

# 웹기반 기술을 사용한 구조해석

## Structural Analysis through Web-based Technology

박 헌 성\*  
Park, Hunsung

신수봉\*\*  
Shin, Soobong

### ABSTRACT

A web-based platform for structural analysis has been developed. The proposed web-based platform is a 2-tier system composed of client and server sides to reduce the overburden on the server side. *Smart Client* is applied for the client side to improve the handling speed and *UI*. For the server side, *MSSQL* is applied to deal with database and used as a storage for the web. *XML WebService* is adopted for the networking between the client and the server. To examine the efficiency of the developed web-based platform, a sample study is carried out with a nonlinear earthquake analysis on a SDOF system.

### 1. 서 론

인터넷의 발전은 신속한 정보의 공유를 의미한다. 위에 언급한 정보에는 텍스트기반의 정보뿐만 아니라 활용가능한 프로그램의 공유등 광범위한 정보를 의미한다. 이는 시공간적인 제약을 최소화하는 인터넷의 이점을 잘 반영하는 것으로, 개인사용자 뿐만 아니라 많은 산업분야 역시 인터넷의 활용도는 매우 크다. 현재 여러 분야에서 수많은 어플리케이션들이 개발되어 있고, 구조공학 분야에서도 *MCEER*의 *Virtual Laboratory for Earthquake Engineering*<sup>(1)</sup>, 도로교통기술원의 *DACS for Web*<sup>(2)</sup>등에서 서비스를 실시하고 있으나 그 적용범위가 제한적이라 활용도는 미비하다. 또한 최근에는 구조해석 정보와 모델링 정보를 포함하는 데이터베이스의 구축을 통한 e-business의 도입으로 협업의 효율성을 높이고자 하는 연구가 진행중이다.<sup>(3)</sup> 하지만 이것 역시 구조해석 정보를 지원할 뿐 실제로 해석과정을 수행하는 것은 아니다. 구조해석적 관점에서 볼 때, *MCEER*의 가상실험실<sup>(1)</sup>, *DACS for Web*<sup>(2)</sup>과 같은 웹기반의 어플리케이션들은 stand alone형식으로 개발된 상용프로그램들이<sup>(4),(5)</sup> 가지는 update의 관리적, 초기 구입비의 경제적 측면에서의 한계를 극복하는 대안이 될 수 있을 뿐만 아니라 이상호 등(2004)<sup>(3)</sup>에서와 같은 데이터베이스 구축으로 웹상의 유용한 정보의 공유를 통한 활용성을 고려할 수 있다. 그러나 그 대안으로 부족함이 없기 위해서는 기존의 상용프로그램에서와 같은 처리속도와 *UI*(user interface)의 지원이 필수적이다. 구조해석의 특성상 처리속도를 위해서는 해석알고리즘을 포함하는 business logic을 서버와 물리적으로 독립시켜 서버와 네트워크의 부하를 감소할 수 있는 2

\* 인하대학교 토목공학과, 석사과정

\*\* 정회원 · 인하대학교 토목공학과, 부교수

tier 또는 3 tier의 구성이 필요하고, UI에서는 DHTML만으로 표현되던 web browser 기반 모델 보다는 windows 기반의 UI모델이 필요하다. 또한 웹 저장소로써 데이터베이스를 사용하여 정보의 공유가 원활하도록 해야 한다. 본 연구에서는 첫째, 다수 사용자의 접속시에도 서비스가 원활하도록 구조해석에 적합한 클라이언트 / 서버 모델을 설정하고 둘째, 클라이언트의 배포 프로그램으로 Smart Client를 사용하여 stand alone형 어플리케이션과 같은 처리속도와 UI를 지원하며 셋째, XML WebService를 이용하여 확장성, 재사용성을 고려하고 네트워크 부하를 감소시키며 넷째, 저장소로써 MSSQL을 사용한 데이터베이스 서버를 구축하여 XML WebService를 지원하는 방식으로 웹 환경을 설정하였다. 본 연구에서는 단자유도계의 비선형 지진해석에 상기한 웹기반 기술을 접목하여 수행하였으며, 본 논문에는 단자유도계의 비선형 지진해석을 수행한 결과를 수록하였다.

## 2. 클라이언트 / 서버 모델의 구성

클라이언트 / 서버 모델의 설정을 위해서는 서버와 네트워크 부하의 문제를 고려해야 한다. 이것은 클라이언트 / 서버의 물리적 위치 설정과 전송되는 데이터의 조절을 통해서 그 해법을 찾을 수 있다. 구조해석의 특성상 해석모델과 알고리즘에 따라 많은 연산과정을 필요로 할 수 있다. 때문에 다수 사용자에게 서비스를 위해서는 해석과정을 포함하는 business logic을 서버에서 수행하는 것은 서버의 부하를 초래하므로 현재의 사용자 PC성능을 고려하여 business logic을 클라이언트로 이양하는 2 tier 또는 별도의 business logic 서버를 두는 3 tier, N-tier방식의 구성은 그러한 문제를 해결해 줄 수 있다.<sup>(6)</sup>

Table 1. Client / Server system architecture

1 tier	
	<ol style="list-style-type: none"> <li>1) 한 시스템에 서버와 클라이언트의 프로그램이 공존</li> <li>2) 개발 시간이 적고, 관리와 보안이 쉬움</li> <li>3) 서버가 비대해지고 서버부하가 커짐</li> </ol>
2 tier	
	<ol style="list-style-type: none"> <li>1) 서버와 클라이언트의 물리적 독립</li> <li>2) 서버의 부하가 줄어들고, 사용자로의 정보제공 원활</li> <li>3) 클라이언트의 비대화, 과도한 데이터 요구에 의한 네트워크의 부하 증가</li> </ol>
3 tier	
	<ol style="list-style-type: none"> <li>1) middle tier를 통한 서버와 클라이언트의 기능 분산</li> <li>2) 서버와 클라이언트의 한쪽으로는 비대화 방지</li> <li>3) 초기 설치비용이 크고, 개발의 어려움 때문에 때로는 2 tier보다 성능저하</li> </ol>

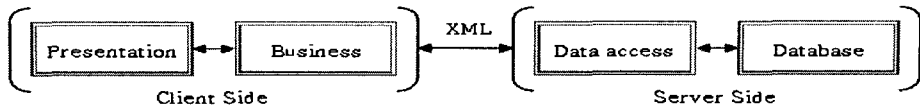


Fig 1. Client / Server system architecture for this study

본 연구에서는 Fig 1 에서와 같은 클라이언트 / 서버 시스템을 구성한다. 기본적으로 2 tier system으로 서버 부하를 고려하였으며, data access logic을 서버에 위치하고 XML webservice로 데이터의 전송을 지원하는 구성으로 네트워크 부하를 감소시켰다.<sup>(7)</sup>

### 2. 1. Smart Client를 이용한 클라이언트 어플리케이션

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

Table 2. Difference as a kinds of client(클라이언트의 종류에 따른 차이점)

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

기존의 상용프로그램은 Fat Client를 사용함으로써 처리속도와 UI를 지원하였다. 하지만 Fat Client는 stand alone 형식에 적합하기 때문에 웹기반 환경에서 Fat Client와 같은 효율성을 갖기 위해서는 표 2에 나타난 바와 같이 Smart Client가 요구된다. 초기 웹 어플리케이션은 DHTML 방식의 Thin Client였으나 인터넷 페이지의 제약으로 인해 윈도우 기반의 인터페이스를 가능하게 하는 Active X Control이 등장했다. 하지만

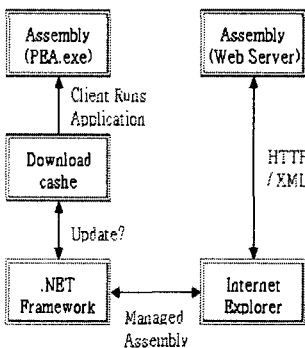


Fig 2. Smart Client

Active X Control은 너무 쉬운 배포로 인하여 보안문제가 발생하였고 대체된 것이 Smart Client이다. 2 tier 시스템에서 Smart Client의 이용은 매우 편리하다. 어셈블리단위로 클라이언트의 컴퓨터에 배포되어 사용되어지기 때문에 컴포넌트 베이스로 개발된 presentation logic, business 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 단순하게 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 Framework*에 의해 제어되기 때문에 *Smart Client*의 이용에는 *.NET Framework*은 반드시 필요하다. 현재의 윈도우에서는 추가의 설치 작업이 필요하다. 하지만 윈도우 서버 2003을 포함한 이후의 버전에서는 기본적으로 탑재되므로 이 부분은 시간이 해결해 줄 것이다.<sup>(8)</sup> *.NET Framework*는 *Smart Client*를 사용하기 위해 코드인증을 거쳐야 하는 과정을 통해 보안을 강화한다.<sup>(9)</sup> 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 모델링과 뷰어 어셈블리 조직도이다.

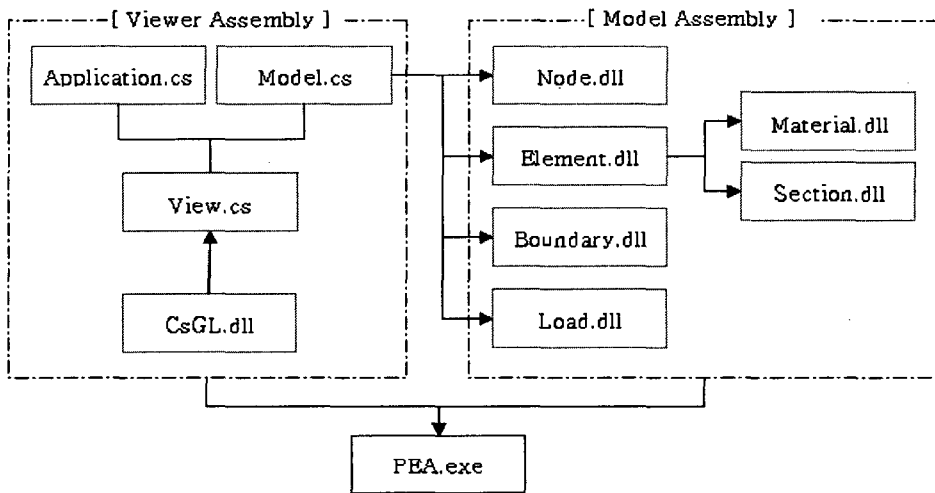


Fig 3. Structure of modeling and viewer assembly

## 2.2 MSSQL을 이용한 데이터베이스 서버

웹 어플리케이션의 경우 데이터베이스는 접근성, 보안성의 측면에서 파일처리보다 우수하다.<sup>(3)</sup> 2 tier 시스템에서는 일반적으로 data access logic이 클라이언트에 위치하나 본 연구에서는 data access logic을 서버에 위치하고 XML을 통해 데이터를 전송하게 된다. 이것은 불필요한 데이터의 전송을 감소함으로써 네트워크 부하를 줄일 수 있다. 따라서 XML지원에 원활한 MSSQL로 데이터베이스를 구축하여 최적의 서비스 시스템을 이루었다. 구축된 데이터베이스에는 사용자 개개인의 모델링정보인 전용정보와 모든 사용자들이 사용가능한 공용정보를 이분화함으로써 정보의 공유가 가능하면서 보안을 강화하도록 했다.

데이터베이스에서 데이터의 관리를 위해서는 *Data Provider*가 필요하고 본 연구에서는 MSSQL을 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 데이터베이스 서버 부하를 줄인다. 다음은 데이터베이스에 초기 접속시 pooling process의 개략도를 나타낸다.

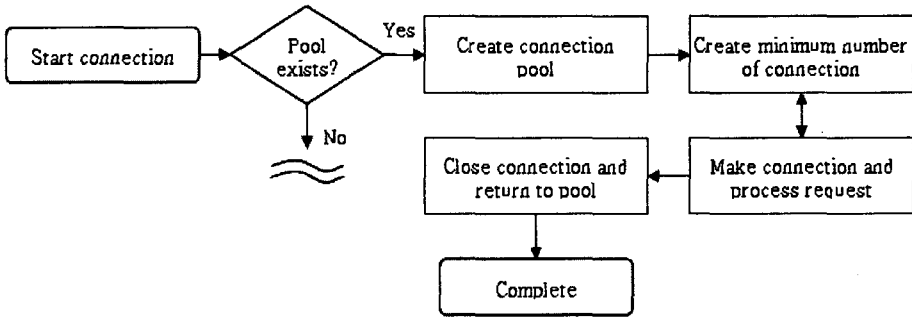


Fig 4. Connection of pooling process<sup>(10)</sup>

### 2.3 XML WebService를 이용한 데이터 전송

웹서버는 *Smart Client*의 배포와 클라이언트가 요청한 정보의 제공을 담당한다. 그렇기 때문에 웹서버는 항상 클라이언트의 요구에 응답할 준비가 되어 있어야 한다. 클라이언트가 요청한 정보는 데이터베이스 서버의 내용일 수도 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 것<sup>(7)</sup>과 둘째, *business logic*을 포함하는 *middle tier*의 역할도 수행할 수 있다는 이점을 얻을 수 있다. 후자의 경우는 3 tier, N-tier로의 확장을 의미하며, 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.

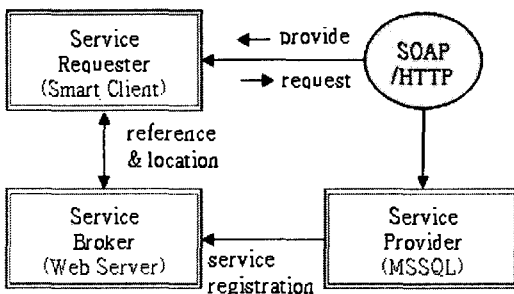


Fig 5. Architecture of XML WebService<sup>(11)</sup>

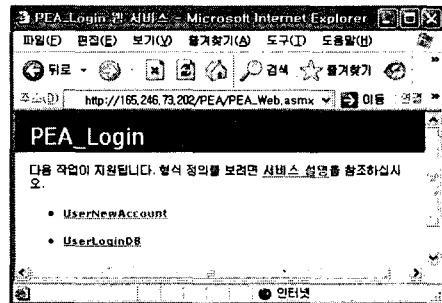


Fig 6. Web method at UDDI

## 2.4 단자유도계 지진해석으로의 적용

개발된 웹기반 플랫폼의 검증을 위해 단자유도계의 비선형 지진해석에 적용하였다. 완전 재료로 가정하였고 Newmark- $\beta$  방법을 이용하여 시간영역에서의 응답을 계산하였다. 적용된 식은 다음과 같다.<sup>(12)</sup>

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

여기서,  $h$  = 시간증분.

표 3과 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 3과 그림 7은 예제의 입력데이터와 인터페이스를 나타낸다.

Table 3. Input data for sample study

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

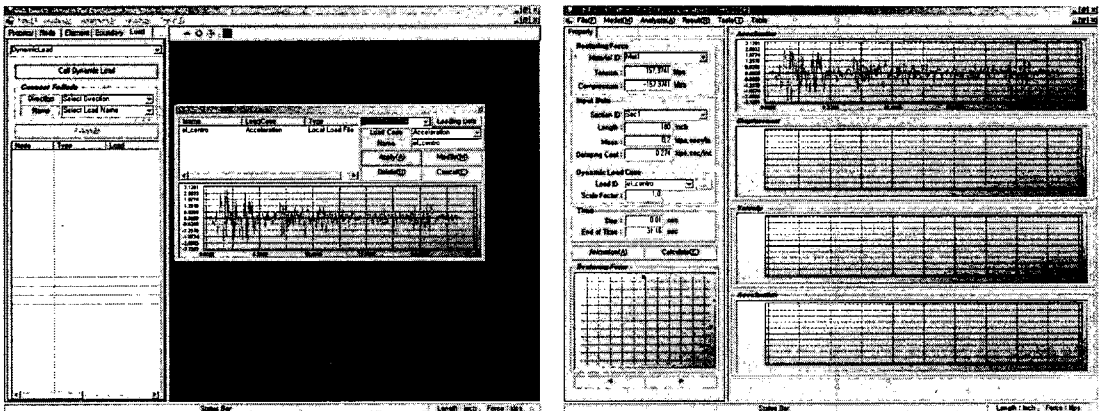


Fig 7. Input interface for sample study

예제에 대한 결과는 그림 8에서와 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.

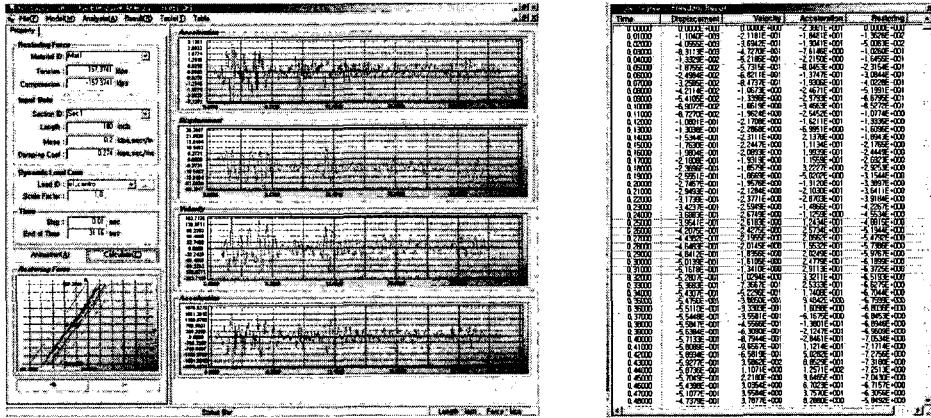


Fig 8. Output interface for sample study

### 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 구조해석의 특성에 가장 적합한 클라이언트 / 서버모델을 2 tier 시스템으로 설정하여 business logic, presentation logic을 클라이언트로 위임함으로써 처리속도의 문제와 서버부하의 문제를 해결하였다. 또한 Smart Client를 이용함으로써 UI, 보안의 문제를 해소하고, XML Webservice를 이용하여 네트워크 부하의 감소를 고려하였으며, 데이터베이스를 사용함으로써 사용자 편의의 웹 환경을 구축하였다. 현재 개발된 플랫폼은 적용범위가 한정되어 있으므로 해석모델의 다양화, UDDI에 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.

## 참고 문헌

1. Virtual Laboratory for Earthquake Engineering : <http://cee.uiuc.edu/sstl/java/>
2. 도로교통기술원 : <http://research.freeway.co.kr>, 한국도로공사
3. 이상호, 정동균, 김봉근, “강교량정보의 웹서비스를 위한 3차원 형상정보 운영모델 개발”, 대한토목학회 정기학술대회, 2004
4. MIDAS : [www.midasuser.com](http://www.midasuser.com), MIDAS Information Technology Co.Ltd.
5. SAP : [www.csiberkeley.com](http://www.csiberkeley.com), Computer & Structures Inc.
6. 류형규, 이순천, 류시원, 신성호, UML 기반 객체지향 클라이언트 / 서버 구축, 홍릉과학출판사, 서울 2000, p.455
7. 김유철, “Smart Client Programming with Windows Forms”, MSDN Webcast, <http://www.microsoft.com/Korea/MSDN/training/webcasts>, 2004.
8. 정성태, “IE에서 닷넷 스마트 클라이언트 개발”, 마이크로 소프트웨어, 2004.
9. Richer, J. “Applied Microsoft .NET Framework, Microsoft”, 2002.
10. M. Williams, Visual C# .NET, *Microsoft*, Vol. 1, 2002.
11. 윤홍란, “Web Service & Base Technologies”, 숙명여자대학교, 2002
12. Mario Paz, Structural Dynamics, Chapman & Hall, New York, 1997, p824