



한국정보과학회
Korea Information Science Society

제31권 제1호

2004 봄

학술발표논문집 (B)

Proceedings of
The 31st KISS Spring
Conference



한국정보과학회
KOREA INFORMATION SCIENCE SOCIETY
2004년 4월 23일~25일 KIST

29. 디자인 패턴 기반 소프트웨어의 테스트 가능성 분석	강영남 · 최은민	427
30. BtoC 기반의 온라인 쇼핑몰 테스트 기법에 관한 연구	김연수 · 최은민	430
31. 소프트웨어 분석 체계	박대성 · 강성원	433
32. 정보화 사업을 위한 표준/지침 분류체계 연구	변현진 · 문재형 · 신신애 · 이현중	436
33. 원격 MMIS 소프트웨어 개발을 위한 시험 프레임워크 개발	이종복 · 서상문 · 서용석 · 장귀숙 · 금종용 · 구인수	439
34. 컴포넌트기반 원격프로세스 심사지원 도구의 설계와 구현	박정민 · 이은석	442
35. 상호운용 성숙도 모델을 이용한 상호운용성 측정 사례 연구	문소영 · 류동국 · 서지연 · 김영철	445
36. 분산환경에서의 비즈니스 정보시스템 아키텍처 분류	이혜선 · 이은배 · 고현희 · 박재년	448
37. 사용자 인터페이스에서 비즈니스 이벤트의 가시적 응집도 향상 모델과 측정 방법	김진영 · 김정옥 · 유철중 · 장옥배	451
38. 유비쿼터스 환경을 위한 컴포넌트 기반 자바가상머신의 설계	윤승환 · 이승룡	454
39. 소프트웨어 개발과정에서 정량적 품질평가를 위한 프레임워크	강승훈 · 이길섭 · 이승중	457
40. 웹 응용 제공학에 기반한 핵심 애셋의 추출	이광우 · 이병정 · 김희천 · 우치수	460
41. MAG : WAP 푸쉬 서비스를 제공하는 모바일 응용 생성기	강이지 · 박은희 · 음두현	463
42. 자동화된 시맨틱 웹서비스의 조합을 위한 OWL-S 모델의 확장	이승근 · 이세훈 · 이정현	466
43. TDD를 위한 개선된 테스트 프레임워크 설계	손병길 · 류호연 · 박재홍	469
44. 내장형 실시간 시스템의 소프트웨어 아키텍처 평가 절차 및 성능평가 기준 고려사항	권도형 · 최윤석 · 정기원	472

4. 인간과 컴퓨터 상호작용

1. 스마트폰 상에서의 사용자 행위추론/예측기반 지능형 합성 캐릭터	이두호 · 한상준 · 조성배	475
2. 유전자 프로그래밍을 이용한 대화형 에이전트의 동적 답변 생성	김경민 · 임성수 · 조성배	478
3. 적응적인 시맨틱 정보 추출을 위한 대화형 사용자 질의 인터페이스 개발	조명현 · 손진현	481
4. 직접조작을 위한 NK-landscape 기반 IGA 인터페이스	민현정 · 조성배	484
5. 캠퍼스에서 개인화된 위치기반 서비스를 제공하기 위한 유비쿼터스 에이전트	강현지 · 한상준 · 조성배	487
6. 3D 한복 애니메이션을 위한 세형별 크기 조정 지식베이스의 설계 및 구현	오수정 · 이보란 · 남양희	490
7. AR-Station : 도시설계를 위한 가상현실 협업 시스템	임진욱 · 김병철 · 이현정 · 원광연	493
8. HMM을 이용한 제스처 기반의 게임 인터페이스	장상수 · 박혜선 · 김상호 · 김함준	496
9. e-Learning상에서 온라인시험 응시자 인종에 관한 연구	조길익 · 최덕훈	499

상호운용 성숙도 모델을 이용한 상호운용성 측정 사례 연구

문소영^o 류동국* 서지연 김영철
 홍익대학교 컴퓨터정보통신
 국방과학연구소*
 whit2@selab.hongik.ac.kr^o

A case study about interoperability measurement using interoperability maturity model

SoYoung Moon^o DongKuk Ryu* JiYoun Seo R.YoungChul Kim
 Dept. of Computer and Information Communication, Hongik University
 Agency for Defense Development*

요약

본 논문에서는 컴포넌트 기반 시스템 개발시 시스템간의 상호운용성을 높일 수 있는 방안을 제시하는데 있다. 다시 말하자면 다른 웹 서비스 방식과 분산 객체 방식의 컴포넌트 기반 시스템들의 상호운용 성숙도 정도를 측정하려는 시도에 있다. 적용 사례에서는 클라이언트는 .NET 환경을 서버는 EJB 환경 상에서 개발된 EJB와 .NET 컴포넌트는 웹 서비스 방식과 IIOP.NET를 이용한 분산객체 방식 두 가지 방식으로 상호운용 하였다. 본 논문에서는 이 두 가지 상이한 상호운용 방식에 대하여 성숙도 모델인 LISI를 기반으로 상호운용 측정 기법을 적용하여 상호운용 능력을 측정하려 노력하였다. LISI 상호운용 능력을 측정한 결과 웹 서비스에 의한 방식이 분산 객체 방식보다 높은 상호운용 능력이 있음이 나타난다.

서론

규모가 크고 복잡한 소프트웨어는 일반적으로 다양한 소프트웨어 모듈들로 구성된다. 기존 컴포넌트(legacy component) 또는 서로 다른 언어들로 작성된 이종 컴포넌트들을 재사용하고자 할 때 다른 언어로 개발된 소프트웨어 컴포넌트들 사이에 상호 협력의 문제가 발생한다 [1]. 이러한 이유로 개발자들은 상호운용성에 대해서 많은 관심을 가지게 되었다.

일반적인 상호운용의 형태는 시스템간의 단순한 정보 교환 및 교환이었다. 정보 기술 환경이 발전하고 복잡해짐에 따라, 상호운용의 형태도 분산 환경에서의 시스템 간에 일부 기능을 공유하는 형태로 발전하고 있다.

이종 응용 체제(heterogeneous application)의 상호운용(interoperability)은 서로 다른 프로그래밍 언어와 플랫폼에서 운영되는 체제들이 다른 네트워크 상에서 통신할 수 있도록 하는 능력을 말한다[2].

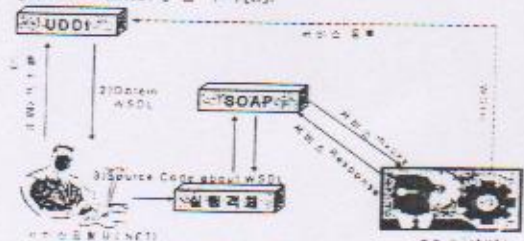
이들 언어(EJB, COM/DCOM)는 컴포넌트 인터페이스를 통해 컴포넌트의 기능을 접근하기 위한 상호운용성 문제 해결책을 제공한다 [2]. 최근 많은 정보 시스템이 분산 환경에서 EJB와 .NET과 같은 컴포넌트 모델을 사용하여 개발되는 추세이다. 또한 개발되는 시스템은 데이터 일부 프로그램을 공유하여 개발되어 상호 유기적인 동작을 하는 경우도 있다. 본 논문의 2장에서는 관련연구로 웹 서비스를 이용한 상호운용과 분산 객체 수준에 있는 IIOP.NET을 통한 상호운용에 대해서 살펴보고, 3장에서는 웹 서비스와 분산객체를 이용한 적용 사례를 살펴볼 것이며, 4장에서는 LISI 기반 상호운용 성숙도 모델을 이용하여 웹 서비스와 분산객체의 상호운용의 성숙도

를 측정해 볼 것이다. 5장에서는 어떤 접근 방법이 상호운용에 적합한지를 논할 것이다.

2. 관련 연구

2-1. 웹 서비스를 이용한 상호운용성

웹 기술들이 발전하면서, 웹 서비스(web service)라는 개념들이 나타나게 되었다. 웹 서비스란 웹 기술과 XML을 이용한 분산 컴퓨팅(distributed computing)이라고 할 수 있다. 서비스는 애플리케이션이 될 수도 있고, 프로세스일 수도 있고, 하나의 컴포넌트일 수도 있다. 그러나 서비스를 이용하는 입장에서는 어떤 기술로 어떻게 만들어져 있는가는 중요하지 않다[3]. CORBA, RMI, DCOM과 같은 기존 기술들 대신에 웹 서비스를 이용하는 이유는 웹이 접근하기 쉽고, 어디에서나 사용할 수 있기 때문이다. 또한 웹 서비스의 목적은 이기종 플랫폼의 연동이다[4]. 웹 서비스의 장점은 플랫폼 및 언어의 독립성, JIT(Just-in-time)통합이다[5].



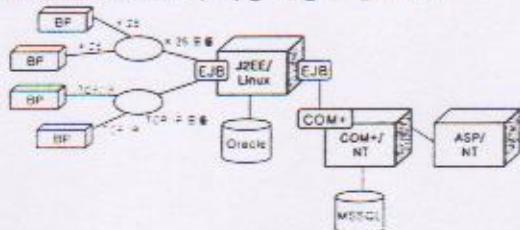
<그림 1> 웹 서비스를 통한 상호운용 절차

<그림 1>에서와 같이 무선은 서비스를 제공하는 EJB

는 UDDI에 서비스를 등록한다. 서비스 요청자 .NET은 UDDI에서 서비스를 검색한다. 제공받은 서비스를 검색한 후 WSDL을 얻어와 이에 맞는 언어로 소스코드를 작성한다. XML과 HTTP를 이용해서 어느 플랫폼이든지 해당 서버와 서비스에 접근할 수 있도록 해주는 SOAP(Simple Object Access Protocol)을 통하여 서비스를 요청하고, 응답을 받아들일 수 있다.

2-2. 분산객체를 이용한 상호운용성

JAVA/EJB와 .NET/COM+ 같이 재사용 가능한 컴포넌트로 이뤄진 시스템들을 통합하려 한다. 또한 다른 플랫폼에서 개발한 컴포넌트를 사용하려할 때 각 시스템을 재구축하지 않고 재사용한다는 것은 시간과 노력 및 비용을 절약하게 된다. 이렇게 하는 방법이 바로 상호운용인데, 이전의 시스템에 RMI(Remote Method Invocation)와 CORBA(Common Object Request Broker Architecture) 연동 부분이 있을 경우 IIOP(Internet Inter-ORB Protocol)에 대한 지원이 필요하다.



<그림 2> EJB 시스템과 COM+ 시스템 연결

EJB 시스템에서 NET로 구축된 시스템의 COM을 직접 호출하고, 거꾸로 NET에서 EJB를 호출할 필요가 있을 수 있다. <그림 2>는 EJB 시스템과 COM+ 시스템의 상호운용을 위한 상황을 도식화하여 보여주고 있다. 상호운용을 도모하기 위해서 <그림 4>의 IIOP.NET을 이용한다.

3. 적용 사례

EJB와 .NET 간의 상호운용성을 웹 서비스와 분산객체를 이용하여 구현해보았다.

제한조건으로는 서버 환경에 EJB를 채택하였고, 클라이언트 환경은 .NET하에서 시도해본다.

3-1 웹 서비스를 이용한 구현

웹 서비스를 이용해서 서버로부터 파일을 얻어오고, 클라이언트가 서버에게 파일을 전송해주는 시나리오로 구현하였다.

```

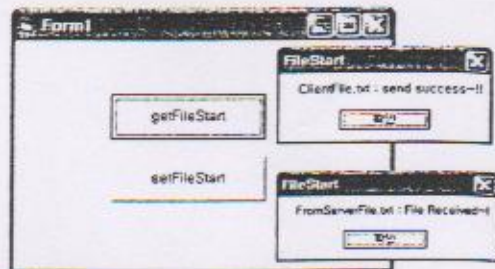
서버 EJB 소스의 일부분
public void setFileDataHandler(dh)

try {
    fos = new FileOutputStream("FromClientFile.txt");
    dh.writeToFos;
    fos.flush();
} catch {
}
    
```

```

'클라이언트 .NET 소스의 일부분
WScript.Echo ("Create Client Succeeded
===== ")
'wsdl과 wsml을 이용하여 SoapClient 객체를 초기화한다.
call client.mssoapinit(wsdl, "", "", wsml)

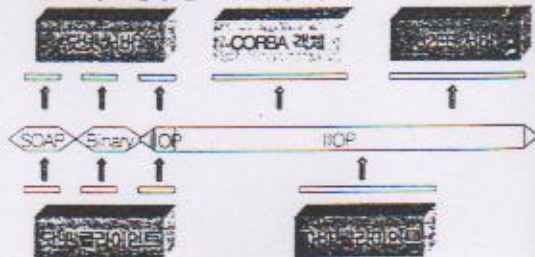
.....
WScript.Echo ("Init Client Succeeded -----" )
Set ReceivedAttach = Client.getFile()
FileSaveName = "FromServerFile.txt"
ReceivedAttach.SaveToFile FileSaveName, True
    
```



<그림 3> 웹 서비스를 이용한 파일전송 실행 화면

3-2. IIOP.NET를 이용한 분산 객체 구현

IIOP.NET은 CORBA의 핵심 부분인 IIOP 기반의 닷넷 리모팅 채널과 IDL(Interface Definition Language)과 CLS(Common Language Specification)간 변환 등을 제공한다. 자바에는 이미 RMI IIOP가 제공되고 있으므로 IIOP.NET을 이용하면 자바의 RMI IIOP 기반의 애플리케이션과 상호운용할 수 있다[6,7].



<그림 4> IIOP.NET 개요

```

서버 EJB 소스의 일부분
.....
AdderImpl adder = new AdderImpl();
publish the reference with the naming service;
Context initialNamingContext = new InitialContext();
initialNamingContext.rebind("adder", adder);
System.out.println("Server Ready.");
    
```

```
// 클라이언트 NET C# 소스의 일부
IopClientChannel channel = new IopClientChannel();
ChannelServices.RegisterChannel(channel);

CorbalInit init = CorbalInit.GetInit();
NamingContext nameService =
    init.GetNameService(nameServiceHost, nameServicePort);
double result = adder.add(sum1, sum2);
Console.WriteLine("result: " + result);
```

```
C:\WJAVA_HOME\Wj26>input the two summands
sum1:
30
sum2:
50
result: 80
```

<그림 5> 서버와 클라이언트 실행

4. 성능도 모델을 이용한 상호운용성 측정

구분	항목	주요내용	기타사항	비고
1. 목적	1.1 목적	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화
	1.2 목적	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화
2. 범위	2.1 범위	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화
	2.2 범위	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화
3. 참조문헌	3.1 참조문헌	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화
	3.2 참조문헌	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화
4. 용어	4.1 용어	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화
	4.2 용어	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화
5. 측정 방법	5.1 측정 방법	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화
	5.2 측정 방법	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화
6. 결론	6.1 결론	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화
	6.2 결론	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화	상호운용성 측정 방법의 표준화

<그림 6> LISI의 상호운용성 레벨

사실 LISI[10,11]는 미국 DoD 정보시스템간의 상호운용성을 측정하기 위한 상호운용 성능도 모델이라서 상용 컴포넌트 시스템간의 성능도를 측정하려는 시도가 적절하지 않을 수 있다. 그러나 본 논문의 의도는 컴포넌트 기반 시스템간의 상호운용성은 현재 중요한 이슈가 되고 있어 우리 Lab에서는 시도로써 LISI 그 자체로 측정하였다. <그림 6>의 같이 상호운용 성능도 모델을 이용하여 두 가지 상호운용 기법에 대하여 상호운용 능력을 측정하였다. <그림 6>은 LISI 모델을 기반으로 상호운용 능력을 성능도에 기반하여 수준을 정의한 것이다[8,9]. <그림 6>을 보면, 웹서비스의 상호운용성을 측정할 결과 결과는 3c, 응용체계는 5b, 기반구조는 4b, 데이터는 4a이다.

다. 결과적으로 IMM을 이용한 웹서비스의 상호운용성은 레벨은 3c임을 알 수 있다. 그리고 분산객체를 이용한 상호운용성을 측정할 결과 결과는 1b, 응용체계는 5b, 기반구조는 4b, 데이터는 3c 로써 IMM을 이용해 살펴본 전체적인 레벨은 1b에 머무름을 알 수 있다.

5. 결론 및 향후 과제

4장에서 결과를 본대 상호운용 능력을 측정할 결과 웹서비스에 의한 방식이 분산 객체 방식보다 보다 높은 상호운용 능력이 있음을 알 수 있다. 분산 객체 방식의 경우 상호운용 절차부분에서 웹 서비스에 비교하여 낮은 평가를 보였다. 웹 서비스는 표준화된 절차와 규격을 통하여 높은 상호운용 능력을 보여준다. 그리고 시스템 환경과 상호운용성을 생각하면 항상 느슨한 연결(loosely coupling)과 표준(standardization) 기반으로 시스템을 구성하는 것이 좋다. 성능도 모델을 이용하여 상호운용성을 측정할 결과 비교적 비슷했지만, 표준기술을 사용한다는 의미에서 웹 서비스를 이용하는 것이 상호운용성에 도움이 된다는 것을 알 수 있다. 그러나 웹 서비스는 트랜잭션 처리 관련 규약이 미흡하다. 또한, 성능이나 튜닝에 대한 더 구체적인 방법들이 벤더들의 책임만으로 되어 있고, 표준화에 대해서는 미흡한 편이다. 웹 서비스 기술은 만능이 아니며 비동기라는 측면 때문에 단순하지만 오히려 복잡해지기도 한다[3]. 이러한 문제로 향후엔 Memo 스타일의 어댑터를 개발하여 비동기적인 측면을 해소하고자 하고, 상호운용 성능도 모델을 상용 컴포넌트 기반 시스템의 상호운용성 측정을 할 수 있도록 보완해야 할 것이다.

참고 문헌

- [1] Barrett, Kaplan, Wiledn, "Automated Support for Seamless Interoperability in Polylingual Software Systems", SIGSOFT 1996
- [2] C.C. Chiang, "The use of adapters to support interoperability of components for reusability", 2002
- [3] 이승준, 마이크로소프트웨어, "웹 서비스와 상호운용성", 2003, 12
- [4] 이영석, 마이크로소프트웨어, "현실로 다가온 웹 서비스, MS와 차라 플랫폼의 연동", 2003, 10
- [5] 최충병, 주국우, 최지영 공저, "자바 개발자를 위한 XML", 2002, 5
- [6] 이승준, 정근욱, 김경운, 마이크로소프트웨어, "J2EE.NET 분산 도체 상호운용성", 2003, 11
- [7] <http://img.ncl.com/interactnet>
- [8] "Level of Information System Interoperability (LISI)", C-ISR Architecture Working Group, 1998
- [9] "국방정보체계 상호운용성 수준(LISI) 업무편람", 국방부, 2002
- [10] 국방과학연구소, "컴포넌트 기반 세계 상호운용 적 합성 평가 및 인증 기술 연구", 2004
- [11] 국방과학연구소, "국방정보체계 상호운용 수준 LISI", 2000, 12