

## 웹기반 단자유도 시스템 비선형 지진해석

# Web-Based Nonlinear Earthquake Analysis of SDOF Systems

박헌성\*

Park, Hunsung

신수봉\*\*

Shin, Soobong

---

### ABSTRACT

A web-based platform for nonlinear earthquake analysis on SDOF systems has been developed in the research. The proposed web-based platform is a 2-tier system composed of client and server sides. *Smart Client* is applied for the client side to improve the handling speed and *UI* compatible to the stand alone systems. For the server side, *MSSQL* is used as a storage for the web. *XML Webservice* is adopted for the networking between client and server. To examine the efficiency of the developed web-based platform, a trial study on a SDOF nonlinear structural system under earthquake excitation has been carried out.

---

### 1. 서 론

과거 10년보다 앞으로 10년간의 인터넷 발전은 그 잠재적 성장을 짐작할 수 없을 정도로 크다. 당장 지금의 현실만을 보더라도 인터넷을 통한 많은 어플리케이션들이 실생활과 관련되어 개발되어 있다. 비단 토목분야로 국한하더라도 인터넷 지도검색, GPS(global positioning system)를 이용한 mobile device의 navigation system, CM분야의 PMIS(project management information system)를 예로 들 수 있다. 이는 공간적 한계성의 극복이라는 인터넷의 특징을 잘 반영한 것으로 점차 다양한 분야로 확대되고 있고, 구조분야로도 MCEER의 Virtual Laboratory for Earthquake Engineering, 도로교통기술원의 DACS for Web등에서 서비스를 실시하고 있다. 웹기반의 플랫폼은 기존의 상용화된 범용 해석 프로그램인 SAP, MIDAS등의 stand alone형식이 가지는 한계점을 극복할 수 있는 대안이 될 것이다. 그 한계점은 크게 몇 가지로 구분할 수 있는데, end user의 관리가 사실상 불가능하기 때문에 모든 end user의 일관적인 update가 어렵다는 관리적인 측면과 초기 구입비등의 경제적 측면, 웹상의 유용한 정보들을 공유할 수 없는 활용적인 측면등을 예로 들 수 있다. 그러나 그 대안으로 부족함이 없기 위해서는 기존의 상용프로그램에서와 같은 처리속도와 UI(user interface)의 지원이 필수적이다. 처리속도를 위해서는 해석알고리즘을

---

\* 인하대학교 토목공학과, 석사과정

\*\* 정회원 · 인하대학교 토목공학과, 부교수

포함하는 business logic을 물리적으로 독립시켜 서버와 네트워크의 부하를 감소할 수 있는 2 tier 또는 3 tier의 구성이 필요하고, UI에서는 DHTML만으로 표현되던 web browser 기반 모델 보다는 windows 기반의 UI모델이 필요하다. 이에 본 연구에서는 첫째, Smart Client를 이용하여 stand alone과 같은 처리속도와 UI를 지원하면서 서버부하의 감소. 둘째, XML WebService를 이용하여 확장성, 재사용성의 고려와 네트워크 부하의 감소. 셋째, 저장소로써 MSSQL을 사용한 데이터베이스 서버의 구축으로 XML WebService의 지원하는 방식으로 웹 환경을 설정하였다. 개발된 플랫폼의 검증을 위해서 단자유도계의 비선형 지진해석을 수행하였으며 시험서비스를 실시 중이다.

## 2. 웹기반 플랫폼의 구성

서버 - 클라이언트 모델의 설정을 위해서는 서버와 네트워크 부하의 문제를 고려해야 한다. 이는 business logic의 물리적 위치관계와 전송되는 데이터의 조절을 통해서 그 해법을 찾을 수 있다. 과거와 달리 현재는 2 tier, 3 tier의 개념으로 business logic을 메인서버로부터 물리적으로 독립시킴으로 인해 서버 부하를 감소시키며, presentation logic을 클라이언트로 이양하면서 네트워크 부하의 문제를 최소화하고 있다. 이에 본 연구에서는 어셈블리 단위로 구성되는 Smart Client 기술을 이용하여 business logic, presentation logic을 클라이언트로 이양하는 2 tier 개념의 서버 - 클라이언트 모델을 기반으로 하고 XML WebService를 통하여 UI를 배제한 데이터만을 전송하도록 고려하였으며, XML WebService의 지원을 위해 MSSQL을 이용하여 데이터베이스 서버를 구축하였다. 아래의 그림은 본 연구의 총괄적인 서버 - 클라이언트의 구성관계를 나타낸다.

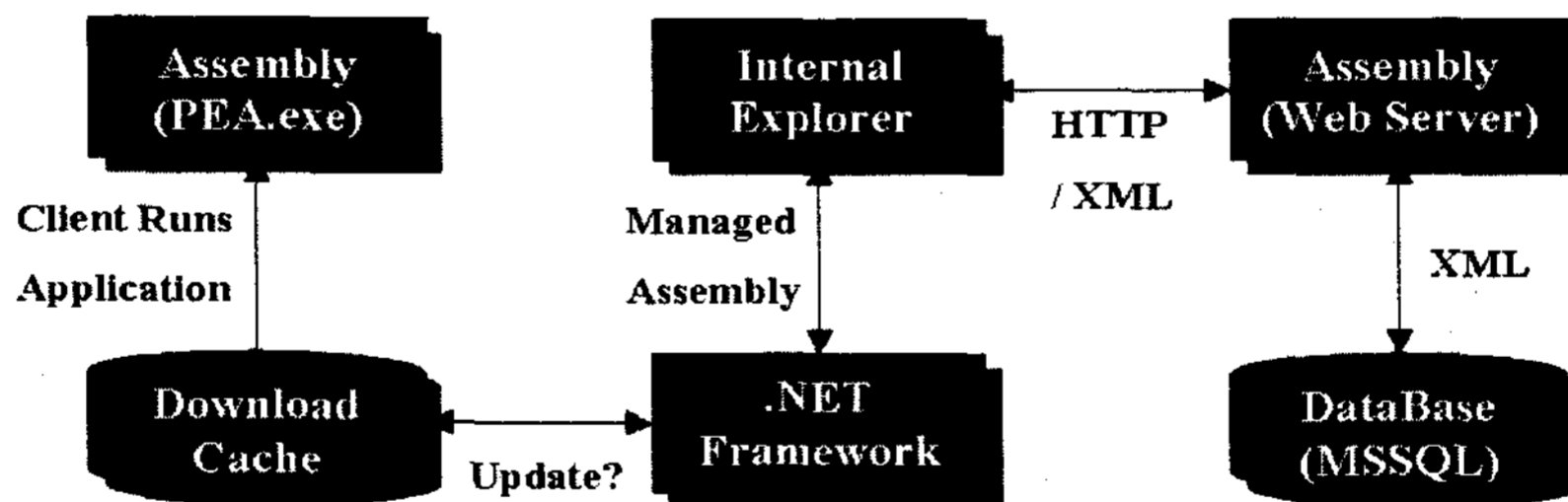


그림 1. 서버 - 클라이언트의 구성

### 2.1 서버 - 클라이언트 시스템

서버 - 클라이언트 시스템은 presentation logic, business logic, data access logic의 물리적 위치관계에 따라 세 가지 형태로 구분된다. 이는 초기 모델인 1 tier 시스템에서 파생되었으며 서비스 목적과 환경에 맞게 구성되어야 한다. 여기에는 서버 및 네트워크 부하의 문제를 고려하여야 하며 최근에는 사용자의 요구가 많아짐에 따라 2, 3 tier 시스템의 모델이 지배적이고 대용량의 서비스를 목적으로 하는 경우에는 3 tier 시스템으로 운용된다.

### 1 tier 시스템

- ▶ 한 시스템에 서버와 클라이언트의 프로그램이 공존
- ▶ 개발 시간이 적고, 관리와 보안이 쉬움
- ▶ 서버가 비대해지고 서버부하가 커짐

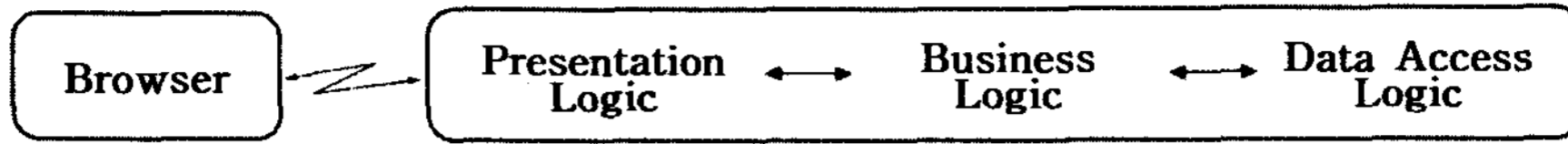


그림 2. 1 tier 시스템

### 2 tier 시스템

- ▶ 서버와 클라이언트의 물리적 독립
- ▶ 서버의 부하가 줄어들고, 사용자로의 정보제공 원활
- ▶ 클라이언트의 비대화, 과도한 데이터 요구에 의한 네트워크의 부하 증가

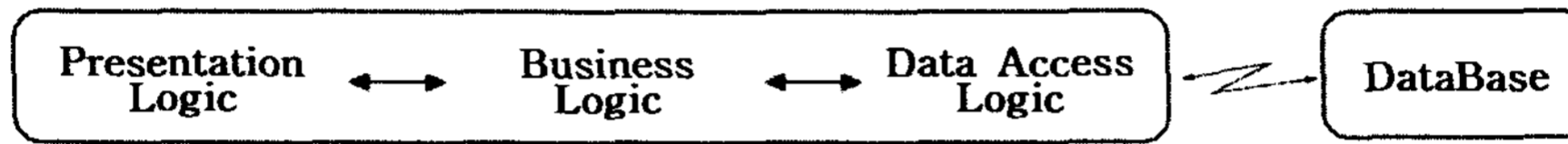


그림 3. 2 tier 시스템

### 3 tier 시스템

- ▶ middle tier를 통한 서버와 클라이언트의 기능 분산
- ▶ 서버와 클라이언트의 한쪽으로의 비대화 방지
- ▶ 초기 설치비용이 크고, 개발의 어려움 때문에 때로는 2 tier보다 성능저하

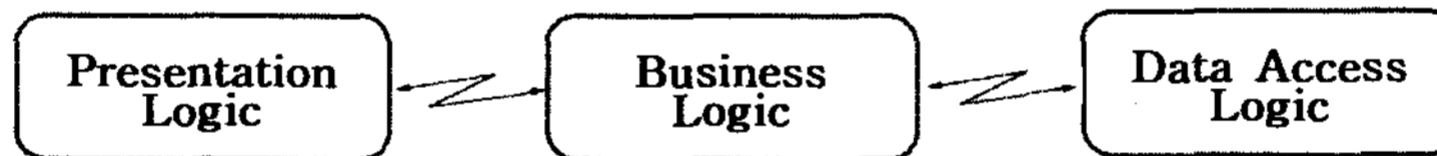


그림 4. 3 tier 시스템

구조해석의 경우 모델과 해석방법에 따라 그 과정이 매우 복잡해 질 수 있다. 때문에 해석을 담당하는 business logic을 통합 서버시스템(1 tier)이나 별도의 서버시스템(3 tier)을 두는 것은 바람직하지 못하다. 그렇기 때문에 최근의 PC의 성능을 감안하여 business logic을 클라이언트로 이양하는 2 tier 시스템을 기본적으로 채택하는 것이 유리하다. 하지만 business logic을 담당할 수 없는 모바일 계통의 클라이언트를 고려하기 위해서는 3 tier로의 확장도 염두에 두어야 한다.

## 2.2 Smart Client를 이용한 클라이언트 어플리케이션

Business logic의 클라이언트로의 이양을 위해서는 배포가 선행되어야 한다. 배포의 방법에는 CD나 기타 보조기억장치를 통한 배포와 웹을 통한 배포로 나눌 수 있고, 배포되는 응용프로그램은 크게 *Fat*, *Smart*, *Thin Client*의 세 가지로 구분되어 질 수 있다. 다음은 배포되는 응용프로그램의 종류에 따른 차이를 표로 나타낸 것이다.

표 1. 클라이언트의 종류에 따른 차이점

종류	UI	응답	관리	배포	연결상태	요구조건
<i>Fat</i>	상	상	중	중	off	레지스트리 등록
<i>Smart</i>	상	상	상	상	on / off	.NET Framework
<i>Thin</i>	중	환경에 따라	상	상	on	많은 전송량

웹을 통한 어플리케이션으로 초기 개발된 것은 *DHTML* 방식의 *Thin Client*였다. 하지만 클라이언트의 요구 조건이 많아짐으로 인해 *DHTML*만으로는 표현하기 어려운 부분이 있었고, 대체된 것이 *Active X Control*이었다. 실제로 *Active X*는 많은 어플리케이션에 사용되었으나 너무 쉬운 배포로 인해 보안에 문제가 발생해 *Smart Client*가 등장하게 되었다. 2 tier 시스템에서 *Smart Client*의 이용은 매우 편리하다. 어셈블리 단위로 클라이언트의 컴퓨터에 배포되어 사용되어지기 때문에 presentation logic, business logic, data access logic을 어셈블리에 포함시키는 방법으로 단순화하여 배포할 수 있기 때문이다. 배포된 어셈블리는 클라이언트 컴퓨터의 레지스트리에 등록되는 것이 아니기 때문에 "DLL Hell"이라 불리는 DLL 오류에 대해서도 안전하고, 서비스 공급자의 입장에서 단순히 제작된 어셈블리를 웹서버에 게시하기만 하면 클라이언트에서 버전 확인 후 업데이트된 버전일 경우 자동으로 다운되기에 배포에 관한 관리적 효율성과 재사용, 확장성이 매우 높다. 이는 모두 .NET Framework에 의해 제어되기 때문에 *Smart Client*의 이용에는 .NET Framework은 반드시 필요하나, 현재의 윈도우에서는 추가의 설치 작업이 필요하다. 하지만 윈도우 서버 2003을 포함한 이후의 버전에서는 기본적으로 탑재되므로 이 부분은 시간이 해결해 줄 것이다. .NET Framework는 *Smart Client*를 사용하기 위해 코드인증을 거쳐야 하는 과정을 통해 보안을 강화한다. 하지만 코드인증을 받기 위해서는 *Active X*를 이용하거나 약간의 설치작업을 필요로 하는 문제가 있다. 본 연구에서는 후자의 방법을 통하여 사용자가 코드인증 어플리케이션을 설치하는 과정으로 코드인증을 수행한다. 다음은 어셈블리 단위로 제작된 구조물의 모델링에 관한 어셈블리 조직도이다.

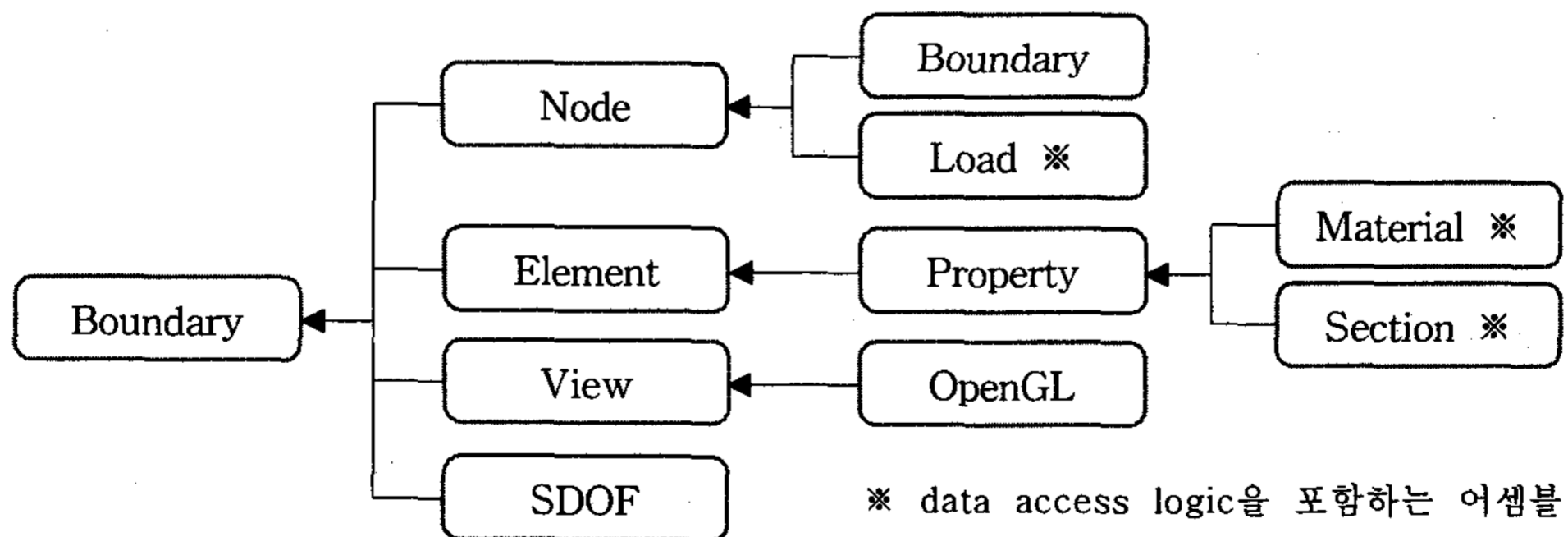


그림 5. 모델링 어셈블리 구조

### 2.3 MSSQL을 이용한 데이터베이스 서버

웹 어플리케이션의 경우 데이터베이스는 접근성, 보안성의 측면에서 파일처리보다 우수하다. 본 연구에서는 데이터베이스로서 관계형 데이터베이스인 *MSSQL*을 사용하여 *XML*을 지원에 원활하도록 하였다. 데이터베이스의 사용은 사용자 개개인의 모델링정보인 전용정보와 모든 사용자들이 사용가능한 공용정보를 이분화하여 제공함으로써 데이터의 휴대가 불필요하도록 하여 웹기반으로서의 이점을 높이고자 했다. 아래의 코드는 ID가 생성되었을 때 그에 따른 데이터베이스가 생성되는 코드를 나타낸다.

```
string sql = "Insert into UserIDTable(ID, PassWord, Name, Information)
            values('"+user+"', '"+pwd+"', '"+name+"', '"+info+"')";

try
{
    DBConn conn = new DBConn("server=***.***.***.***; uid=****; pwd=*****; database=UserID");
    conn.Open();
    conn.ExecuteNonQuery(sql);
    CreateDB_Table(user);
    conn.Close();
}
}
```

데이터베이스에서 데이터의 관리를 위해서는 *Data Provider*가 필요하고 본 연구에서는 *MSSQL*을 사용하기 때문에 *SQL Server .NET Data Provider*를 사용한다. *Data Provider*는 전송되는 정보의 양을 *DataSet*에 의해 조절하게 된다. 이는 정보의 요청을 매번 서버와의 접속을 통해서가 아닌 데이터 테이블 객체를 로컬 컴퓨터에 위치시키는 방법으로 네트워크 부하를 줄인다. 다음은 데이터베이스에 초기 접속시 pooling process의 개략도를 나타낸다.

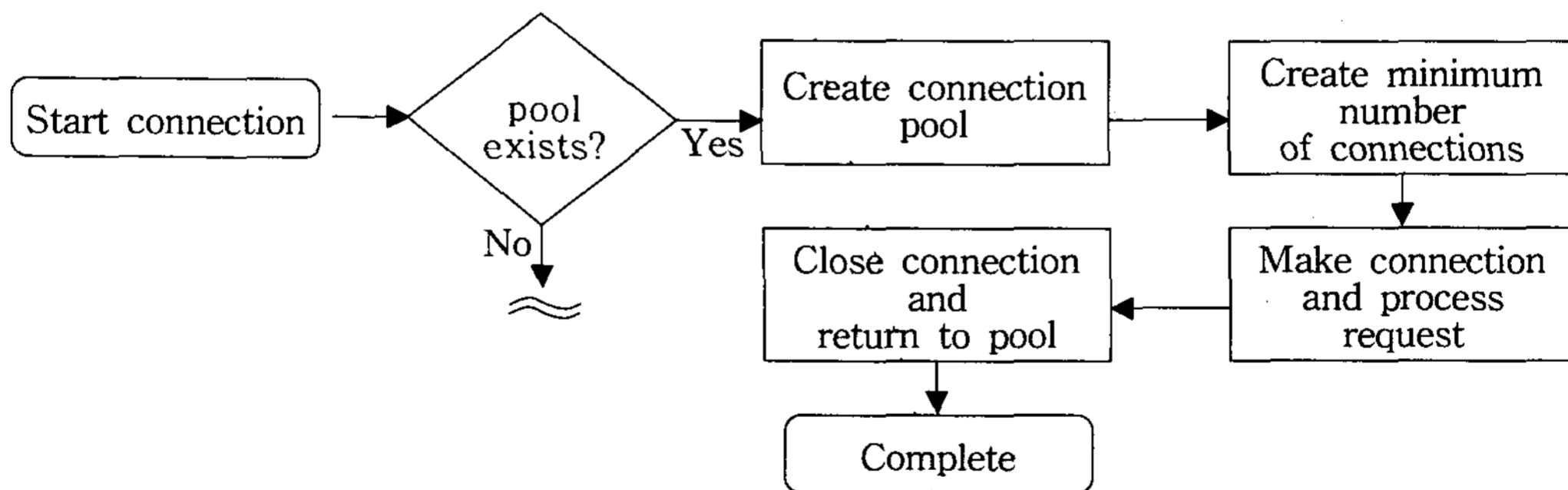


그림 6. pooling process 개략도

### 2.3 XML WebService를 이용한 네트워크

웹서버는 *Smart Client*의 배포와 클라이언트가 요청한 정보의 제공을 담당한다. 그렇기 때문에 웹서버는 항상 클라이언트의 요구에 응답할 준비가 되어 있어야 한다. 클라이언트가 요청한 정보

는 데이터베이스 서버의 내용일수도 있고, 기타 웹의 *UDDI(Service Broker)*에 게시된 내용일 수도 있다. 그러나 요청한 정보가 무엇이든 간에 웹을 통한 전송을 *SOAP* 프로토콜을 통한 *XML WebService*가 담당됨으로써 첫째, *UI* 정보의 전송을 배제한 오로지 데이터만의 전송으로 네트워크 부하를 줄일 수 있다는 것과 둘째, *business logic*을 포함하는 *middle tier*의 역할도 수행할 수 있다는 이점을 얻을 수 있다. 후자의 경우 모바일 클라이언트로의 확장이나 클라이언트 영역의 비대화를 방지하는 수단으로도 사용될 수 있다. 또한 *XML WebService*는 이종시스템을 지원하기 때문에 클라이언트의 환경에 제약을 받지 않는다. 이는 클라이언트의 종류에 상관없이 같은 정보를 제공해 줄 수 있음을 의미한다. 비록 본 연구에서는 *Smart Client*를 이용하여 특정한 클라이언트의 환경하에서만 서비스를 실시하고 있으나 차후에 있을지 모를 다른 환경의 클라이언트 확장까지도 고려될 수 있다.

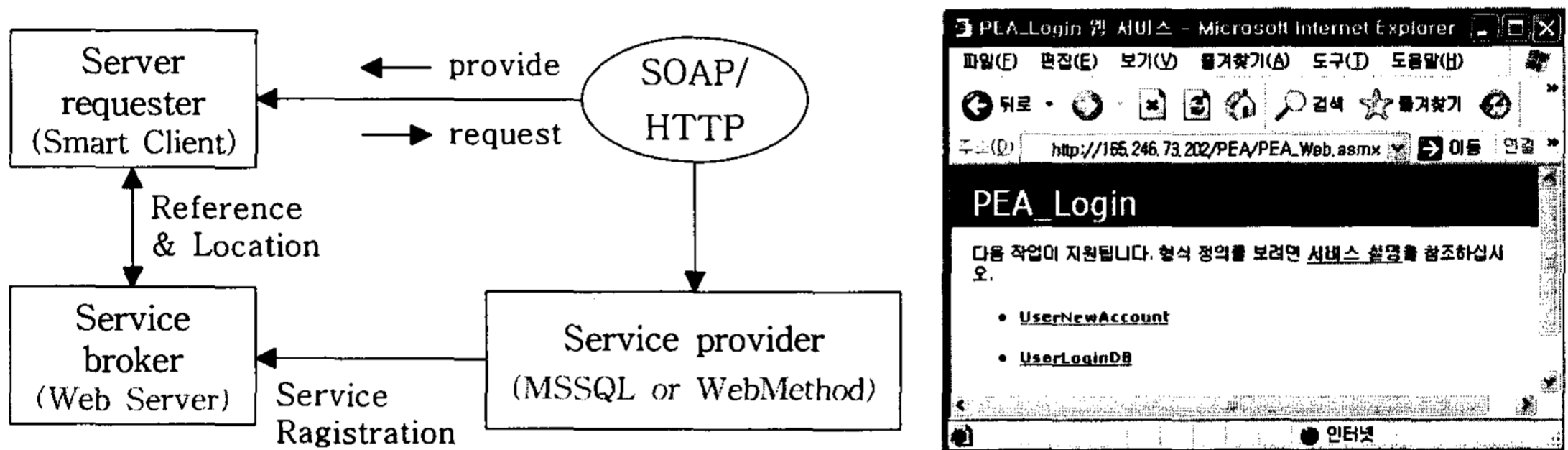


그림 7. XML WebService의 구성과 UDDI에 등록된 웹 메소드

## 2.4 단자유도계 지진해석으로의 적용

개발된 웹기반 플랫폼으로 단자유도계의 비선형 지진해석에 적용하였다. 완전 탄소성 재료로 가정하였고 Newmark- $\beta$  방법을 이용하여 시간영역에서의 응답을 계산하였다. 적용된 식은 다음과 같다.

$$\begin{aligned} \dot{u}_{n+1} &= \dot{u}_n + h(1 - \gamma)\ddot{u}_n + h\gamma\ddot{u}_{n+1} \\ u_{n+1} &= u_n + h\dot{u}_n + h^2\left(\frac{1}{2} - \beta\right)\ddot{u}_n + h^2\beta\ddot{u}_{n+1} \end{aligned} \quad \text{with } \gamma = \frac{1}{2}, \beta = \frac{1}{6} \quad (1)$$

여기서,  $h$  = 시간증분.

표 2와 같은 구조물에 대한 데이터와 지반 가속도를 입력후 해석을 실시하게 되는데, 클라이언트의 PC에 다운로드된 어셈블리에 의해 해석이 수행된다. Material, section, ground acceleration의 데이터는 메인프레임에서, 기타 정보는 단자유도계 프레임에서 입력받는 형식으로 메인프레임의 데이터는 재사용이 가능하도록 개발하였다. 표 2와 그림 8은 예제의 입력데이터와 인터페이스를 나타낸다.

표 2. 예제 입력데이터

Spring constant	12.35 <i>Kip/in</i>
Damping coefficient	0.274 <i>Kip · sec/in</i>
Mass	0.2 <i>Kip · sec<sup>2</sup>/in</i>
Max. restoring force(tension)	157.37 <i>Kip</i>
Max. restoring force(compression)	-157.37 <i>Kip</i>
Natural period	0.8 sec
Time step	0.01 sec

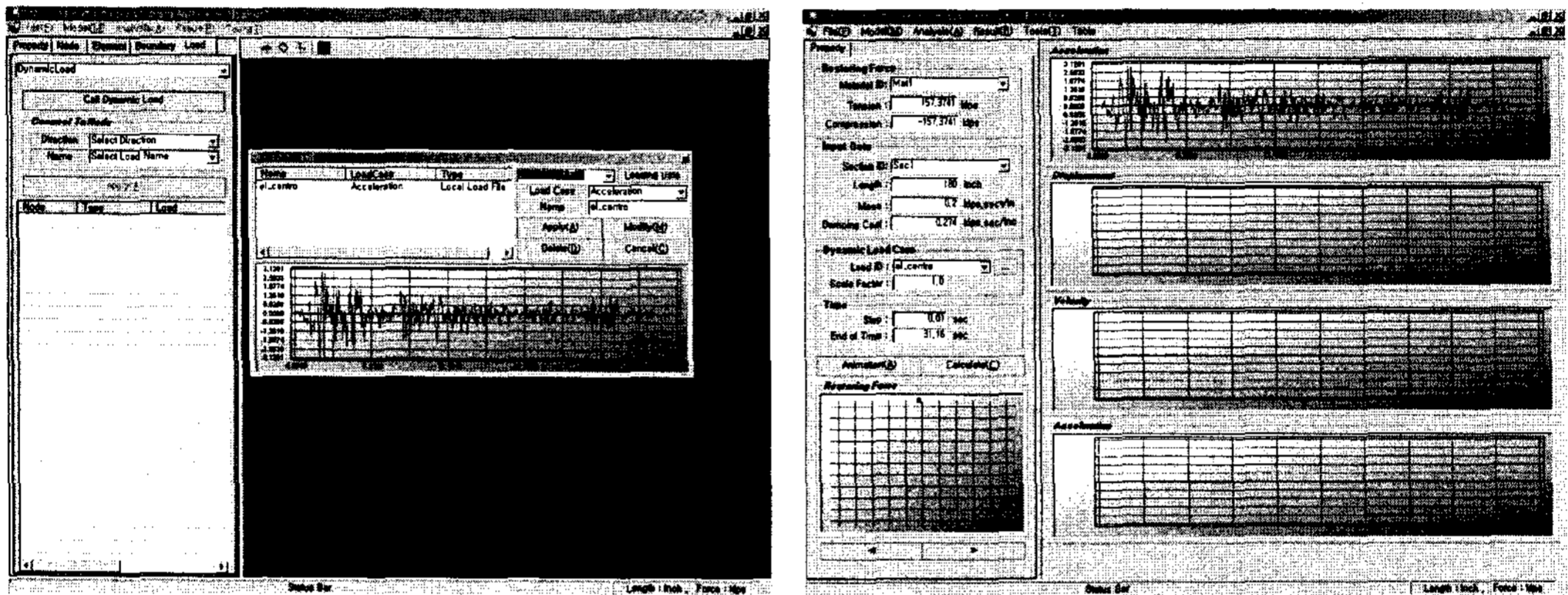


그림 8. 예제 입력화면

예제에 대한 결과는 그림 9에서와 같이 변위, 속도, 가속도, 복원력에 대한 시간이력 그래프와 표 형태의 텍스트로 확인할 수 있다. 시간증분, 지반가속도 또는 동하중의 스케일의 변화가 가능하도록 하여 사용자 입장에서 입력데이터의 불필요한 반복없이 쉽게 결과를 확인할 수 있도록 설정하였다. 입력된 구조물 데이터는 데이터베이스 서버에 저장되어 파일의 휴대없이 재사용이 가능하도록 하여 웹의 이점을 활용할 계획이다.

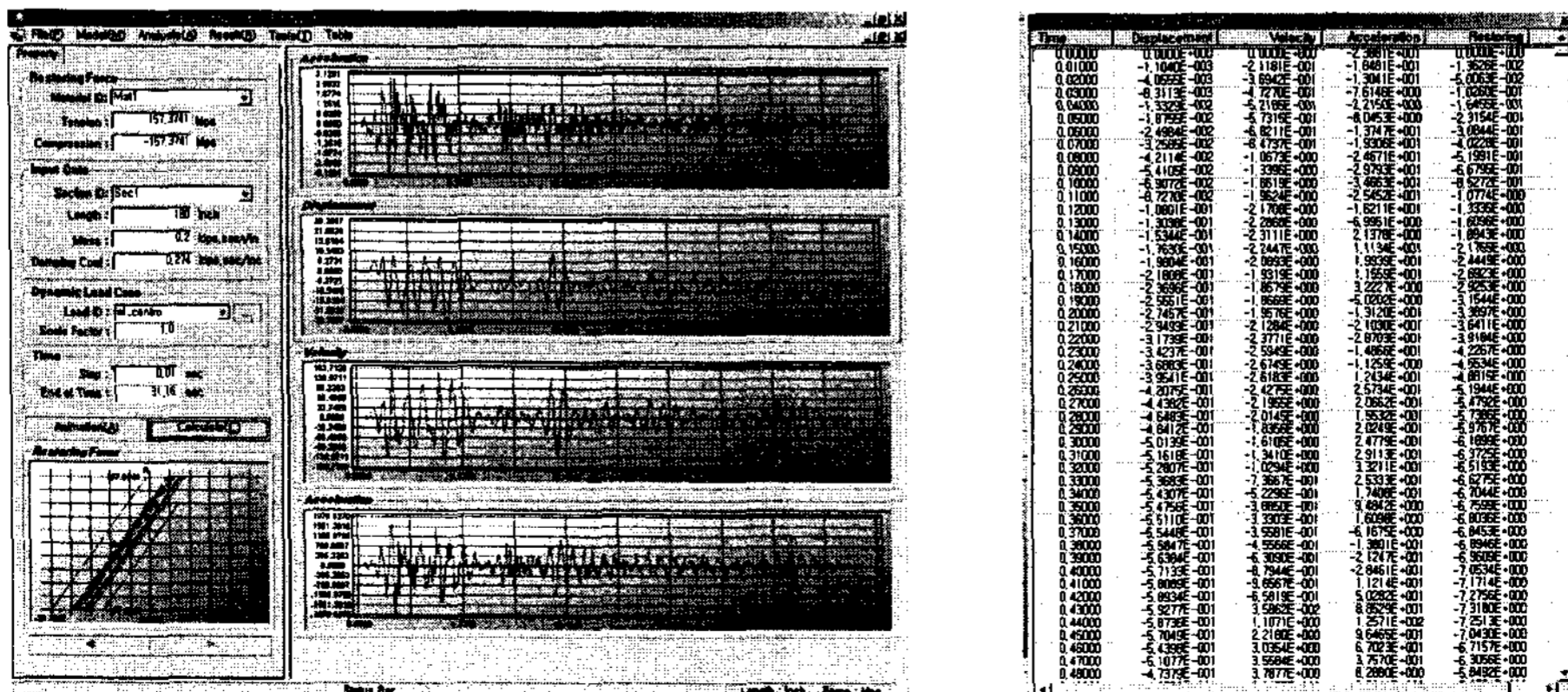


그림 9. 예제 결과화면

### 3. 결론

본 연구에서는 단자유도계의 비선형 동해석을 수행할 수 있는 웹기반 플랫폼을 개발하였다. 웹기반의 플랫폼은 그동안 로컬영역에서만 수행되던 기존 상용프로그램의 제약인 사용자관리와 통합적인 웹상의 정보공유 문제를 극복할 수 있는 좋은 대안이 될 수 있다. 실제로 새로운 버전의 배포, 웹상의 정보를 이용하기 위해서 별도의 플랫폼을 필요로 한다는 것은 시간, 경제적인 재투자를 해야 한다는 점에서 이롭지 못하다. 하지만 웹기반의 플랫폼이 기존 상용프로그램과 같은 수준의 서비스를 위해서는 처리속도, UI 그리고 서버 및 네트워크 부하를 고려하여야 한다. 따라서 본 연구에서는 *Smart Client*를 이용함으로써 처리속도, UI, 서버부하의 문제를 해소하고, *XML WebService*를 이용하여 네트워크 부하의 감소를 고려하였으며, 데이터베이스를 사용함으로써 사용자 편의의 웹 환경을 구축하였다. 현재 개발된 플랫폼은 적용범위가 한정되어 있으므로 해석모델의 다양화, *UDDI*에 등록된 웹기법의 사용을 통해 보다 일반적인 형태로의 확장이 필요하다.

### 감사의 글

본 연구는 서울대학교 지진공학연구센터(KEERC)를 통한 한국과학재단 우수연구센터 지원금에 의해 수행되었으며, 이에 감사드립니다.

### 참고문헌

1. ALTools : [www.altools.co.kr](http://www.altools.co.kr).
2. INAVI : [www.inavi.co.kr](http://www.inavi.co.kr).
3. 창해소프트 : [www.css.co.kr](http://www.css.co.kr).
4. 도로교통기술원 : <http://research.freeway.co.kr>, 한국도로공사.
5. Virtual Laboratory for Earthquake Engineering : <http://cee.uiuc.edu/sstl/java/>.
6. MIDAS : [www.midasuser.com](http://www.midasuser.com), MIDAS Information Technology CO.Ltd.
7. SAP : [www.csiberkeley.com](http://www.csiberkeley.com), Computer & Structures Inc.
8. 박성준, "C# & .NET Programming Bible, 영진닷컴", 2003.
9. 김유철, "Smart Client Programming with Windows Forms", MSDN Webcast, <http://www.microsoft.com/Korea/MSDN/training/webcasts>, 2004.
10. 정원혁, "SQL Server 2000, 대림", 2001.
11. 정성태, "IE에서 닷넷 스마트 클라이언트 개발", 마이크로 소프트웨어, 2004.
12. Richer, J. "Applied Microsoft .NET Framework, Microsoft", 2002.
13. Williams, M. "Visual C# .NET, Microsoft", 2002
14. Lee, S. H., Jeong, D. G. and Kim, B. G., "강교량정보의 웹서비스를 위한 3차원 형상정보 운 영모델 개발", 대한토목학회 정기학술대회, 2004
15. 민경섭, "클라이언트/서버 시스템 구조의 발전 과정", 서울대학교
16. 하성광, "한글 비주얼 C#, 대림", 2002
17. 윤홍란, "Web Service & Base Technologies", 숙명여자대학교, 2002
18. Mario Paz, "Structural Dynamics, Chapman & Hall", 1997