

---

# Web Services 상호 운용성을 위한 테스트 프레임워크 구축 및 시뮬레이션

임산송\* · 이원석\*\* · 이강찬\*\* · 전종홍\*\* · 이승윤\*\* · 정희경\*

## A Test Frameworks and Simulation for Web Services Interoperability

San-Song Im\* · Won-Suk Lee\*\* · Kang-Chan Lee\*\* · Jong-Hong Jeon\*\*

Seung-Youn Lee\*\* · Hoe-Kyung Jung\*

### 요 약

최근 많은 기업들이 플랫폼 또는 프로그래밍 언어에 의존적이지 않고 상호간의 원활한 정보 교환을 지원하는 HTTP와 같은 인터넷 표준 프로토콜과 XML 표준 기술 기반의 새로운 분산 컴퓨팅 환경인 웹서비스를 도입하고 있지만, 각각의 업체에서 제공하는 플랫폼들이 지원하는 표준들의 버전 불일치 문제로 인해 상호 운용성에 어려움을 겪고 있다.

이러한 상호 운용성 문제를 해결하기 위해 WS-I 에서는 기본 프로파일 및 사용 시나리오를 제공하는 등 일련의 표준화 작업을 진행하고 있지만, 아직 국내에서는 관련 연구가 미흡한 상태이다.

이에 본 논문에서는 주위에서 쉽게 접할 수 있는 도서 정보 검색 서비스를 서로 다른 플랫폼을 기반으로 웹서비스를 개발하여 상호 운용성 테스트 프레임워크를 구축하고 WS-I 기본 프로파일을 적용하여 상호 운용성을 시뮬레이션 및 검증하였다.

### ABSTRACT

Recently many corporations are introducing web services which supports mutual smooth information exchange without depending on platform or programming language and new distributed computing environment based on XML standard technology and internet standard protocol such as HTTP. However they are suffering difficulty in interoperability because of the problem of version inconsistency of standards supported by platforms that each vender offers. To solve these interoperability problem, although WS-I is progressing a consecutive standardization work by method of that offer Basic Profile and Usage Scenario, it is in the state that the domestic research connected with this is not sufficient yet.

Accordingly, this treatise constructed interoperability Test Framework by developing web services on the basis of different platform from books information retrieval services that can easily be come with and simulated and verified the interoperability by applying WS-I Basic Profile.

### 키워드

웹서비스, 상호 운용성, WS-I, XML, SOAP, WSDL, UDDI

### 1. 서 론

인터넷 관련 기술이 발전하고 웹의 이용이 급속하게 증가함에 따라 웹을 기반으로 하는 다양한 어플리케이션들이 생겨나게 되었지만 각각의 플랫폼 또는 프로그래밍 언어에 의존적이고 상호간의 원활한 정보 교환을 지원하지 않는 등의 문제로 인하여 공개 표준 기술 기반의 새로운 분산컴퓨팅 환경으로 웹서비스가 등장하였다.

그러나 웹서비스 또한 각 플랫폼들에서 지원하는 표준들의 버전 불일치 문제로 이중 플랫폼으로 구축된 웹서비스 객체 간의 상호 운용에 어려움을 겪고 있다. 이를 해결하기 위해 Microsoft, IBM, Sun Microsystems, BEA 등 유명 웹서비스 플랫폼 벤더들을 주축으로 웹서비스 상호 운용성 표준화 기구인 WS-I(Web Services-Interoperability Organization)를 설립하여 표준화 작업을 진행하고 기본 프로파일(Basic Profile)을 제정하여 배포하고 있다[1,2]. 그리고 아직 국내에서는 웹서비스 상호 운용성에 관련된 연구가 미흡한 실정이다.

이에 본 논문에서는 이중 플랫폼간의 웹서비스 상호 운용성을 테스트 및 검증하기 위하여 WS-I 기본 프로파일을 분석하고, 주위에서 쉽게 접할 수 있는 도서정보검색 서비스를 이용하여 .NET 기반의 웹서비스 제공자와 Java 기반의 웹서비스 요청자 환경을 구현하고 WS-I에서 제공하는 테스트 도구를 이용하여 테스트 프레임워크를 구축하고 시뮬레이션 및 결과를 분석하였다.

### II. 관련연구

#### 2.1 웹서비스 아키텍처

웹서비스는 HTTP와 같은 인터넷 표준 프로토콜과 XML Schema, SOAP, WSDL, UDDI 등의 XML 관련 표준 기술들을 기반으로 서비스 제공자(Service Provider), 서비스 레지스트리(Service Registry), 서비스 요청자(Service Requester)로 구성된 소프트웨어 컴포넌트들이 서로 느슨하게 결합되어 있는 새로운 분산 컴퓨팅 환경이다[3,4].

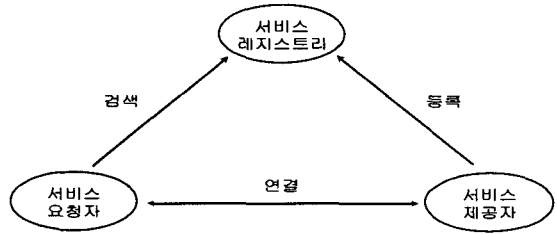


그림 1. 웹서비스 아키텍처  
Fig. 1 Web Services Architecture

그림 1은 웹서비스의 기본 구조로써 웹서비스가 동작하는 원리를 개념적으로 보여주고 있다. 서비스 제공자는 자신이 제공하는 서비스에 관련된 정보를 서비스 레지스트리에 등록하고 서비스를 이용하고자 하는 서비스 요청자는 서비스의 정보를 수집하기 위해 인터넷 디렉토리 시스템을 사용하여 원하는 서비스의 존재 여부와 위치에 대한 정보를 얻어 서비스 제공자와 직접 통신을 하게 된다.

#### 2.2 WS-I 테스트 도구

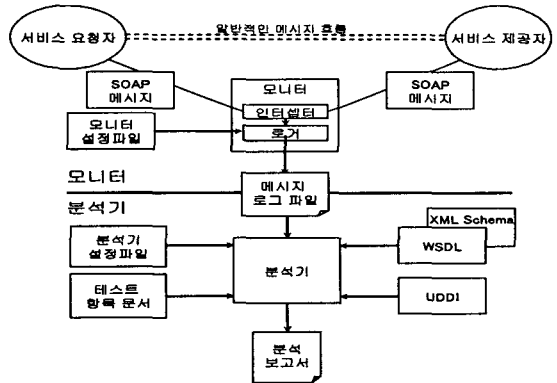


그림 2. WS-I 테스트 도구 구조  
Fig. 2 WS-I Test Tools Architecture

그림 2에서는 WS-I 테스트 도구의 구조를 보여주고 있다.

WS-I에서는 기본 프로파일을 기준으로 웹서비스 상호 운용성을 테스트 할 수 있는 테스트 도구로 모니터와 분석기를 Java와 C# 두 가지 버전으로 제공한다[5].

모니터는 웹서비스 제공자와 요청자 사이에 위치하고 내부의 인터셉터를 이용하여 교환되는 요청 및 응답 메시지를 모니터링 한다. 이렇게 모니터링 된 메시지는 로그를 통해 메시지 로그 파일로

생성되고 메시지 로그 파일은 분석기가 분석을 하기 위한 대상이 된다. 분석기는 테스트 항목 문서에 나타난 사항들을 기준으로 메시지 로그 파일에 나타난 정보들을 각 항목 별로 분석하게 되고 분석된 결과를 결과 보고서 파일에 기록하여 저장한다.

### III. 웹서비스 설계 및 구현

#### 3.1 웹서비스 설계

##### 3.1.1 .NET 웹서비스 제공자 환경 설계

웹서비스 제공자 환경은 .NET 프레임워크를 런타임 환경으로 하고 ASP.NET 웹 컨테이너와 도서 정보를 담고 있는 데이터베이스 그리고 WebMethod로 크게 구성되고 RPC 기반의 메시지 전송 모델을 이용하도록 설계하였다.

서비스 요청자에서 HTTP를 통해 검색될 도서 정보에 대한 요청 메시지가 들어오면 ASP.NET에서 SOAP 메시지를 받아 WebMethod로 전달한다. WebMethod는 SOAP 메시지에서 키워드를 찾아 데이터베이스의 해당 도서 정보를 찾아 일치하는 도서 정보를 XML 형식으로 변환하고 SOAP 메시지로 만들어 ASP.NET으로 넘겨준다. ASP.NET은 넘겨받은 SOAP 메시지를 HTTP를 통해 응답 메시지로 서비스 요청자 측으로 전달하게 된다.

그리고 WSDL 문서는 웹서비스 생성 시에 각 플랫폼에서 제공하는 툴이 자동으로 생성해 주는데 .NET 기반의 웹서비스에서는 .NET 프레임워크가 자동으로 WSDL 문서를 생성한다.

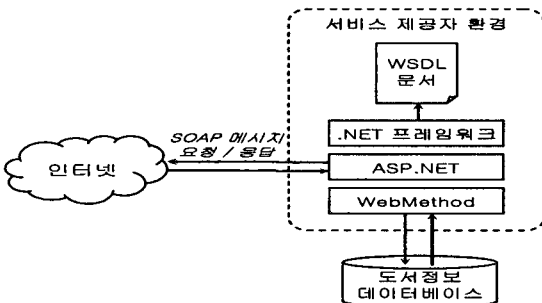


그림 3. .NET 웹서비스 제공자 환경 구조  
Fig. 3 .NET based Web Services Provider Architecture

그림 3은 .NET 도서 정보 웹서비스 제공자의 구조를 보인다.

##### 3.1.2 Java 웹서비스 요청자 환경 설계

웹서비스 요청자 환경은 J2RE(Java 2 Runtime Environment)를 런타임 환경으로 하고 Java 웹서비스의 RPC(Remote Procedure Call) 기반 메시지 전송 모델 중 가장 간단한 클라이언트 호출 모델인 정적 스텝(Static Stub) 방식을 사용하였다[6].

정적 스텝 웹서비스 호출 모델은 서비스 제공자에서 생성된 WSDL 문서를 통해 스텝 클래스를 생성하고 스텝을 통해 서비스 제공자 측과 연결되는 방식으로 Java RMI의 통신 모델과 유사한 클라이언트 호출 모델이다.

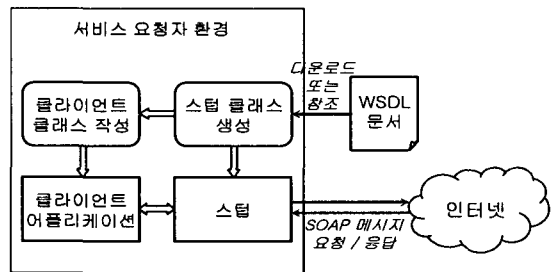


그림 4. Java 웹서비스 요청자 환경 구조  
Fig. 4 Java based Web Services Requester Architecture

그림 4는 Java 웹서비스 요청자 환경의 구조를 보인다.

#### 3.2 웹서비스 구현

##### 3.2.1 .NET 웹서비스 제공자 환경 구현

웹서비스 제공자의 구현은 C#을 이용하여 ASP.NET 기반으로 System.Web.Services.WebService 클래스를 상속받아 WebMethod 속성을 사용하여 웹서비스로 사용할 수 있게 구현하였다. 또한 SoapRpcMethod 속성을 사용하여 RPC 기반의 웹서비스 호출 모델을 사용하게 하였다[7].

본 시스템의 서비스 제공자는 요청자가 요청한 도서 정보 목록에 대한 응답을 배열 형식으로 반환하게 하였다.

##### 3.2.2 Java 웹서비스 요청자 환경 구현

웹서비스 요청자는 Sun JWS DP 1.4[8], Apache Axis 1.1[9], Bea WebLogic 8.1.2[10] 등 서로 다른 세 개의 플랫폼을 이용하여 각각 구현하였다.

각 서비스 요청자 환경은 동일하게 정적 스텝 클라이언트 호출 모델 기반으로 각각의 플랫폼에서 제공하는 툴을 사용하여 하나의 .NET 서비스 제공자의 같은 WSDL 문서를 통해 스텝 클래스를 생성하여 구현하였다.

클라이언트 어플리케이션은 Java SWING을 이용하여 구현하였고 원하는 도서 정보의 검색을 용이하게 하기 위해 제목, 저자명, 출판사, ISBN을 키워드로 사용하여 검색을 할 수 있게 하였고 검색 결과는 데이터 출력 창에 테이블 형식으로 출력하고 검색된 도서 정보를 클릭하면 이미지를 볼 수 있게 하였다.

#### IV. 상호 운용성 테스트 프레임워크 구축 및 시뮬레이션

##### 4.1 상호 운용성 테스트 프레임워크 구축

3장에서 설계 및 구현된 웹서비스 제공자와 요청자 환경, 그리고 WS-I에서 제공하는 Java 버전의 테스트 도구를 이용하여 웹서비스 상호 운용성을 테스트 할 수 있는 프레임워크를 구축하였다.

그림 5는 상호 운용성 테스트 프레임워크의 구성과 테스트 시나리오를 보여주고 있다.

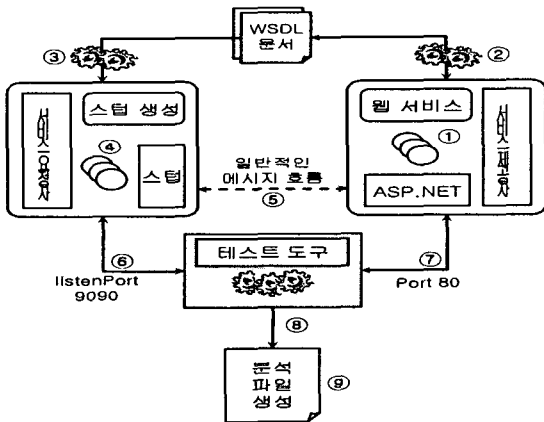


그림 5. 상호 운용성 테스트 프레임워크 구조 및 테스트 시나리오

Fig. 5 A Test Framework architecture and Testing Scenario for Web Services Interoperability

테스트 시나리오의 과정은 아래와 같다.

- ① .NET 기반의 도서 정보 웹서비스 제공자 구축
- ② .NET 프레임워크가 WSDL 문서 생성
- ③ WSDL을 참고로 Java 스텝 클래스 생성
- ④ Java 기반의 서비스 요청자 및 클라이언트 어플리케이션 구축
- ⑤ .NET 기반의 웹서비스 제공자와 Java 기반의 웹 서비스 요청자간의 메시지 송수신 테스트
- ⑥ WS-I 테스트 도구를 서비스 제공자와 요청자

- 사이에 두어 응답 메시지 전송
- ⑦ .NET 웹서비스 제공자의 응답 메시지 수신
- ⑧ 메시지 모니터링을 통한 WS-I 분석 파일 생성
- ⑨ WS-I 분석 파일 검증

##### 4.2 시뮬레이션

본 논문에서는 시뮬레이션의 객관성을 높이기 위해 하나의 웹서비스 제공자와 세 개의 웹서비스 요청자로 시뮬레이션을 하였다. 시뮬레이션을 위한 웹서비스 플랫폼은 현재 시장에서 많이 사용되는 .NET 플랫폼 기반의 서비스 제공자와 Java 기반의 서비스 요청자 플랫폼은 Sun JWSDP, Apache Axis, BEA WebLogic을 이용하였다.

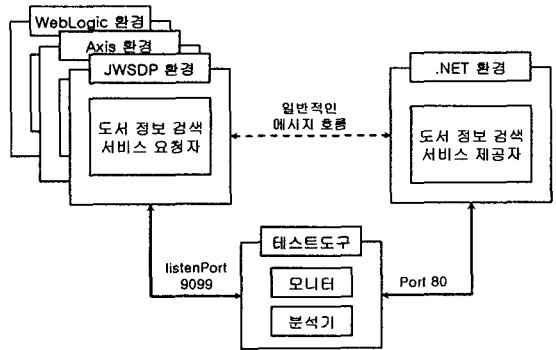


그림 6. 시뮬레이션 구성도

Fig. 6 A Simulation architecture for Test Framework

그림 6은 시뮬레이션을 진행하기 위한 테스트 프레임워크의 구성을 보여주고 있다.

시뮬레이션을 진행하기 위해 먼저 WS-I 테스트 도구의 모니터를 실행 시키고 각 서비스 요청자에서 서비스 종단점(Service Endpoint)의 URL을 모니터 설정 파일에서 설정된 listenPort 9099으로 수정하여 요청 메시지를 서비스 제공자로 직접 보내지 않고 모니터로 보내게 한다. 모니터는 요청 메시지를 모니터링 한 후 서비스 제공자 측으로 보내고 서비스 제공자로부터 반환된 응답 메시지 또한 모니터가 받아서 모니터링한 후 서비스 요청자 측으로 전달 되게 된다. 이와 같은 방법으로 Sun JWSDP, Apache Axis, BEA WebLogic으로 구축된 세 개의 서비스 요청자 환경 모두를 이용하여 시뮬레이션을 하였고 그 결과를 분석하였다.

##### 4.3 시뮬레이션 결과 및 분석

WS-I 테스트 도구의 분석기는 discovery 7개 항목, description 49개 항목, messages 42개 항목에

대해 테스트를 수행하고 그 결과를 passed, failed, warning, notApplicable, missingInput 으로 구분하여 반환한다[11].

본 테스트 프레임워크에서는 UDDI 서비스 레지스트리를 사용하지 않았으므로 discovery 항목에 관련된 테스트는 수행하지 않았고, 각 플랫폼 별 description과 message 항목에 대한 상호 운용성 시뮬레이션 결과는 아래의 표 1과 같이 나타났다.

표 1. 웹서비스 상호 운용성 시뮬레이션 결과  
Table. 1 A Result of Web Services Interoperability Simulation

	결과 항목	Sun JWSDP	Apache Axis	Bea WebLogic
description	passed	28	28	28
	failed	2	2	2
	warning	0	0	0
	notApplicable	18	18	18
	missingInput	1	1	1
messages	passed	10	28	30
	failed	2	4	4
	warning	0	1	0
	notApplicable	53	32	31
	missingInput	0	0	0

본 시뮬레이션의 결과에서 description에 관한 결과 항목들은 동일한 WSDL 문서를 참조하고 있으므로 플랫폼의 종류에 상관없이 같은 결과를 보여주고 있다.

description과 messages 두 가지 항목에서 failed 나 warning이 발생된 항목들은 모두 soapenc:arrayType과 soap:encodingStyle 속성을 사용한 부분으로 웹서비스의 SOAP 메시지 Body 내부에 배열 형식과 확장 인코딩 스타일을 사용한 부분이다. 이는 아직 많은 플랫폼들의 API들이 배열 형식과 확장 인코딩 스타일을 지원하지 않으므로 WS-I 기본 프로파일의 현재 버전에서는 배열 형식의 데이터 타입과 확장 인코딩 스타일의 사용을 권장하지 않고 있다.

또한 notApplicable과 missingInput이 나타난 항목들은 WS-I 테스트 도구의 현재 버전에서는 테스트를 지원하지 못하는 부분에 대한 결과로 서비스 수행에는 큰 영향을 주지 않고 차후 버전에서는

지원될 것으로 보여진다.

## V. 결 론

최근 인터넷 관련 기술이 발달하고 웹의 이용이 급속하게 증가함에 따라, 웹을 기반으로 하는 이질적인 어플리케이션간의 상호 작용을 체계적으로 지원하기 위한 새로운 분산컴퓨팅 환경으로 공개 표준 기술 기반의 웹서비스가 등장하였다.

그러나 웹서비스 또한 각 플랫폼 별로 지원하는 표준들의 버전 불일치와 같은 문제로 인해 상호 운용성에 어려움을 겪게 되었고 이에 대한 해결책으로 WS-I에서 웹서비스 상호 운용성을 위한 표준을 제시하게 되었다.

이에 본 논문에서는 서로 다른 웹서비스 플랫폼을 이용하여 웹서비스를 구축하고 WS-I 테스트 도구를 이용하여 상호 운용성을 테스트 할 수 있는 프레임워크를 구축하고 시뮬레이션을 한 결과를 검증하였다. 따라서 차후 웹서비스 구축 시에 상호 운용성을 위한 참고 자료로 활용될 수 있을 것이고 국내 웹서비스 상호 운용성 연구의 중요한 기반 기술이 될 것으로 사료된다.

향후 연구 과제로는 WS-I 기본 프로파일 이외에 현재 WS-I에서 웹서비스 상호 운용성을 위해 진행 중인 SOAP 바인딩 프로파일, SOAP Attachments 프로파일, Security 프로파일에 관한 연구도 진행되어야 할 것이다.

## 참고문헌

- [1] Web Services Interoperability Organization, <http://www.ws-i.org>
- [2] WS-I, Basic Profile Version 1.0 final, <http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html>
- [3] W3C, Web Services Architecture, <http://www.w3.org/TR/ws-arch>
- [4] New to SOA and Web services, <http://www-106.ibm.com/developerworks/webservices/newto/websvc.html>
- [5] WS-I, Implementation Tools, <http://www.ws-i.org/deliverables/workinggroup.asp?wg=testingtools>
- [6] Creating Web Service Clients with JAX-RPC, <http://java.sun.com/webservices/docs/1.2/tu>

torial/doc/index.html

- [7] 관리되는 코드의 XML Web services를 위한 코드 모델, <http://msdn.microsoft.com/library/kor/default.asp?url=/library/KOR/vbcon/html/vbconcodemodelforwebservicesinmanagedcode.asp>
- [8] Java Web Services Developer Pack, <http://java.sun.com/webservices/downloads/webservicespack.html>
- [9] WebServices - Axis, <http://ws.apache.org/axis/>
- [10] BEA WebLogic Platform 8.1, <http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/platform>
- [11] Basic Profile 1.0 Test Assertions Version 1.1, <http://www.ws-i.org/Testing/Tools/2004/10/BasicProfileTestAssertions.xml>

저자소개

**임산송(San-Song Im)**



1998년 배재대학교 컴퓨터공학과(학사)  
 1987년 배재대학교 컴퓨터공학과(공학석사)  
 1993년 배재대학교 컴퓨터공학과(박사수료)

1995년~현재 중부대학교 학술정보센터  
 ※관심분야 : 멀티미디어 문서정보처리, SGML/XML, SVG, MathML, Web Database

**이원석(Won-Suk Lee)**



1996 배재대학교 전자계산학과(이학사)  
 1998 충남대학교 컴퓨터공학과(공학석사)  
 2003 충남대학교 컴퓨터공학과(박사수료)

1998~2000 교육부산하 한국교육학술정보원 연구원  
 2000~2002 해동정보통신(주) 기술연구소 전임연구원  
 2003~현재 한국전자통신연구원 표준연구센터 연구원  
 ※관심분야 : 데이터베이스, XML, 웹 서비스, 시맨틱 웹

**전중홍(Jong-Hong Jeon)**



1993 한림대학교 전자계산학과(학사)  
 1996 한림대학교 컴퓨터공학과(석사)  
 1996~1999 한국정보시스템 기술개발연구소 주임연구원

1999~현재 한국전자통신연구원 표준연구센터 선임연구원  
 ※관심분야 : 차세대 웹 기술, 시맨틱 웹, 웹 서비스, KMS/EIP, CSCW, Workflow

**이강찬(Kang-Chan Lee)**



1994 충남대학교 컴퓨터공학과(공학사)  
 1996 충남대학교 컴퓨터공학과(공학석사)  
 2001 충남대학교 컴퓨터공학과(공학박사)

2001~현재 한국전자통신연구원 표준연구센터 선임연구원  
 2001~현재 W3C 대한민국 사무국 코디네이터  
 2002~현재 정보통신부 지정 IT 국제 표준 전문가  
 2003~현재 ASTAP NGW EG Rapporteur  
 2003~현재 웹 코리아 포럼 웹 서비스 WG 의장  
 2004~현재 TTA 웹 프로젝트 그룹(PG401) 부의장  
 ※관심분야 : 데이터베이스, 정보통합, XML, 미디어이터, 웹 서비스

**이승윤(Seung-Yun Lee)**



1991 광운대학교 전자계산학과(석사)  
 1997 광운대학교 컴퓨터공학과(박사)  
 1999~현재 한국전자통신연구원 표준연구센터 선임연구원

2002~현재 IPv6 포럼코리아 네트워크 WG 의장  
 2003~현재 한국전자통신연구원 표준연구센터 서비스융합표준연구팀 팀장  
 2003~현재 APAN IPv6 Task Force Technology Group Chair  
 2004~현재 ANF IPv6 Task Force Chair  
 2004~현재 정보통신부 지정 IT 국제표준전문가  
 ※관심분야 : 웹 서비스, 멀티미디어, 차세대 인터넷, IPv6



**정희경(Hoe-Kyung Jung)**

1985년 광운대학교 컴퓨터공학과(공학사)

1987년 광운대학교 컴퓨터공학과(공학석사)

1993년 광운대학교 컴퓨터공학과(공학박사)

1994년~현재 배재대학교 IT공학부 컴퓨터공학과 부교수

※관심분야 : 멀티미디어 문서정보처리, XML, SVG, Web Service, Semantic Web, MPEG-21