

# XML Web Service를 이용한 조류계산 프로그램의 분산처리

論 文

52A-4-3

## Distributed Processing of Load Flow Program Using XML Web Service

崔 章 欽\* · 金 建 中\*\*

(Jang-Hum Choi · Kern-Joong Kim)

**Abstract** - System analysis programs have been developed by several different research groups. Especially, the programming languages and the developing environments of algorithm modules and user interface modules are different. And therefore, the differences have degraded interoperability and reusability of the system analysis modules. In order to solve this problems, a general binary interface has designed and the component based on the interface has developed as well. However, sometimes each interface is incompatible because those are designed on the particular vendor. In this paper, we deals with XML web service, sort of distributed processing architecture, which is not restricted by not only the existing internet standard but also any programming language or any vendor. Because of its platform independent, each module can be updated and extended independently.

**Key Words** : XML web service, processing architecture, interface, component, interoperability, reusability

### 1. 서 론

현재까지 계통 해석 프로그램의 개발은 각기 다른 연구 그룹에 의해 독자적으로 개발되어 왔다. 따라서 각 연구 그룹별로 계통 해석 프로그램 개발 언어가 다르고 각 연구결과들 간의 통합 및 유지보수가 용이하지 않았다. 즉 기존의 계통 해석 프로그램은 사용자 화면을 거의 고려하지 않고 알고리즘 위주의 개발에 치중하였다. 그에 반해 최근의 사용자 요구사항은 편리한 사용자 화면을 요구하고 있어 기존의 알고리즘 개발 언어로는 화면개발이 불가능하고 최근의 사용자 화면 개발용 언어를 사용해야 하는 실정에 있다. 그로인해 계통 해석 프로그램을 구성하는 알고리즘 모듈과 사용자 화면 모듈은 서로 다른 언어로 개발되어 각 모듈간의 통합 및 유지 보수가 용이하지 않았다. 또한 각 알고리즘간에도 개발 언어가 다르고 개발 환경이 달라 계통 해석 모듈의 통일성 및 재활용성을 저해하는 요인이 되어 왔다. 이러한 문제의 해결 방안의 하나로써 일반화된 바이너리 인터페이스를 디자인하고 그 인터페이스를 기반으로 하여 컴포넌트(바이너리이므로 컴파일이 필요없음)를 개발하였다. 이러한 컴포넌트를 개발함으로써 사용자를 위한 응용프로그램을 개발할 때 바이너리 컴포넌트를 재가공하지 않고 직접 사용할 수 있도록 하였다. 그러나 이러한 인터페이스는 특정 회사에서 제안한 경우가 많아 서로 다른 인터페이스를 사용하는 경우에는 호환되지 않는 경우가 있다.

결국 기존의 컴포넌트는 바이너리 인터페이스를 디자인하고, 그 인터페이스를 기반으로 하여 컴포넌트를 개발함으로써 개발 언어, OS, 하드웨어 등의 종속적 요인을 배제하고자 하였지만 서로 다른 인터페이스 사이의 호환이 이루어지지 않기 때문에 개발 언어에 대한 종속성만을 배제하게 되었다. 따라서 바이너리 인터페이스를 기반으로 한 컴포넌트를 이용한 전력계통 해석 프로그램은 OS, 하드웨어 등의 사용환경에 종속적으로 개발되었다. 이러한 바이너리 인터페이스를 기반으로 한 컴포넌트 구조(예를 들면, Microsoft사의 COM, SUN사의 beans 등)는 OS, 프로그램 개발 언어, API (Application Programming Interface), SDK(Standard Development Kit)가 진화함에 따라 전력계통 해석 모듈과 사용자 화면 모듈을 동시에 변경해야 하며, 해석 모듈의 변경 또는 사용자 화면 모듈의 변경이 발생할 때마다 라이브러리를 업데이트해야만 사용이 가능하였다.

이런 문제를 해결하기 위해서는 각 모듈의 적용 머신을 분리하여 화면 모듈과 계통해석 모듈이 서로 독립적인 구조를 가지도록 해야 한다. 이러한 구조가 분산 구조이며 Microsoft사의 DCOM, SUN의 RMI 등이 이러한 분산 구조를 지원하고 있다. 그러나 각각의 회사가 제안하고 있는 분산 구조는 데이터 전송 규약이 제조사마다 다르기 때문에 서로의 결합이 용이치 않았다.

이에 본 논문에서는 W3C에서 인정한 국제 표준인 SOAP (Simple Object Access Protocol) 규약의 데이터 메시지 포맷팅을 기반으로 하여 통신하는 XML Web Service를 이용한 인터페이스를 제안하고자 한다.[1] 특히 전력계통에서 계통해석 및 계통운용의 기본이 되고 있으며 계통의 운용뿐만 아니라 계통의 확장계획, 계통의 제어 등에서도 그 기반을 이루고 있는 조류계산 프로그램에 대해 XML Web Service를 적용하고자 한다. XML Web services에 의해 수행되는 조류계산

\* 正 會 員 : 忠南大 電氣工學科 博士課程

\*\* 正 會 員 : 忠南大 電氣工學科 教授 · 工博

接受日字 : 2002年 11月 14日

最終完了 : 2003年 2月 12日

의 가장 큰 목적은 OS, 하드웨어등의 사용환경이 다른 각각의 사용자들이 조류계산 컴포넌트를 사용하기 편리하도록 제공하는 것이다. 즉 XML Web Service에서 조류계산 해석 모듈을 다양하게 제공하면 조류계산 해석 모듈을 사용하고자 하는 사용자는 조류계산용 사용자화면 프로그램이나 Web 화면을 구성하여 조류계산을 수행하거나 조류계산을 기반으로 한 다른 알고리즘을 수행하는 응용 프로그램을 구성하고 조류계산용 XML Web services에 간단한 XML 기반 메시지를 보냄으로써 조류계산을 수행할 수 있게 된다. 이때 사용자가 보내는 메시지는 조류계산에 의해 해석될 계통의 정보, 조류계산 해석 옵션, 정보 출력 옵션 등과 같은 매개 변수를 조류계산용 XML Web services에 제공하게 되고, 조류계산용 XML Web services는 조류계산 해석을 수행하여 그 결과를 간단한 XML 기반 응답 메시지로 해당 호출 응용 프로그램에 알려 준다. 그러면 해당 응용 프로그램에서 그 결과를 이용하여 사용자에게 원하는 정보를 출력할 수 있게 된다.

이와 같이 XML Web Service를 이용함으로써 알고리즘 서버를 인터넷에 공유하여 계산 루틴을 필요로 하는 다양한 요구자가 알고리즘을 개발하지 않고, 서버를 액세스하여 개발 시간과 노력을 줄일 수 있으며 각 사용목적에 맞는 다양한 응용 프로그램들이 통일된 IO로써 개발될 수 있다. 또한 강력한 계산 서버로 계산 루틴을 옮기는 구조를 제안함으로써 안정도 해석 등과 같은 많은 계산량을 요구하는 계통 해석 모듈의 계산 속도를 향상시킬 수 있으며 다수의 서버를 사용한 분산 컴퓨팅을 하기 위한 발판 마련할 수 있게 된다.

## 2. XML Web Service를 이용한 조류계산 분산처리

### 2.1 라이브러리 기반의 프로그램 구조

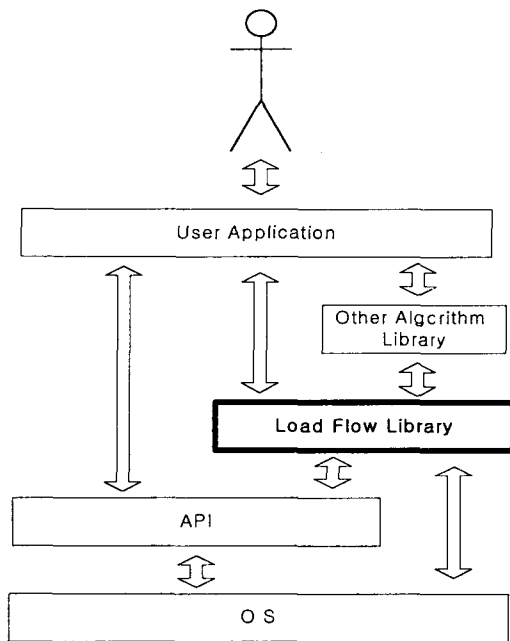


그림 1 기존의 라이브러리 기반 프로그램 아키텍처  
Fig 1 The existing library-based program architecture

[그림 1]은 기존의 프로그램 개발에서 주로 사용되고 있는 라이브러리를 기반으로 하는 프로그램의 구조를 나타내고 있다. 라이브러리를 기반으로 하는 프로그램 구조에 의해 개발되는 전력계통 해석 프로그램들은 조류계산 해석 모듈을 개발한 후 그것을 내부적으로 사용하는 다른 계통 해석 모듈을 개발하거나 조류계산 해석 모듈을 편리하게 사용하기 위한 사용자 화면용 프로그램을 개발하고 있다. 이때 조류계산 해석 모듈, 조류계산을 응용한 다른 계통 해석 모듈, 그리고 사용자 화면용 프로그램은 각기 다른 개발 환경에서 개발되고 있기 때문에 시스템의 통합 및 유지보수 그리고 시스템의 확장에 많은 시간과 비용이 요구되고 있다.

그에 반해 사용자는 사용자 화면 모듈을 통하여 여러 라이브러리의 다양한 기능을 사용하기를 희망하며, 편리하고 쉬운 다양한 인터페이스를 위하여 진화된 플랫폼(OS, Hardware, API)으로 이동하기를 희망한다. 그런데 기존의 라이브러리를 기반으로 하는 프로그램 구조는 플랫폼(OS, Hardware, API, SDK)에 종속되는 인터페이스(프로토타입의 인터페이스)를 통하여 라이브러리를 호출하기 때문에 사용자의 플랫폼이 진화해 감에 따라 조류계산 해석 모듈과 사용자 화면 모듈은 동시에 변경해야 한다. 즉 조류계산 해석 모듈, 조류계산 해석 모듈을 이용한 계통 해석 모듈, 그리고 사용자 화면 모듈을 분리하여 유지보수 및 확장을 하는 것이 아니라 함께 유지보수 및 확장을 해야 하는 부담을 가지고 있다.

### 2.2 Web Service Server와 클라이언트

기존의 라이브러리를 기반으로 하는 프로그램의 구조는 사용자의 플랫폼(OS, Hardware, API, SDK)에 종속적이고 각각의 해석 모듈과 사용자 화면 모듈을 동시에 유지보수 및 확장을 해야 하는 부담을 안고 있다. 이에 본 논문에서는 Web Service Server를 이용하여 그 기본 구조가 쉽게 변하지 않는 각각의 해석모듈은 Web Server에서 수행하고 사용자 요구에 따라 빠르게 변화하는 응용프로그램은 클라이언트에서 관리하는 분산처리 기법을 제안한다. 이때 Web Service Server와 클라이언트 사이의 통신은 국제 표준으로 인정받은 SOAP(Simple Object Access Protocol) 규약의 데이터 메시지 포매팅을 기반으로 하여 통신하는 XML기술을 이용함으로써 응용프로그램 개발자가 특정 개발 언어에 구애받지 않도록 하며 인터넷 프로토콜도 HTTP와 SMTP등을 다양하게 선택할 수 있도록 하였다. 또한 조류계산 해석 모듈을 Web Service Server에서 분산처리함으로써 계통 해석 프로그램이 사용자의 플랫폼(OS, Hardware, API)에 종속되지 않도록 할 수 있으며 각 해석 모듈과 사용자 화면 모듈을 완전히 분리하는 것이 가능하기 때문에 각각에 대해 독자적으로 유지보수 및 확장을 할 수 있다.

Web Service 서버는 인터넷상에 연결되어 있으며, 조류계산 해석 모듈의 인터페이스를 W3C에서 정한 XML의 형태로 공개하고 있다. Web Service 서버의 조류계산 해석 모듈을 사용하는 클라이언트는 일반 응용프로그램이나 웹 응용프로그램 등이 될 수 있다. 클라이언트는 Web Service Client용 라이브러리를 통하여 조류계산 해석 모듈을 탑재하고 있는 Web Service 서버와 통신한다. 기본적으로 Web Service 서버와 클라이언트가 동기되어 작업을 수행하지만 많은 데이터

교환이 필요하거나 호출에 많은 시간이 걸리는 경우에는 비동기 호출도 가능하다.

웹서버상의 조류계산 해석 모듈을 사용하는 방법은 크게 윈도우즈 응용프로그램을 이용하는 방법과 웹서버를 이용하는 웹 응용프로그램이 있다. 윈도우즈 응용프로그램에서는 서버와 직접 데이터를 주고받으며, 클라이언트에서 데이터 직접 처리하기 때문에 데이터 처리가 편리한 반면, 각 윈도우즈 응용 프로그램에 대한 유지보수를 클라이언트별로 해야하는 단점이 있다. 웹서버를 이용한 웹 응용프로그램의 경우에는 사용자가 모든 정보를 서버를 통해 액세스하기 때문에, 서버를 유지보수 함으로써 응용프로그램을 유지 보수 할 수 있는 장점이 있는 반면, 서버에서 많은 사람이 접속할 경우 성능이 저하될 수 있는 문제점이 있다.

### 2.3 Web Service를 이용한 윈도우즈 응용프로그램

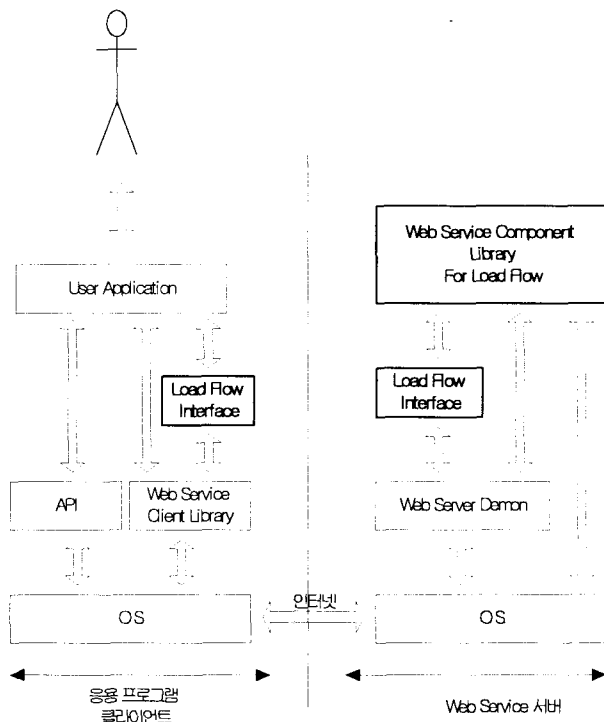


그림 2 Web Service를 이용한 윈도우즈 응용프로그램의 구조  
Fig 2 The structure of windows application program using web service

[그림 2]는 Web Service를 이용한 윈도우즈 응용프로그램의 구조를 보여주고 있다. 윈도우즈 응용프로그램과 Web Service 서버는 1단계의 간단한 구조를 가진다. 윈도우즈 응용프로그램은 서버에서 제공하는 XML을 기반으로 하는 인터페이스를 이용하여 서버의 조류계산 해석 모듈에 접근할 수 있다. 서버와 클라이언트는 인터넷을 통해 통신하기 때문에 일반적인 경우에는 http 프로토콜을 사용하고 보안이 요구되는 경우에는 https 프로토콜을 사용하게 된다. 데이터 구조는 XML기술을 이용한 SOAP을 사용게 되는데 SOAP은 국제표준으로 공개되어 있기 때문에 윈도우즈 응용프로그램 개발자는 특정개발 언어에 의존하지 않아도 된다. 따라서, 응

용프로그램 개발자는 사용자 화면 모듈에 가장 알맞은 언어를 선택할 수 있고, 그것을 이용하여 다양한 화면을 사용자에게 제공할 수 있다. 또한 알고리즘과 응용 프로그램이 완전히 분리되어 있기 때문에 서로 독자적인 개발이 가능하다.

그러나 인터넷을 통해 접속하기 때문에 데이터의 양이 많은 경우에는 속도문제가 있고, 또 인터넷망이 불안정한 경우 데이터의 손실이 발생할 수도 있다. 그러므로, 이 구조는 데이터량이 적고, 인터넷망이 안정적인 내부망으로 구현된 인트라넷(Intranet)에서 권장된다.

### 2.4 Web Service를 이용한 웹 응용프로그램

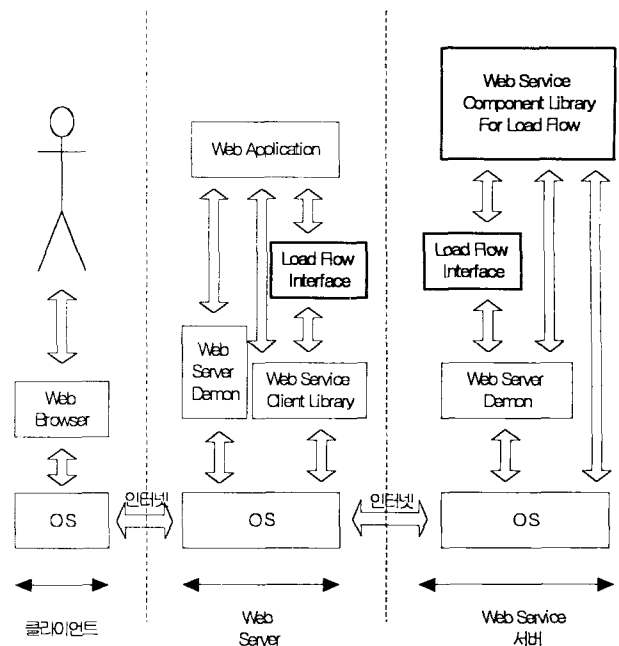


그림 3 Web Service를 이용한 웹 응용프로그램의 구조  
Fig 3 The structure of web application program using web service

웹 응용프로그램과 웹서버 사용자의 관계는 응용프로그램과 웹서비스보다 다소 복잡한 관계로 이루어지며 두 개의 서버를 필요로 한다. 웹 서버는 Web Service 서버와 사용자의 웹 응용프로그램과의 중간 매개체로써 사용자의 작업 요구 사항을 전달받고 그 요구 사항에 따라 Web Service 서버의 조류계산 해석 모듈을 호출한다. 따라서 사용자 화면 모듈과 사용자 요구사항의 처리는 웹 응용프로그램에서 처리되며, Web Service의 호출은 웹 서버에서 수행된다. 이런 구조에서는 사용자가 웹 응용 프로그램에 대한 특별한 인스톨 과정이 없더라도 인터넷 웹 페이지에 접속함으로써 조류계산을 수행하고 그 결과를 확인할 수 있다. 웹 서버와 Web Service 서버를 튼튼한 네트워크로 연결시키면 데이터량이 많더라도 큰 문제가 되지 않는다.

2.4 조류계산 분산처리용 Web Service Server의 구현

본 논문에서는 전력계통 해석과 운용의 기본이 되는 조류 계산에 대한 분산처리용 Web Service Server를 구성하였다. 조류계산 분산처리용 Web Service Server는 XML Web Service를 이용하여 구현되어 윈도우즈 응용 프로그램이나 웹 응용 프로그램에서 직접 액세스가 가능하며 다른 계통 해석 모듈의 서브 루틴으로도 이용이 가능하다. 또한 XML Web Service를 이용하여 조류계산을 서버에서 분산처리함으로써 조류계산은 사용자의 플랫폼에 전혀 영향받지 않는다. 그리고 SOAP 규약에 의한 XML 메시지의 데이터 포맷이 변경되지 않는다면 조류계산과 각 응용 프로그램들은 완전히 독립되기 때문에 독자적으로 확장 및 유지보수를 수행하는 것이 가능하다.

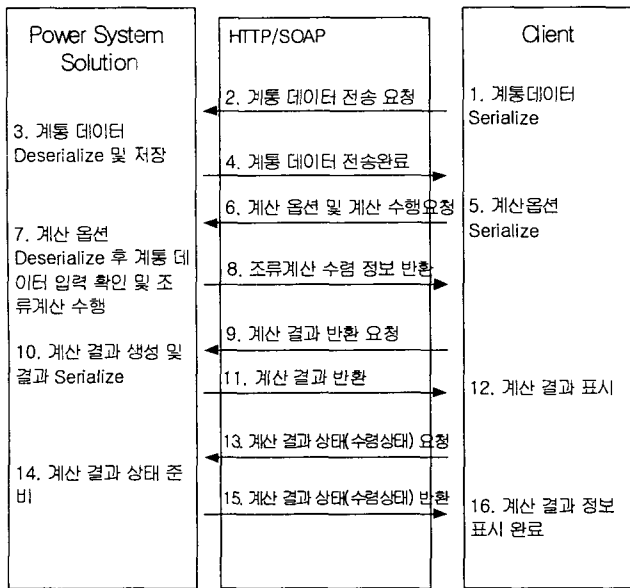


그림 4 Web Server와 응용 프로그램의 통신 시퀀스  
 Fig 4 The communication sequence between web server and application programs

[그림 4]는 조류계산 분산처리용 Web Server와 응용프로그램과의 통신 시퀀스를 보여주고 있다. 조류계산 분산처리용 서버와 클라이언트 사이는 HTTP/SOAP을 통해서 통신을 하고 있으며 그 통신 시퀀스는 다음과 같다.

- ① 데이터를 클라이언트가 서버로 보낸다.
- ② 서버는 클라이언트가 보낸 데이터를 보관한다.
- ③ 클라이언트는 계산옵션과 함께 계산 수행 함수를 호출한다.
- ④ 서버는 저장된 데이터를 가지고 클라이언트의 계산옵션에 따라 계산을 수행한다.
- ⑤ 클라이언트는 계산된 결과를 요구한다.
- ⑥ 서버는 계산된 결과를 클라이언트로 보낸다.
- ⑦ 클라이언트는 계산된 결과의 수렴상태를 요구한다.
- ⑧ 서버는 계산된 결과의 수렴상태를 반환한다.

3. 사례연구

XML Web Service를 이용한 조류계산의 분산처리시 조류 계산 해석 모듈과 각 응용 프로그램이 완전히 분리되어 개별적으로 관리가 가능해짐을 보이기 위하여 조류계산 분산 서버를 구축하고 그 서버의 조류계산 해석 모듈을 이용한 응용 프로그램을 구성하였다. 응용 프로그램은 조류계산 해석 모듈에 종속되지 않고 각기 다른 개발 환경에서 개발하고 관리할 수 있음을 증명하기 위하여 윈도우즈 응용 프로그램과 웹 응용 프로그램으로 구성하였다.

3.1 윈도우즈 응용프로그램

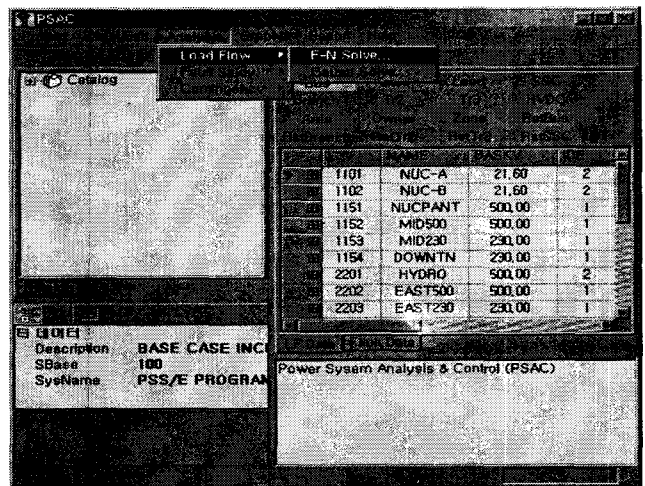


그림 5 윈도우즈용 응용프로그램의 실행 화면  
 Fig 5 The example of an executed windows application program

[그림 5]에 소개된 윈도우즈 응용프로그램은 파일을 입력받아 조류계산을 수행하는 프로그램으로 알고리즘 부분을 클라이언트에서 처리하지 않고 Web Service에서 처리하며 사용자의 데이터 편집요구는 윈도우즈 응용 프로그램의 Grid (데이터를 표처럼 사용하는 형태) 화면을 통해 처리할 수 있도록 구성되어 있다.

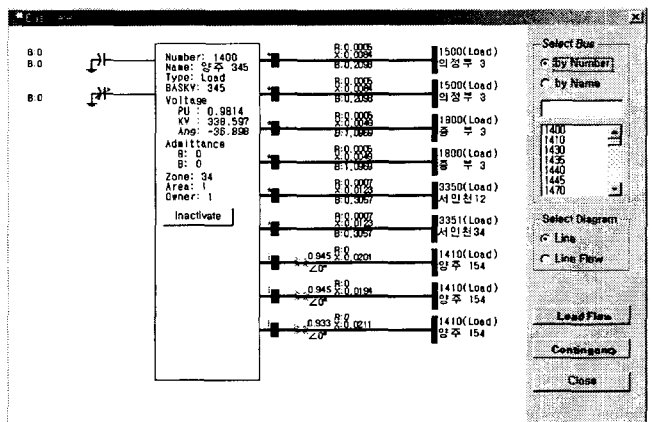


그림 6 윈도우즈용 응용프로그램의 모션 Diagram  
 Fig. 6 The Bus diagram of windows application program

[그림 6]은 각 모션별로 데이터를 편집하거나 결과값을 표시하기 위한 모션 Diagram이다. 이 부분도 조류계산 해석 부분과 관련이 없는 편집이나 결과 표시 부분이기 때문에 윈도우즈 응용프로그램에서 구현이 가능하다.

결국 사용자의 요구사항을 다양하게 반영하면서 해석 알고리즘과 사용자 화면을 분리하여 독립적으로 확장하는 것이 가능하게 된다.

### 3.2 웹 응용프로그램

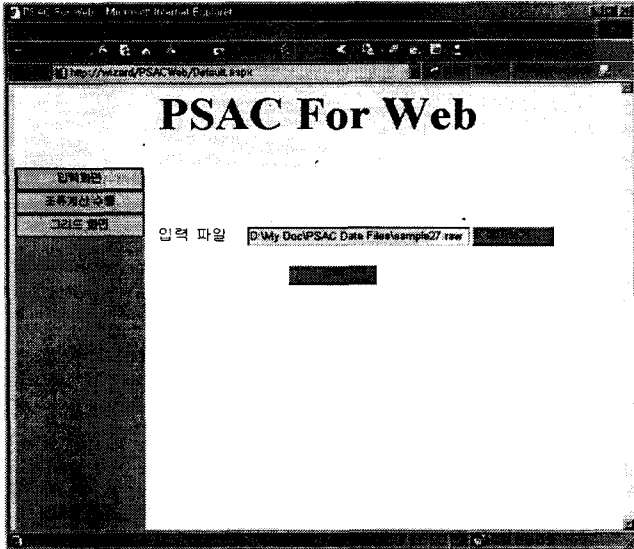


그림 7 웹용 응용 프로그램의 데이터 입력 화면  
Fig. 7 The data input form of web application program

[그림 7]은 웹 응용 프로그램의 첫 연결화면으로 입력파일을 선택하고 서버로 전송할 수 있다. 파일 형태는 PSS/E 28 버전의 파일을 지원한다.

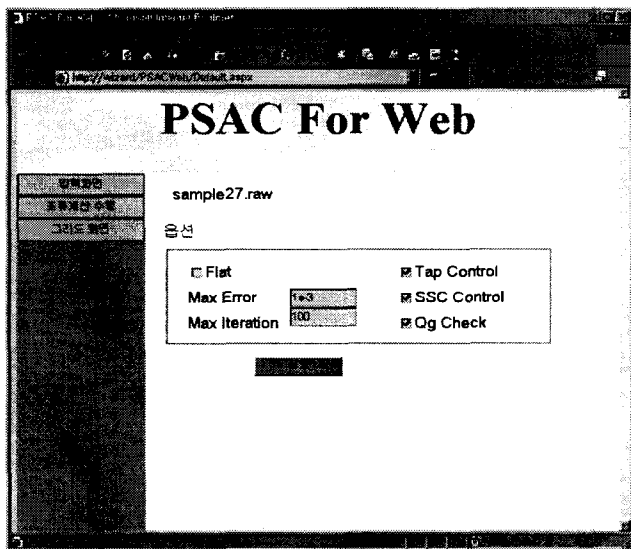


그림 8 웹용 응용 프로그램의 조류계산 수행 화면  
Fig. 8 The execution of the load flow of web application program

[그림 8]는 웹 응용 프로그램에서 데이터 입력후 조류계산을 수행하기 위한 화면이다. 화면에서 보듯이 조류계산 수행에 필요한 옵션 등을 지정할 수 있다. 마찬가지로 조류계산 데이터에 대한 편집 등이 필요하다면 그에 해당하는 화면을 구성하여 데이터 편집 기능을 수행할 수 있으며 사용자의 요구에 맞도록 다양한 화면을 구성할 수 있다.

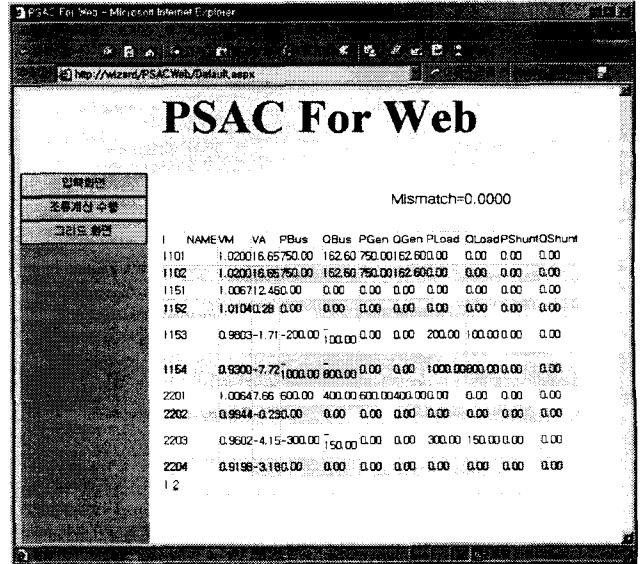


그림 9 웹용 응용 프로그램의 조류계산 결과 표시 화면  
Fig. 9 The result of the load flow of web application program

[그림 9]는 웹 응용 프로그램에서 조류계산을 수행한 후에 결과를 표시하는 화면이다. 본 프로그램에서는 데이터 표시 방법을 표의 형태로 하였지만 사용자의 요구에 따라 다양한 결과 출력 화면을 구성할 수 있다.

웹을 통해 응용 프로그램을 구성하는 경우에는 사용자가 응용 프로그램을 보유하고 있을 필요가 없다. 단지 응용 프로그램을 탑재하고 있는 웹서버에 접속함으로써 조류계산을 수행할 수 있게 된다. 이러한 경우에도 윈도우즈 응용 프로그램과 마찬가지로 해석 알고리즘과 사용자 화면을 독자적으로 개발하고 유지보수하는 것이 가능하다.

### 4. 결 론

최근의 프로그램 동향은 일반화된 바이너리 인터페이스를 디자인하고 그 인터페이스에 기반한 컴포넌트를 개발하여 사용자를 위한 응용프로그램을 개발할 때 바이너리 컴포넌트를 재가공없이 사용하도록 하는 것이다. 하지만 이 인터페이스는 개발된 환경에 따라 호환이 되지 않는 경우가 있다. 그러나 XML Web Service는 표준 기술을 사용하여 특정 환경에 구애 없이 사용이 가능하다.

이러한 XML Web Service를 사용함으로써 알고리즘 서버를 인터넷에 공유하여 계산 루틴을 필요로 하는 다양한 요구자가 알고리즘을 개발하지 않고, 서버를 액세스하여 개발 시간과 노력을 줄일 수 있으며 강력한 계산 서버로 계산 루틴을 옮기는 아키텍처를 제안함으로써 계산 속도를 향상시키고 다수의 서버를 사용한 분산 컴퓨팅을 하기 위한 발판 마련할

수 있다. 또한 각 해석 알고리즘과 응용 프로그램의 분리를 통하여 시스템의 유지보수 및 확장이 용이하게 된다.

하지만 XML Web Service는 인터넷 기반이므로 데이터의 보안과 연결의 신뢰도가 떨어진다. 또한 많은 데이터의 전송이 필요한 경우, 인터넷 속도에 따라 많은 시간이 걸릴 수 있다. 따라서 추후의 연구과제는 이러한 문제점을 보완하는 방향으로 진행되어야 할 것이다.

### 참 고 문 헌

- [1] Short, Scott, "Buinding XML Web Service for the Microsoft .NET Platform", Microsoft Press, 2002
- [2] Eric Newcomer, "Understanding Web Services: XML, WSDL, SOAP, and UDDI", Addison Wesley Professional, 2002
- [3] Andreas Eide, Chris Miller, Bill Sempf, Srinivasa Sivakumar, Mike Batongbacal, Matthew Reynolds, Mike Clark, Brian Loesgen, Robert Eisenberg, Brandon Bohling, Russ Basiura, Don Lee, "Professional ASP.NET Web Services", Wrox Press Inc., 2001
- [4] Richard L. Burden, J.Douglas Faires, "Numerical Analysis", ITP, 1997
- [5] Hadi Saadat, "Power System Analysis", WCB/McGraw-Hill, 1999

## 저 자 소 개



### 최 장 흠 (崔 章 欽)

1972년 5월 26일생. 1997년 충남대 공대 전기공학과 졸업. 1999년 동 대학원 전기공학과 졸업(석사). 현재 동 대학원 전기공학과 박사과정.

Tel : 042-821-7609 FAX : 042-823-7970  
E-mail : mu-sic@power.ee.cnu.ac.kr



### 김 건 중 (金 建 中)

1953년 2월 12일생. 1975년 서울대 공대 전기공학과 졸업. 1977년 동 대학원 전기공학과 졸업(석사). 1985년 동대학원 전기공학과 졸업(공박). 1977년 해군제2사관학교 교수. 현 충남대학교 전기공학과 교수

Tel : 042-821-5659 FAX : 042-823-7970  
E-mail : kjkim@ee.chungnam.ac.kr