if condition = "PHPTableRequired">
<? //insert a table to show the status of ongoing projects
include "/users/www/htdocs/budget/inc/connect.inc" ;
if(!$year) $year = date("Y") ;
if(!$month) $month = date("m") ;
$depts = array ("KHNP", "MOST", "MOCIE", "KNFC") ;
echo "<table>\n<tr>\n";
foreach ($depts as $dp) {
$si = 0 ;
$sql = "SELECT * FROM $atype_table WHERE acode =
'$project_code' AND tyear = '$year' ORDER BY no" ;
$result = mysql_query($sql) ; // number of total projects
while($data = mysql_fetch_array($result)) {
if ($data['dept'] == $dp) {
$rcode[$dp][++$si] = $data['rcode'] ;
$iname[$dp][$si] = $data['iname'] ;
$class[$dp][$si] =
($data['class'] ? ($data['class']) : "&nbsp;." ;
$period[$dp][$si] = $data['period'] ;
$pm[$dp][$si] = $data['pm'] ;
}
}
$maxSize = $si + 2 ;
echo "<td class = 'table80' colspan = '$maxSize'>$dp</td>\n";
}
echo "</tr>\n</table>\n";
?>
</if>
</body>
</doc>
```

(그림 10) PHP 명령어가 담긴 동적문서

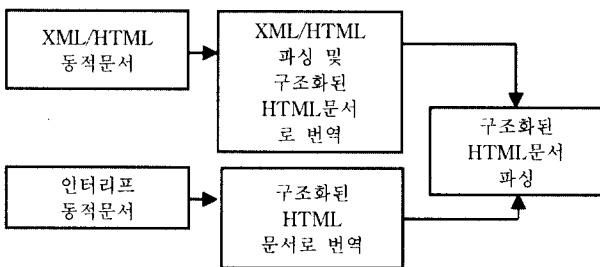
PHP의 강력한 점 중 하나는 다양한 데이터베이스를 지원한다는 것이다. 따라서 PHP 태그를 IDP에서는 새로운 동적 컴포넌트로 추가하게 되었다. PHP태그를 이용하면 데이터베이스 또는 네트워크 내의 어떤 문서를 동적문서에 쉽게 삽입하여 문서를 더욱 동적으로 만들 수 있다. (그림 10)은 MySQL 데이터베이스로부터 HTML 테이블을 첨가하는 동적문서의 예이다.

노트패드나 XML 문서 편집기를 이용하여 XML 동적문서를 작성하면 XML 파서가 문서를 해석, 동등한 HTML문서로 변환하여 HTML 문서와 같은 처리과정이 이루어진다. 따라서 최종 동적문서는 (그림 7)에 있는 HTML문서와 동등한 문서가 된다.

인터리프와 하위 호환성(backward compatibility)을 유지하고 인터리프에서 HTML로 완벽한 전환을 이루기 위해서 Perl언어로 작성한 번역기를 내장하였다. 사용자가 인터리프를 이용하여 동적문서를 작성하여도 내장된 번역기를 통해 최종문서는 HTML형태를 갖게 되며 이를 MS 워드로 열고 편집이 가능하다. (그림 11)은 IDP의 문서처리 부분을 도시하며 (그림 12)는 IDP 파서의 내부를 보여주고 있다.



(그림 11) IDP의 문서처리 흐름도



(그림 12) IDP 파서의 상세도

3. 웹기반 설계

동적문서는 마이크로소프트 윈도우 운영체제에서 작성하고 문서처리는 유닉스 시스템에서 이루어지는 것이 일반적인 상황이다. 따라서 이러한 다중운영체제 환경에서 모든 동적문서를 생성, 실행 및 관리할 수 있는 시스템이 존재한다면 사용자의 편의성은 극대화될 것이다. 이것이 웹기반 설계시스템이 개발된 이유이다.

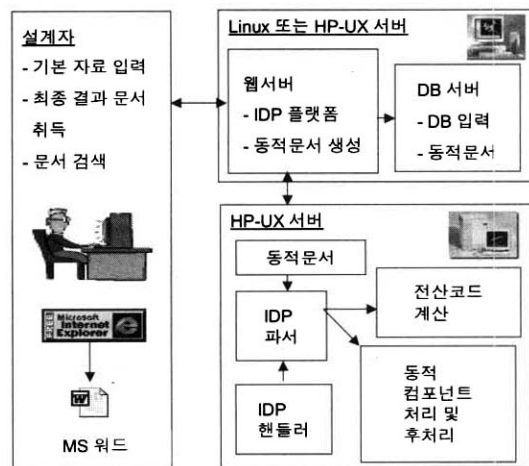
3.1 웹의 특성

웹은 개방성, 유용성, 확장성 등의 장점을 갖고 있기 때문에 설계에 유용하게 쓰일 수가 있다. 특히 웹의 하이퍼텍스트 전송 프로토콜(Hypertext Transfer Protocol, HTTP)은 다중운영체제 환경에 적합하다. 1990년대 말 이래로 한전원 자력연료 주식회사(KNFC)는 웹기반 설계자동화 시스템을 개발하여 왔으며[12], 여기에 문서중심 설계 개념을 도입함으로써 IDP의 아키텍처가 완성될 수 있었다.

LAMP(Linux/Apache/MySQL/PHP)환경(<표 2>)에서 서버측과 클라이언트측 프로그래밍을 적절히 사용하여 웹상에서의 설계절차를 설계마법사(design wizard)와 같은 방식으로 만들어 설계 초보자도 쉽게 설계결과물을 만들 수 있도록 하였다. 따라서 설계자가 웹 브라우저를 통해 설계에 필요한 기본 입력자료를 입력하면 완전한 동적문서가 생성 및 실행되어 최종 설계문서가 자동적으로 완성된다. (그림 13)은 IDP의 전반적인 아키텍처이다.

<표 2> 웹 개발 환경

	Server-Side Programming	Client-Side Programming
OS	Linux/HP-UX	MS Windows
Engine Program	Web Server (Apache)	Web Browser (MS Internet Explorer)
Languages	PHP, Perl, MySQL	HTML, JavaScript

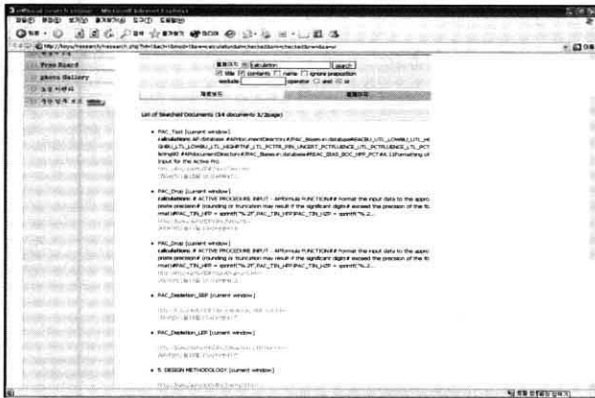


(그림 13) IDP의 아키텍처

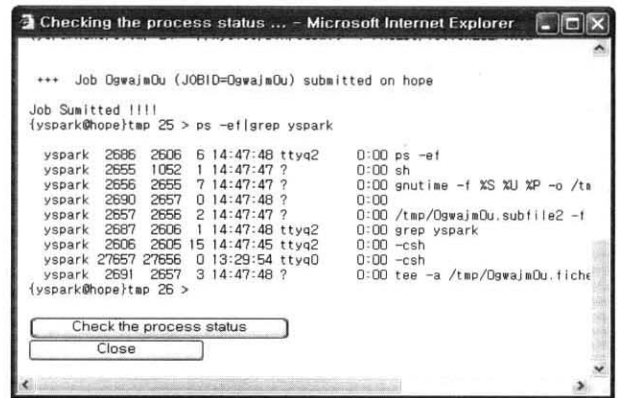
3.2 웹상에서 동적문서 관리

동적문서를 관리하기 위해 PHP와 MySQL[13]을 이용하여 웹데이터베이스 시스템을 개발하였다. 다양한 업무에 대한 수많은 동적문서와 매뉴얼 등 참고문서들이 웹데이터베이스 시스템에 저장되어 있다.

모든 동적문서와 참고문서들은 웹의 검색엔진을 통해 쉽게 검색이 가능하다(그림 14) 참조). 따라서 사용자의 손끝에서 모든 정보를 얻을 수 있는 것이다.



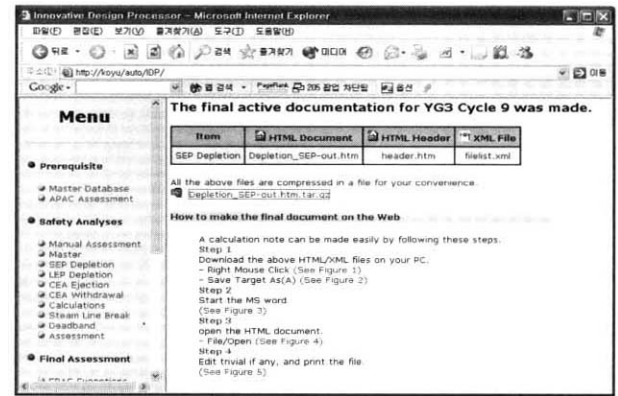
(그림 14) 문서 검색엔진



(그림 17) 전송된 문서를 웹상에서 실행

4. 핵설계자동화 시스템의 설계 및 구현

(그림 15)부터 (그림 18)은 동적문서가 어떻게 생성되고 실행되며 웹상에서 취득되는지를 보여준다. 설계자로 하여금 동적문서 편집에 더 많은 자유를 제공하기 위해 (그림 15)와 같은 간단한 위치위 HTML 편집기가 제공된다. (그림 16)은 설계자가 웹에서 생성된 동적문서 또는 MS 워드 등을 이용하여 작성한 동적문서중 하나를 선택할 수 있음을 보여준다.



(그림 18) 다운로드 가능한 최종 동적문서



(그림 15) 웹 상에서의 동적문서 작성



(그림 16) 동적문서 생성 및 파일 전송 단계

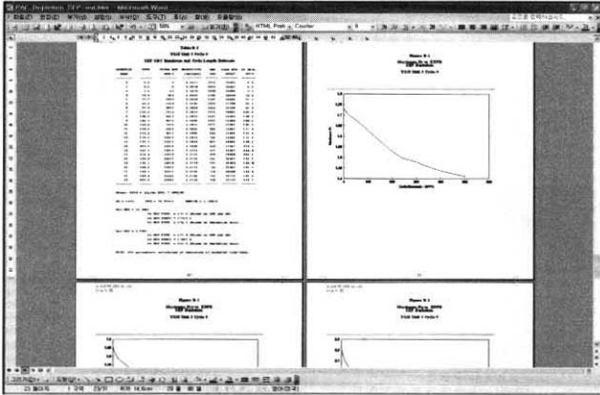
4.1 핵설계

PAC방법론에 의한 핵설계는 원자로심의 핵물리적 특성이 기존 안전해석 결과를 만족한다는 것을 확인하는 것이다. IDP를 이용하여 다음의 설계 항목에 대한 계산이 가능하다.

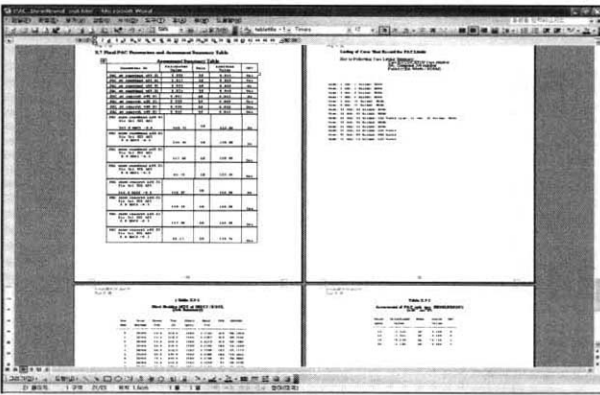
- 노심연소계산(Depletion calculations)
- 첨두출력 및 축방향 출력분포(Peaking factors and axial shapes)
- 제어봉 삽입계산(Symmetric rodged calculations)
- 미세격자계산(Fine-mesh data)
- 반응도계수(Reactivity coefficients)
- 동특성인자(Kinetics parameters)
- 정지반응도가(Scram worth and stuck rod worth)
- 붕소희석 사고해석(Boron dilution analysis)
- 증기관파단 사고해석(Steamline break analysis)
- 제어봉낙하 사고해석(CEA drop analysis)
- 제어봉이탈 사고해석(CEA ejection analysis)
- 제어봉인출 사고해석(CEA withdrawal analysis)
- 단일제어봉인출 사고해석(Single CEA withdrawal analysis)

(그림 19)는 (그림 15)에서 (그림 18)에 이르는 과정을 거쳐 얻어진 노심연소계산 결과를 보여주고 있다. 단일제어봉인출

사고해석 결과는 (그림 20)에 나타내었다. 이와 같이 IDP를 통해 핵설계가 고도로 자동화되었기 때문에 과거 수개월에 걸쳐 작업할 핵설계 업무를 1주일 이내에 끝마칠 수 있게 되었다.



(그림 19) 노심연소계산결과



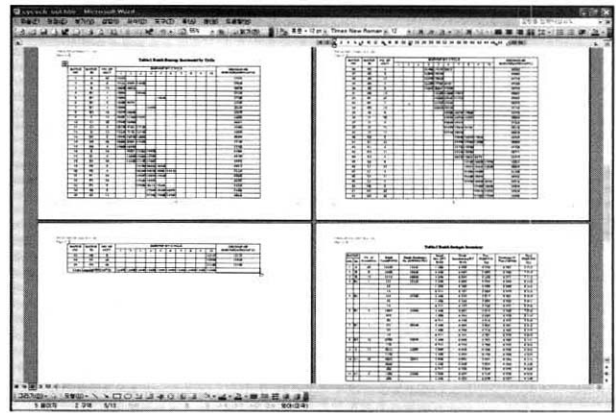
(그림 20) 단일제어봉인출 사고해석결과

4.2 장기주기계획표 작성

한전의 발전소 운영계획수립을 위한 장기주기 계획표 작성 업무는 교체노심 핵설계 업무 중 하나이다. 점반응도 모델과 선형반응도 모델을 이용하여 다가오는 노심 주기에 필요한 우라늄의 양과 농축도 등을 계산하여야 한다[14].



(그림 21) 장기주기계획표 작성을 위한 동적문서



(그림 22) 최종 장기주기계획표

이 작업을 위한 동적문서를 작성, IDP가 처리하도록 하면 쉽게 장기주기계획표를 작성할 수 있다. (그림 21)은 웹상에서 동적문서를 작성하는 것을 보여주고 있고 (그림 22)는 생성된 장기주기계획표이다.

4.3 평가 및 토의

위와 같이 구현된 시스템은 기존 수작업에 의해 생산된 결과와 비교할 때 생산 속도와 품질 면에서 뛰어난 결과를 보여주고 있다. 무엇보다 사용자는 동적 컴포넌트를 적절히 활용하여 자신의 업무에 맞는 동적문서를 쉽게 작성할 수 있게 되었다.

5. 결 론

설계자동화 시스템 IDP를 한국표준형 원자력 발전소의 노심핵설계 업무에 적용하기 위해 개발하였다. 이를 도입함으로써 핵설계업무에 드는 노력이 크게 감소되고 생산성이 향상될 것이 기대된다. 이 기술의 기본 개념은 열수력설계, 핵연료봉설계 등 다른 설계 업무에도 적용이 가능하므로 IDP는 핵연료설계의 전문분야에 걸쳐 사용이 가능하다.

설계 업무뿐 아니라 IDP는 모든 일반 업무에도 사용이 가능한데 그 이유는 최종 결과물은 컴퓨터코드의 수행 여부와 관계없이 일반적으로 표와 그림 등이 담긴 보고서 형태이기 때문이다. MCNP코드를 이용한 핵임계 계산이나 단순한 월 말 진도 보고서 생산 등이 그 예라 하겠다. 따라서 IDP는 이미 친숙해진 웹업무 스타일(web-work style) 환경에서 설계 및 연구 개발 업무 전반에 활용될 수 있다.

참 고 문 헌

[1] Robert Spinrad, "Dynamic Documents," Harvard University Information Technology Quarterly, Vol.VII, No.1, pp. 15-18, 1988.
 [2] Douglas B. Terry and Donald G. Baker, "Active Tioga

documents : an exploration of two paradigms," Electronic Publishing-Origination, Dissemination and Design, Vol.3, No.2, pp.105-122, 1990.

[3] Paul M. English and Raman Tenneti, "Interleaf active documents," Electronic Publishing-Origination, Dissemination and Design, Vol.7, No.2, pp.75-87, 1994.

[4] J. A. Brown, R. P. Harris, W. M. McDonald, R. E. Henderson, "The PAC Methodology for Reload Design Assessment," Proc. of ANS Topical Meeting - Advances in Nuclear Fuel Management II, ANS, USA, 1997.

[5] M. P. RUBIN and S. G. WAGNER, "Users Manual for the Active Procedure Toolkit (Aptk)," CE-CES-167-P Rev. 10, ABB CE Nuclear Operations, 1999.

[6] Chuck Musciano and Bill Kennedy, HTML & XHTML : The Definitive Guide, 5th Edition, O'Reilly & Associates, Inc., USA, 2002.

[7] Eric Meyer, Cascading Style Sheets : The Definitive Guide, O'Reilly & Associates, Inc., USA, 2000.

[8] W3C, Extensible Markup Language (XML) 1.0, W3C Recommendation, 2nd Edition, 2000.

[9] Doug Tidwell, XSLT, O'Reilly & Associates, Inc., USA, 2001.

[10] Rasmus Lerdorf and Kevin Tatroe, Programming PHP, O'Reilly & Associates, Inc., USA, 2002.

[11] VOLKER GOEBBELS, "Scientific Applications with PHP," Proc. of International PHP 2003 Conference," http://www.phpconference.de/2003/slides/business_track/goebbels_scientific.pdf, Frankfurt, Germany, 2003.

[12] Y. S. PARK, et al., "Interactive Nuclear Design Analysis

Process Automation on World Wide Web," Proc. of PHYSOR 2000, ANS, Pittsburgh, USA, 2000.

[13] George Reese, MySQL Pocket Reference, O'Reilly & Associates, Inc., USA, 2003.

[14] Y. S. PARK, et al., "Establishing the Long-Term Fuel Management Scheme Using Point Reactivity Model," Journal of Nuclear Science and Technology, Vol.31, No.10, pp.1001-1010, 1994.



박용수

e-mail : yspark@knfc.co.kr
 1985년 한양대학교 원자력공학과(공학사)
 1987년 한양대학교 대학원 원자력공학과
 (공학석사)
 1988년~1996년 한국원자력연구소 연구원
 1997년~현재 한전원자력연료(주) 설계기술원
 선임연구원

관심분야 : XML, 설계자동화, 인공지능 등



김종경

e-mail : jkkim@nural.hanyang.ac.kr
 1980년 뉴욕 주립대 원자력공학과(공학사)
 1982년 미시간대 대학원 원자력공학과
 (공학석사)
 1986년 미시간대 대학원 원자력공학과
 (공학박사)

1986년~1987년 한국원자력연구소 유체과학자
 1987년~현재 한양대학교 원자력공학과 교수
 관심분야 : 실시간 시스템, 전산수학 등