

웹 서비스를 이용한 이종 분산 컴포넌트 기반 시스템간 상호운용 성숙도 향상 연구

김기두*, 김동호, 김영철
홍익대학교 일반대학원 전자전산전공
e-mail: kdkim@won.hongik.ac.kr

A Study on Interoperability Maturity Elevation Between Heterogeneous Distributed Component Based Systems Using Web Services

Ki-Du Kim*, Dong-Ho Kim, R.Young-Chul Kim
Dept of Electronics and Computer Science Engineering ,
Hong-Ik University

요 약

본 논문은 현재 이종 분산 컴포넌트 기반 시스템간의 상호운용성을 부각하고자 한다. 컴포넌트 기반으로 개발된 시스템들이 이종의 플랫폼(CORBA, Net, JAVA)플상에 각 컴포넌트의 고유 기능 자체 그대로 상호운용이 가능해야 할 것이다. 그러나 각각의 벤더들에 의해 독자적으로 개발된 컴포넌트 모델/시스템을 사용하여 컴포넌트 표준화나 통합화에 많은 노력이 필요로 한다. 이는 시스템간의 상호운용성을 해결하는데 있다. 본 논문에서는 컴포넌트 상호운용 성숙도 모델(CIMM)을 언급하고 웹 서비스를 이용한 이종 분산 컴포넌트 기반 시스템 간의 상호운용성에 대해 상호운용 성숙도 모델(IMM)로 상호운용 수준평가를 통해 향상 방안을 제시하고자 한다.

1. 서론

소프트웨어 컴포넌트는 개발 및 운용 환경이 개발 벤더에 종속되기 때문에 동일한 개발 환경에서 만들어진 컴포넌트가 아니면 서로 간의 상호 운용성을 기대하기는 힘들다. 현재 이종 분산 컴포넌트 기반 시스템간의 상호운용성은 매우 중요한 일이다. 컴포넌트 기반으로 개발된 시스템들이 이종의 플랫폼(CORBA, Net, JAVA)상에 각 컴포넌트의 고유 기능 자체 그대로 상호운용이 가능해야 한다. 그러나 각각의 벤더들에 의해 독자적으로 개발된 컴포넌트 모델/시스템을 사용하여 컴포넌트 표준화나 통합하

에 많은 노력이 필요로 한다. 이는 시스템간의 상호운용성을 해결하는데 있다. 이종 분산 컴포넌트 기반 시스템내의 컴포넌트는 그 특성상 블랙박스의 형태로 개발되기 때문에 실제 하위 레벨에서의 상호운용성을 측정하기는 힘들다. 아래의 표 1은 국방 LSI 확장한 컴포넌트 기반의 CIMM(Component Interoperability Maturity Model)이다. 우리는 이를 이용하여 컴포넌트간의 상호운용성 수준을 측정하였다[1].

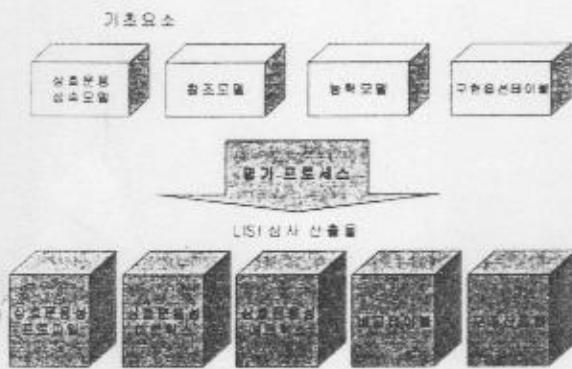
CIMM(Component Interoperability Maturity Model)의 컴포넌트 상호운용 수준은 5단계로 구분한다. 이기종 컴포넌트 간의 상호 운용성을 확보하

는 단계는 CIMM의 레벨 3이다. 즉, 수준 3은 이종 컴포넌트간의 상호운용성이 가능한 단계이다. 우리는 이종 컴포넌트간의 상호운용이 되기 위한 수준 3으로 향상시키는 방법으로 이종 컴포넌트간의 상호운용을 웹서비스를 이용하는 방법을 제안한다.

본 논문의 2장에서는 관련연구로 LISI, 웹 서비스, CIMM을 소개하고, 3장에서는 상호운용 수준 3이 되기 위한 웹 서비스의 활용에 대해서 언급하고, 4장에서 결론을 기술하였다.

2. 관련연구

2.1 LISI (Levels of Information Systems Interoperability)



(그림 1) LISI 구성요소[2]

LISI는 국방 정보체계의 능력을 심사하고 실제적인 해결책을 선택하며, 상호운용성과 수준 향상을 위한 프로세스를 제공해주기 위해 개발되었다. (그림 1)은 LISI 평가의 구성요소를 보여준다. LISI는 기초요소를 갖고 평가프로세스를 통해 나온 산출물로 정보체계 상호운용성을 정의, 평가, 측정, 심사하게 된다. 평가의 기준이 되는 기본요소의 기능은 다음과 같다.

- 상호운용성 성숙도 모델은 상호운용성의 수준을 6단계로 정의하며, 체계간의 상호작용에 대하여 증가하는 복잡도를 설명한다.
- 참조모델은 프로시저(P), 응용(A), 기반구조(I), 자료(D)의 속성으로 6단계로 수준을 구분한다. 각 수준에 정의된 상호운용성 성숙의 정도를 달성하기 위한 속성을 표현한다.
- 능력모델은 정보체계 상호운용 수준의 측정을 결

정하기 위해 요구하는 상세 수준과 LISI 심사를 위한 기초를 제공한다.

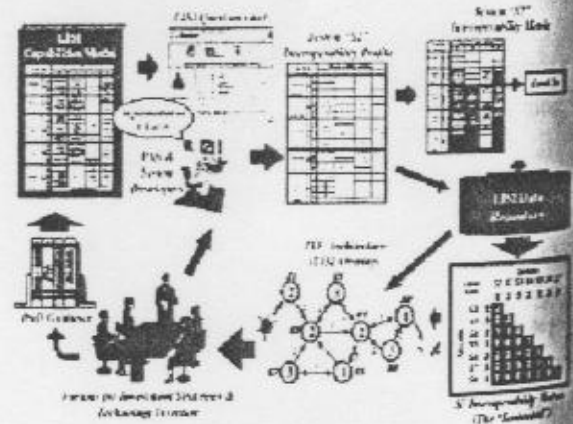
- 구현옵션 테이블은 개발자가 능력모델에 정의된 각 능력에 대한 구현을 위한 이용 가능한 구현선택의 범위를 표현한다.

<표 1>은 LISI 참조모델로 수준별 상호운용 성숙도와 PAID의 정의를 포함한다. LISI 프로세스에서는 <표 1>과 같이 PAID 측정과 함께 상호운용 성숙도를 측정하는 기준을 제시한다.

| Descriptor | Computing environment | Level | Interoperability Attributes | | | |
|------------|-----------------------|-------|-----------------------------|-------------------------|---------------------------------|------------------|
| | | | P | A | I | D |
| Enterprise | Universal | 4 | Enterprise level | Interactive | Multiple dimensional topologies | Enterprise model |
| Domain | Network | 3 | Domain level | GroupWare | World wide network | Domain model |
| Functional | Distributed | 2 | Program level | Desktop automation | Local networks | Program model |
| Connected | Peer to Peer | 1 | Local/site level | Standard system drivers | Simple connection | Local |
| Isolated | Monoc | 0 | Access control | N/A | Independent | Point |

<표 1> 상호운용성숙도와 LISI 참조모델[2]

LISI 프로세스는 공통적인 참조 틀과 성능측정을 제공하며 요구사항 분석부터 시스템 개발, 획득, 배치 그리고 유지 보수하는 정보체계 생명주기를 갖는다. (그림 2)는 LISI를 통해 상호운용성을 평가하는 프로세스를 보여준다[2].



(그림 2) LISI 심사 프로세스[2]

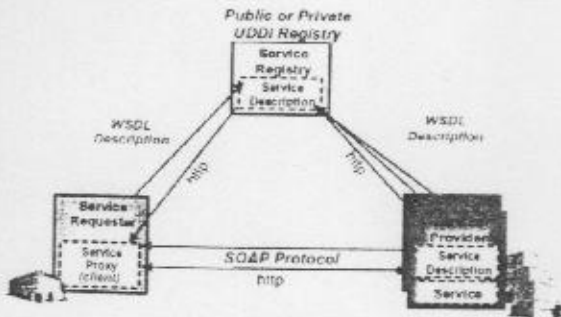
2.2 웹 서비스

웹서비스에서는 어플리케이션간의 상호작용을 해 최근에 만들어지고 있는 상위 프로토콜인 웹 서비스 프로토콜과 공개 표준인 XML을 이용하여

들어진 구조적이고 확장성 있는 프레임워크를 제안하고 있다.

웹 서비스 프레임워크는 통신 프로토콜, 서비스 기술, 서비스 발견의 세부분으로 나뉘어진다. 각각은 웹 서비스간의 통신을 제공하는 SOAP(Simple Object Access Protocol)과 컴퓨터가 읽을 수 있는 정형의 웹 서비스 기술을 제공하는 WSDL(Web Service Description Language)과 웹 서비스 기술을 등록하고 지장하는 UDDI(Universal Description, Discovery, and Integration) 기술이 있다[4].

웹 서비스 아키텍처는 서비스 제공자(Service Provider)와 서비스 요청자(Service Requester), 서비스 레지스트리(Service Registry)의 세 가지 역할로 이루어져있다. 이들 사이의 상호작용으로 'Publish - Find - Bind' 동작이 있다[4]. (그림 3)은 웹 서비스 아키텍처를 통해 웹서비스의 활동들을 보여준다. 웹 서비스는 서비스 제공자가 서비스를 상세 정보와 함께 레지스트리에 등록하는 Publish 기법, 서비스 요청자가 원하는 서비스의 내용을 검색하고 알아오는 Find 기능, 해당 서비스의 수행을 위해 호출하고 초기화하는 Bind 기능을 각각 수행한다[4][5].



(그림 3) 웹 서비스 구조[3]

2.3 CIMM(Component Interoperability Maturity Model)

CIMM은 컴포넌트간의 상호운용성을 측정하기 위해 LISI 평가 모델 중 능력모델을 확장한 모델이다[1]. LISI 모델은 측정기반이 되는 구현옵션에 컴포넌트를 고려하지 않았고, 1998년도에 개발된 능력 모델 또한 현재 OI모델로 개정하였지만 컴포넌트에 대한 고려된 사항이 없다[2]. 이점을 보완 확장한 것이 CIMM이다. CIMM은 <표 2>처럼 컴포넌트를 5개의 수준으로 구분하는데 각 수준별 구분사항은 다음과 같다.

• 의존 컴포넌트 : 가장 상호운용 수준이 낮은 단

계의 컴포넌트이며 개발 언어, 운용환경 등에 의존적인 컴포넌트로 특정 개발언어, 운영체제, 통신 인프라에서만 동작하는 단계이다.

• 독립 컴포넌트 : 독립 컴포넌트는 EJB나 .Net과 같이 특정 컴포넌트 모델이 정의되어 있어 모델에서 정의하는 규격을 만족하는 컴포넌트 간에 상호운용이 가능한 컴포넌트이다. 그러나 서로 다른 컴포넌트 간에는 상호운용이 불가능하다.

• 상호운용 컴포넌트 : 상호운용 컴포넌트 단계에서는 컴포넌트 모델에 제약이 없이 모든 컴포넌트가 컴포넌트 기술적으로는 상호운용이 가능한 단계이다. 그러나 각 컴포넌트간의 상호운용에 대한 세부적인 규약이나 절차는 정의되어 있지 않은 단계이다.

• 도메인 컴포넌트 : 도메인 컴포넌트는 컴포넌트를 공유하는 도메인에서 컴포넌트 표준을 개발하여 제공하므로 도메인 내부에서는 상호운용이 가능한 단계이다.

• 통합 컴포넌트 : 통합 컴포넌트는 컴포넌트의 표준이 정의되고 모든 컴포넌트 사용자에게 표준으로 개발되어 각각의 도메인에서 공통으로 사용가능한 단계이다[4].

| Level | 상호운용수준 | | C (Component) | 설명 |
|-------|------------|--------------|----------------------|--|
| | 범위 | 컴퓨팅환경 | | |
| 5 | Enterprise | Universal | 통합 (Enterprise) | <ul style="list-style-type: none"> b: 전 세계적 상호운용 a: 국가적 상호운용 -통합 환경의 컴포넌트 상호운용 가능 |
| 4 | Domain | Integrated | 도메인 (Domain) | <ul style="list-style-type: none"> b: 도메인간 상호운용 a: 도메인 내부 상호운용 -도메인 컴포넌트 상호운용 가능 |
| 3 | Functional | Distributed | 상호운용 (interoperable) | <ul style="list-style-type: none"> b: 컴포넌트 상호운용 기술적 명안 제시 a: 컴포넌트 규약 및 절차정의 -컴포넌트 간 상호운용 가능 -상호운용 규약 및 절차부재 |
| 2 | Connected | Peer to Peer | 독립 (independent) | <ul style="list-style-type: none"> b: 운용환경 독립 a: 개발언어, 운용환경 의존 -컴포넌트 모델종속 |
| 1 | Isolated | Manual | 의존 (dependent) | <ul style="list-style-type: none"> b: 운용환경 의존 a: 개발환경 의존 -개발환경 및 운용환경에 의존 |

<표 2> 컴포넌트 상호운용 성숙도 모델[1]

3. 컴포넌트 기반 시스템간의 상호운용 능력 문제

하나의 독립된 단위로 사용되는 컴포넌트의 경우는 외부 컴포넌트와 상호운용을 하지 않기 때문에 상호운용성을 고려하지 않아도 된다. 하지만 시스템

간의 통합이나 조립을 위해 개발된 컴포넌트의 경우 상호운용성을 고려해야한다. 하지만 현재 개발되는 컴포넌트들은 각각의 벤더들이 고유의 환경에서 개발되기 때문에 이종의 컴포넌트간의 상호운용성이 이루어 지지 않는 상황이다.

3.1 웹 서비스의 상호운용 수준 측정

상호운용성은 이종 분산 컴포넌트 기반 시스템간 상호연동을 위해 필요한 요소이다.

| 수준 | 특성 | 응용 사례 | 비판기준 | 데이터 |
|---------------|-------------------------------------|--|---|---|
| 3 상호 연동 | 1 무시각적 비지시적의 고차원 질적 다양성 존재 | 기업들간의 서비스-모형 공유한 상용화 예제 유용한 서비스-모형 (e.g. java 서비스, 웹용 Java applet, JAVA 기반 L2/3등) | 다양한 벤더명 사후-사후-사후 다양 기술들 | 비-벤더적 모델 |
| | 2 국방자료 비시시적의 고차원 질적 다양성 존재 | 1.10. 상호운용 (가짜) 응용 상용화 예제 공공서비스-비시시적 서비스-모형 예제들 (Database Gateway Picture) 2001. 12.17등) | 공제당(IBM) Haupt, Sotchi 2001. 12.17 공제당, IBM, MS 공제당, IBM, MS 2001. 12.17 | 공제당(IBM) Haupt, Sotchi 2001. 12.17 공제당, IBM, MS 공제당, IBM, MS 2001. 12.17 |
| 4 모듈 연동 | 1 비시시적의 고차원 질적 다양성 존재 | 1.10. 상호운용 (가짜) 응용 상용화 예제 공공서비스-비시시적 서비스-모형 예제들 (Database Gateway Picture) 2001. 12.17등) | 공제당(IBM) Haupt, Sotchi 2001. 12.17 공제당, IBM, MS 공제당, IBM, MS 2001. 12.17 | 공제당(IBM) Haupt, Sotchi 2001. 12.17 공제당, IBM, MS 공제당, IBM, MS 2001. 12.17 |
| | 2 국방자료 비시시적의 고차원 질적 다양성 존재 | 1.10. 상호운용 (가짜) 응용 상용화 예제 공공서비스-비시시적 서비스-모형 예제들 (Database Gateway Picture) 2001. 12.17등) | 공제당(IBM) Haupt, Sotchi 2001. 12.17 공제당, IBM, MS 공제당, IBM, MS 2001. 12.17 | 공제당(IBM) Haupt, Sotchi 2001. 12.17 공제당, IBM, MS 공제당, IBM, MS 2001. 12.17 |
| 5 완전 연동 | 1 비시시적의 고차원 질적 다양성 존재 | 1.10. 상호운용 (가짜) 응용 상용화 예제 공공서비스-비시시적 서비스-모형 예제들 (Database Gateway Picture) 2001. 12.17등) | 공제당(IBM) Haupt, Sotchi 2001. 12.17 공제당, IBM, MS 공제당, IBM, MS 2001. 12.17 | 공제당(IBM) Haupt, Sotchi 2001. 12.17 공제당, IBM, MS 공제당, IBM, MS 2001. 12.17 |
| | 2 국방자료 비시시적의 고차원 질적 다양성 존재 | 1.10. 상호운용 (가짜) 응용 상용화 예제 공공서비스-비시시적 서비스-모형 예제들 (Database Gateway Picture) 2001. 12.17등) | 공제당(IBM) Haupt, Sotchi 2001. 12.17 공제당, IBM, MS 공제당, IBM, MS 2001. 12.17 | 공제당(IBM) Haupt, Sotchi 2001. 12.17 공제당, IBM, MS 공제당, IBM, MS 2001. 12.17 |

<표 3> 웹 서비스의 상호운용성 수준 측정[6]

<표 3>은 LISI 능력모델을 통해 본 웹 서비스 상호운용 수준이다. 수준 1~2단계는 이종 분산 컴포넌트 간에 상호운용 수준에 이르지 못한다. 수준 3 이상에 도달해야만 이종 컴포넌트간에 상호운용 수준이 된다고 할 수 있다. 웹 서비스는 이종의 컴포넌트들 간에 상호운용을 SOAP과 WSDL의 기능을 이용하여 상호운용을 확보한다.

웹 서비스의 상호운용 성숙도를 측정하였을 때 <표 3>과 같이 수준 3이 됨을 알 수 있었다. 그러나, <표 3>에서 도메인에 대한 절차적(P)와 데이터

(D) 측면이 보완하기만 하면 수준 4이상의 상호운용 성도 가능함을 알 수 있었다. 우리는 이런 모델을 통해 시스템간의 상호운용 수준을 높일 수 있는 방안을 제시한다.

4. 결론

본 연구에서는 웹서비스를 이용한 이종 분산 컴포넌트 기반 시스템간의 상호운용하는 방법에 상호운용성속도(IMM) 통해 상호운용 성숙도 수준 3정도로 측정하였고, 이를 이용하여 상호운용 성숙도 수준 3에서 4이상으로 향상시키기 위한 방법을 언급하였다. 상호운용성속도 측정은 LISI의 상호운용 성숙도 & 능력모델을 통해 측정하였다. 특히 본 연구에서는 웹서비스의 상호운용성 수준 측정뿐 아니라 측정된 웹서비스 수준을 이용하여 보다 높은 상호운용을 하기위해 상호운용성속도모델(IMM)을 통해 향상할 수 있는 요소들을 알 수 있었다.

향후 컴포넌트 기반 개발 단계부터 상호운용 성숙도 모델을 이용하여 컴포넌트 기반 시스템간의 상호운용 성숙도 수준을 측정하여 개발한다면 컴포넌트간의 상호운용이 성공적인 컴포넌트 개발에 높은 효과를 보여줄 것으로 기대한다.

참고문헌

[1] 류동국, 김영철, "컴포넌트 기반 시스템 상호운용성 측정 및 평가를 위한 상호운용 능력 모델 개발," 제 31회 정보 과학회 춘계학술 발표 논문집, 2004.(accepted)

[2] Mark Kasunic "Measuring Systems Interoperability," Software Engineering Institute Carnegie Mellon University, 2001.

[3] 김영철, "컴포넌트 기반 체계 상호운용 적합성 평가 및 인증 기술 연구(최종보고서)," 국방과학연구소, 2004.

[4] Francisco Curbera, Matthew Duftler, Rania Khalaf, William Nagy, Nirmal Mukhi, and Sanjiva Weerawarana, "Unraveling the Web Services Web," IEEE, 2002.

[5] Simeon Simeonov 외, 자바를 이용한 웹서비스 구축, 인포북, 2002, pp 46-49.

[6] 문소영, 김영철 "상호운용 성숙도 모델을 이용한 상호운용성 측정 사례 연구," 제 31회 정보 과학회 춘계 학술 발표논문집, 2004.(accepted)