

상호운용성 시험기술의 이론과 실제

강성원(한국정보통신대학교 (ICU)), 장 용, 박용범(한국정보통신기술협회 (TTA))

1. 서론

다음은 상호운용성 및 상호운용성시험과 관련된 발췌문으로 그 개념이 무엇인지 그리고 얼마나 중요한지를 보여준다.

“...IT 매니저들의 65%는 자사의 응용소프트웨어를 만드는데 있어서 최고의 우선 순위를 갖는 것이 상호운용성 문제라고 말한다...”¹⁾

“... 아주 공개적으로 적대적인 관계에 있는 그 두 회사 [마이크로소프트와 SUN]는 몇 개를 제외하고는 그들의 운영체제와 툴이 부드럽게 협력할 수 있도록 하기로 약속하였다...”²⁾

“... 상호운용성시험은 Win NT 개발시험의 2/3을 차지하고 있다... 빌 게이츠는 개발의 점점 더 많은 부분이 IBM의 CICS와 DB2등을 포

함한 다른 주요 제품들과의 상호운용성을 확보하는데 들어가고 있다고 말하였다...”³⁾

이와 같이 오늘날에는 상호운용성과 상호운용성시험에 대한 중요성이 널리 인식되어 있지만, 역사적으로는 네트워크의 발전과 멀티벤더환경을 지원하고자 하는 통신표준화기구, 통신사업자, 통신장비업계의 노력에 힘입어 상호운용성과 상호운용성시험의 중요성이 통신분야에서 일찍 인식되고 발전하였다. 그러나 이론적으로 상호운용성 문제가 발생하는 이유는 물리적인 통신장치, 통신소프트웨어나 일반 소프트웨어의 경우 모두 동일하므로 특히 오늘날과 같이 분산 소프트웨어가 지배적인 환경에서는 통신분야를 넘어서 상호운용성에 대한 이해와 상호운용성 시험 기술에 대한 연구가 보다 보편적인 개념으

1) “... Sixty-five percent of IT managers say their top priority for developing in-house applications is interoperability...”, Joe Wilcox, “Model for Interoperability”, <http://www.microsoftmonitor.com/archives/002639.html>, 2004.

2) “... The two companies [Microsoft and SUN], long at each others' throats in a decidedly public manner, have agreed to disagree on some things but pledged to ensure that their operating systems and tools work together smoothly...”, Steven Burke, “News Analysis: Sun and Microsoft's Interoperability Pact”, <http://www.linuxpipeline.com/trends/18900168>, 2004.

3) “... Interoperability testing is taking up to two-thirds of development testing on Win NT. ... Bill Gates said an increasing part of development is done to ensure interoperability with other major products, including IBM's CICS and DB2...”, Steven Burke, “Gates Touts NT's Importance,” <http://www.techweb.com/wire/news/june/0603gates.html>, 1997.

로 접근되어 그 결과가 다른 분야에서 공유되어야 할 것이다.

상호운용성시험보다 그 필요성이 일반적으로 더 잘 알려진 시험으로 적합성시험은 어떤 구현이 그 명세에 대하여 올바른가를 시험한다. 적합성시험은 이를 통하여 구현의 오류를 발견하고 명세에 의하여 정의된 구현들이 상호운용하기 위한 필요조건으로서 널리 수행되고 있다. 그러나 적합성시험은 두 개의 중요한 제약이 있다. 첫째, 명세가 보통 불완전하거나 오류가 포함되어 있어 상호운용하여야 할 적합한 구현들도 서로 상호운용하지 않을 수 있다. 둘째로, 명세가 완전하고 무결하다고 하여도 시간, 적합성시험의 방법론적 한계에 의하여 보통 완전한 적합성시험이 불가능하다. 따라서 상호운용하여야 할 구현들이 적합성시험을 통과하고도 실제에 있어서 상호운용하지 않는 경우가 자주 발생하게 된다. 따라서 서비스를 위하여 구현을 배치운영하기에 앞서 상호운용성시험을 수행하는 것은 필수적이다.

통신프로토콜분야에서의 상호운용성 시험기술에 대한 연구는 1990년으로 거슬러 올라간다. Rafiq 등의 연구[1]가 상호운용성시험의 필요성을 제시하였다면, 구체적으로 시험도출방법을 제시하는 연구는 Arakawa, Luo 등의 연구에서 최초로 제시되었다 [2] [3]. 국내에서는 Kang, Kim의 연구가 국제적으로 발표된 상호운용성시험에 관련된 최초의 논문이다 [4]. 상호운용성시험의 이론적 연구는 국내에서 많이 연구가 수행되었고 지금에도 우리나라가 다른 나라들 보다 활발한 이론적 연구를 수행하고 있다 [4][7][8][10][11].

본 논문에서는 상호운용성시험기술에 대한 이론과 실체를 소개한다. 제 II장에서는 상호운용

성시험기술의 이론적 체계를 국내연구를 중심으로 소개하고, 제 III장에서는 상호운용성시험의 현장적용과 현장에서 적용되는 시험체계를 소개한다. 제 IV장에서는 이론적 관점에서 상호운용성시험의 실체를 보았을 때 드러나는 몇 가지 문제점을 논의한다. 마지막으로 제 V장에서는 상호운용성시험에 있어서 이론과 실체가 나아가야 할 바람직한 방향을 논의한다.

II. 상호운용성 시험의 이론

앞 장에서 언급한 바와 같이 상호운용성시험의 이론적 연구는 국내에서 비교적 활발히 진행되었고 논문 [10]에서는 “상호운용성시험 체계와 방법론”이라는 제목으로 실용적 적용을 위한 기본체계로서 정리소개되었다. 이 장에서는 [10]의 상호운용성시험이론을 간략히 살펴본다.

1. 상호운용성

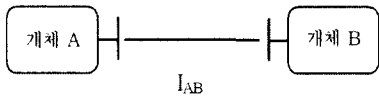
두 개 이상의 개체(entity)가 어떤 기능 혹은 임무를 수행하기 위하여 정보를 전달, 교환, 공유 혹은 협력할 때 그러한 능력을 상호운용성(interoperability)이라고 부른다. 분산소프트웨어와 통신프로토콜에서와 같이 상호운용성은 상호운용에 참여하는 개체들에 필수적인 성질로 간주된다. 여기서 (1) 둘 이상의 개체의 참여와 (2) 개체들이 함께 기대되는 대로 행위하는 것은 상호운용의 두 가지 본질적 특성이다.

2. 상호운용성시험

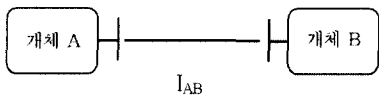
상호운용 정도의 측정은 두 가지 차원에서 수행될 수 있다. 하나는 명세(specification) 차원이



〈그림 1〉 실세계 혹은 소프트웨어시스템의 구조를 나타내기 위한 기본 구성체



〈그림 2〉 객체 A와 B가 연관된 구조

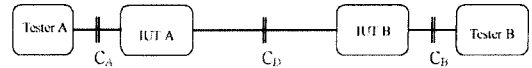


〈그림 3〉 Peer-to-peer 구조의 예

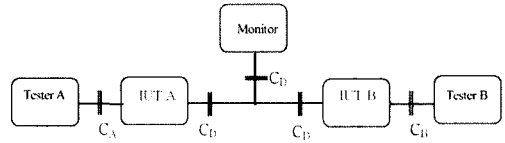
고, 다른 하나는 구현(implementation) 차원이다. 명세차원에서 상호운용의 측정을 상호운용성검증(interoperability validation)이라고 부르고 구현에 대하여 상호운용의 측정을 수행하는 활동을 상호운용성시험(interoperability testing)이라고 부른다. 상호운용성검증은 구현을 위한 명세 자체가 그 명세에 기반한 구현이 상호운용성을 가지도록 올바르게 되어 있는지를 측정한다.

3. 접속, 개체, 연관

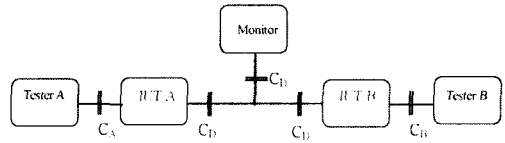
실세계의 개체들은 고립된 개체로서 존재할 수도 있지만 어떤 개체들은 서로 관련(association)지워져 있고 그들 간에 어떤 상호작용이 있다. 이 상호작용의 방식을 접속(interface)이라고 부른다. 소프트웨어시스템도 마찬가지로 서로 관련지워져 상호작용하는 개체들로 구성된다. 보통 소프트웨어 시스템에서의 개체들은 서로 고립되어 있지 않고 상호작용을 통하여 궁극적인 시스템기능 혹은 시스템의 서비스를 창



(a) 상호운용성 시험구조 I



(b) 상호운용성 시험구조 II



(c) 상호운용성 시험구조 III

〈그림 4〉 peer-to-peer시스템을 위한 세 개의 상호운용성 시험 구조

출해 낸다. 이 개체들은 그 안에 더 세부적인 조직을 가진 구성시스템(subsystem)일 수도 있고 더 이상 구성시스템으로 볼 수 없는 원자적 개체일 수도 있다. 소프트웨어 개체들도 실세계와 마찬가지로 접속을 통하여 상호작용하게 된다. 따라서 그림 1과 같은 기호를 사용하여 실세계 혹은 소프트웨어시스템을 모델링할 수 있다.

이과 같은 개념과 기호로서 모델링할 수 있는 가장 단순한 종류의 상호운용의 구조가 그림 2로 두개의 개체는 오직 서로간의 상호작용을 위한 접속만을 갖고 있는 경우이다. 그림 3은 이보다 약간 더 복잡한 구조로 두 개의 개체가 각각 2개의 접속을 갖고 있으면 2개 중 하나의 접속을 통하여 서로 상호작용하는 경우를 보여주고 있다.

4. 상호운용시험구조

시험기(tester)는 (완전한) 시험 시스템의 구성

요소이다. 시험기에는 두 가지 종류가 있다. (1) 능동적 시험기와 (2) 수동적 시험기이다. 능동적 시험기를 “시험기” 그리고 수동적 시험기를 “모니터”라고 부른다. 모니터는 단지 관찰하므로 상호작용에는 영향을 주지 않지만(능동적) 시험기는 관찰할 수도 있고 상호작용을 간섭할 수도 있다. 각각의 시험기는 대상 시스템의 행위에 대하여 그것이 보는 관점에서 올바른에 대하여 판단을 내릴 수 있다. 시험의 대상이 되는 소프트웨어개체를 시험대상구현(IUT: Implementation Under Test)이라고 부른다.

실제시스템은 다양한 구조로 존재하고 따라서 시험을 위하여 그 동작을 제어/관찰할 수 있는 구조도 다양하다. 즉 상호운용성시험구조는 시험대상시스템의 구조를 밀접하게 반영하지만 다양한 주어진 대상시스템에 대하여 적용할 수 있는 다양한 상호운용성 시험구조(interoperability test architecture)가 존재한다. 예를 들어 그림 3의 대상시스템의 구조에 대하여 여러 가지 상호운용성 시험 구조 가운데 특히 그림 4의 3개를 생각할 수 있다. 그림 4에서 (a)는 2개의 시험기를 사용한 상호운용성 시험구조, (b)는 2개의 시험기와 모니터를 사용한 상호운용성 시험구조, (c)는 3개의 시험기를 사용한 상호운용성 시험구조이다.

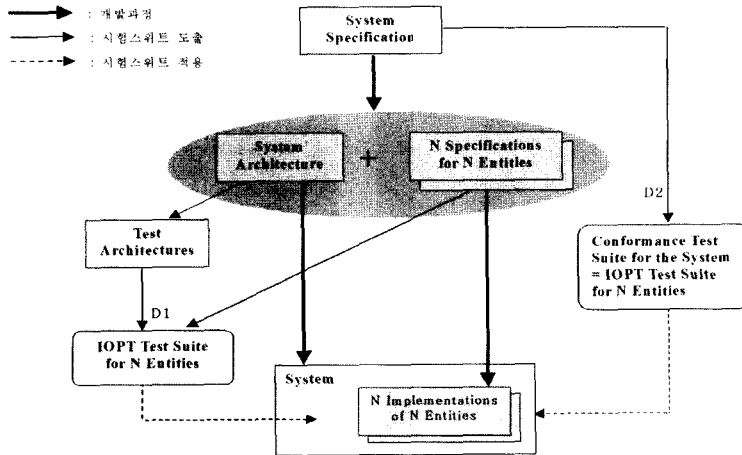
세 개의 구조는 두 개의 IUT 사이의 상호작용을 제어하고 관찰하는 서로 다른 능력을 갖고 있다. 구조 I은 IUT A와 IUT B사이의 상호작용에 대한 어떠한 관찰이나 제어도 수행하지 않는다. 반면에 구조 II는 상호작용을 단지 관찰하고, 구조 III은 상호작용에 대하여 관찰과 제어를 모두 수행한다. 구조 II에서 모니터는 상호작용을 관찰하지만 두 개의 IUT 사이의 상호작용에 영향을 주지 않는다. 따라서, 구조 II는 구조 I보다 더 강력하다. 구조 III에서 시험기 D는 두 개의 IUT

사이에 상호작용을 간섭(interference)할 수 있다. 예를 들어, 링크를 오가는 메시지를 가로챌 수 있고, 지연, 손실, 메시지의 반복을 시뮬레이션 할 수 있고 네트워크(network cloud)를 시뮬레이션 할 수도 있기 때문에, 두 개의 IUT 사이에 폭 넓은 범위의 시험시나리오를 수행할 수 있다. 따라서 구조 III은 가장 강력한 시험구조이고 다른 두 개의 구조가 갖는 능력을 모두 갖고 있다.

실제시험을 수행하기 위하여 이러한 다양한 시험구조 가운데 시험에 대하여 기대되는 완성성 및 시험에 할당할 수 있는 시간, 비용을 고려하여 적절한 시험구조를 선택하게 된다. 그림 4의 시험기 구현에는 많은 시간과 노력이 소요되므로 실제로 Tester A 또는 B 역할을 하는 시험기는 상용화된 사례는 드물고 대부분의 상용 시험기는 모니터이다. 따라서 현실적인 방안으로 그림 4의 (b)와 같은 시험구조가 많이 채택되고 있으며, 이 경우에도 시험을 위한 별도의 시험기를 사용하는 대신 IUT 간의 인터페이스에서 시험 시나리오에 해당하는 이벤트가 발생되도록 하는 자극을 각각의 IUT 상황에 따라 생성하여 시험을 진행하고 시험 결과의 분석은 모니터에 의존한다.

5. 상호운용성 시험스위트 도출

제 1절에서 기술한 상호운용의 정의는 “기대되는 행위”만을 언급하고 그것이 무엇인지 자세히 기술하지 않았다는 의미에서 ‘열린’ 정의이다. 그 이유는 일반적으로 기대되는 상호작용이 시스템마다 다르기 때문이다. 그러므로, 상호운용을 위하여 요구되는 사항은 원칙적으로 구현의 명세를 통하여 명시적으로 주어지거나 명세 작성자와 구현자 사이에 묵시적으로 동의되어



〈그림 5〉 상호운용성 시험도출을 위한 두 가지 접근방법

야 한다. 또한 분산응용이나 통신프로토콜의 상호운용성에서 보통 그러한 기대는 단일한 문서를 통하여 명시적으로 주어지지 않으므로 관련된 명세로부터 유추되거나 도출되어야만 한다.

이러한 점을 고려하고 상호운용의 관점에서 기대되는 행위가 시스템의 명세(specification)에 충분히 상세히 정의되어 있다고 가정하자. 상호운용성시험스위트는 소프트웨어 개발의 여러 단계에서 생산되는 시스템명세, 설계명세, 구현과 같은 개발산출물(artifacts)들로부터 도출될 수 있다. 설계명세는 시스템구조와 시스템의 N개의 구성개체들에 대한 N개의 명세로 이루어진다고 볼 수 있다.⁴⁾ N개의 명세는 시스템 명세에서부터 설계결정에 의하여 구현을 개발하는 과정에서 얻어진 것일 수도 있고, 시스템 명세와 독립적으로 주어진 경우도 있다.⁵⁾

상호운용시험스위트를 도출하기 위하여 그림

5에서처럼 두 개의 경로가 가능하다. 먼저 시험도출접근방법 D1은 시스템구조에서부터 하나 이상의 시험구조를 선정하고 이를 기반으로 N개의 개체에 대한 N개의 명세에서부터 상호운용성시험스위트를 도출한다. 이 시험스위트는 주어진 시스템구조하의 N개의 개체의 상호운용성시험을 위하여 적용된다. D2의 경우 구현은 비록 굵은 화살표와 같이 설계과정을 거쳐 이루어진다 하더라도, 원래의 시스템명세에서부터 시험스위트가 도출된다. 이 경우 시험구조는 전체시스템에 대한 적합성시험구조와 다를 바가 없다. 즉 시스템 내부의 어떤 구성개체가 존재하며 어떤 구조를 형성하고 있는지 알지 못하므로 시험스위트 도출에 반영시킬 수 없다. 따라서 이 시험스위트는 시스템에 대한 적합성시험스위트이다. 다른 한편 이를 동시에 N개체들에 대한 상호운용성시험스위트로 볼 수도 있다. 상호운용성

4) 각각의 구성개체는 다시 다른 개체들로 구성된 내부구조를 가질 수 있다. 논점을 명확히 하기 위하여 여기서는 시스템과 구성개체들이라는 2계층 구조만을 고려한다.

5) 구현프로그램을 상호운용성시험스위트 도출의 소스로 보는 것도 불가능하지 않지만, 이 경우 소스코드 의존적인 시험스위트가 나와, 개발오류를 지정하는 능력이 떨어질 수 있고, 도출과정이 어려울 수 있는 단점이 있다.

험구조 가운데에 구성개체 간의 접속들을 무시하고 단지 구성개체와 시스템외부와의 접속만을 고려하는 구조를 생각할 수 있는데 시스템 적합성시험은 상호운용성시험과 구별되지 않는다.⁶⁾ 따라서 D2의 경우 N개의 개체에 대한 상호운용성시험 문제를 시스템 적합성시험의 문제로 볼 수 있고, 적합성시험의 방법들을 적용할 수 있다.

최초의 명세가 시스템명세 형태로 주어지지 않고, 설계명세 형태로 주어졌다면, 그리고 시험도출을 위하여 설계명세로부터 특정구조를 가진 구조관점에서 내부를 들여다 볼 수 있는 시스템명세를 먼저 도출하였다면, N개체의 상호운용성시험을 위하여 역시 적합성시험 접근방법을 취하고 있는 것이다.

6. 시험절차와 판정

제 5절에서와 같이 시험스위트가 도출된 후에는 시험실행에 들어가게 되는데 이를 위하여 시험절차의 문제, 시험판정의 문제등이 고려되어야 한다. 관련된 상세한 내용이 논문 [10]에 나와 있다.

III. 상호운용성 시험의 실제

상호운용성시험은 상설시험기관이 상호운용성시험을 주관하여 수행하는 경우와 여러 벤더들이 특정종류의 제품을 다른 벤더들의 같은 제품과 상호운용하는지 시험하기 위하여 모여서 수행하는 상호운용성시험이벤트로 나누어 볼

수 있다. 그러나 실제로는 상설시험기관의 경우에도 시험 기관이 기술적, 절차적인 관리를 진행한다는 점에서 차이가 있을 뿐 상호운용성시험의 기술, 절차의 내용적인 측면에서는 상호운용성시험을 이벤트와 다른 없는 형태로 진행하고 있다. 따라서 이 장에서는 먼저 제 1절에서 국내외에서의 주요 상호운용성시험 이벤트에 대하여 간략히 살펴보고 그리고 제2절에서는 이들에 사용되는 기술과 절차에 대하여 살펴본다.

1. 상호운용성시험 이벤트

대표적인 상호운용성시험 상설기관으로는 국내에서는 한국정보통신기술협회(TTA)가 있고 국외에서는 University of New Hampshire의 InterOperability Lab (UNH-IOL)과 European Telecommunications Standards Institute (ETSI)가 있다.

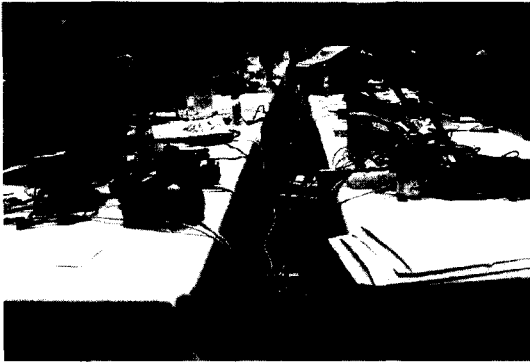
· TTA

TTA의 네트워크시험센터에서는 네트워크 장비의 하드웨어, 소프트웨어의 상호운용성시험을, 수행하고 있다. TTA가 주관하여 1992년부터 InterOperable Network (ION)로 매년 개최되어 오던 ION 워크샵행사가 2002년부터는 개최되는 Interoperability On! 이라는 명칭으로 변경되며 다양한 상호운용성시험이벤트를 연 중 수시로 개최하고 있다. ION에서 최근 관심의 대상이 되는 기술은 IPv6, 블루투스, 무선랜, 데이터 방송등이다.

· UNH-IOL

1998년 설립되어 발전한 세계적으로 대표적인 상설 상호운용성시험실로 멤버로 가입하면

6) 상호운용성시험과 적합성시험의 관계에 대하여는 3절을 참조.



〈그림 6〉 ETSI Plugtest 현장⁷⁾

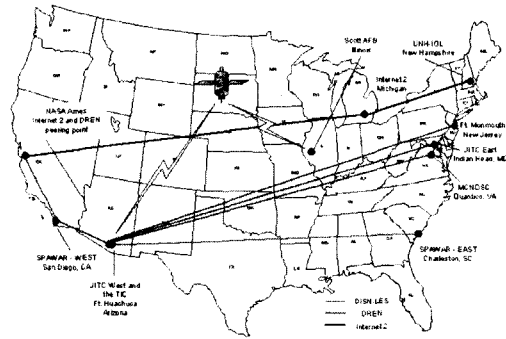
상시 시험서비스를 받을 수 있다. 상호운용성 시험을 위하여는 참여자들로 구성된 컨소시엄을 구성운영한다. 컨소시엄에서 구축한 테스트베드는 UNH-IOL의 소유로 되어 상호운용시험대상이 확대되면서도 지속적인 시험이 가능하도록 하고 있다.

· ETSI Plugtest™

유럽지역표준제정기구인 ETSI 의 서비스로 1999년부터 제공되기 시작하여 현재까지 40여 차례의 상호운용성시험이벤트를 제공하였다. “bake-off” 혹은 “interop”라는 이름으로도 알려져 있고 인증은 하지 않는 기술이벤트이다. ETSI 에서는 이러한 시험기술 및 절차를 자체표준[9]으로 만들었고 이를 궁극적으로 ITU-T표준으로 만들려는 노력을 하고 있다. 그림 6은 Plugtest의 실제상황을 보여주는 사진이다.

· MoonV6

Moonv6 는 북미지역의 IPv6의 배치를 활성화 시키기 위한 과제로 UNH-IOL, Department of Defense(DoD), North American IPv6 Task Force (NAv6TF) 를 중심으로 많은 벤더와 서비스회사들이 참석하여 아주 실제에 가까운 테스트베드



〈그림 7〉 MoonV6 테스트네트워크⁸⁾

를 구축하여 시험을 수행하고 있다. 그림 7는 시험에 참여하는 대상들이 미국전역에 분포되어 있음을 보여준다.

시험대상으로는 ICMP, TCP, UDP 및 핵심 IPv6 기능을 포함하는 기본규격 상호운용성을 비롯하여 라우팅, 이동성, 그리고 DNS, NFS, SSH, FTP 같은 공통네트워크응용과 네트워크 견고성, 보안성등을 시험한다.

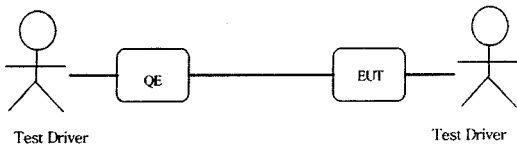
· OMG Corba 상호운용성시험

Object Management Group (OMG)에서는 분산객체기술을 발전시키고 CORBA 제품 사이의 상호운용성을 시험할 목적으로 1997년 Distributed Object Promotion Group (DOPG) 를 설립하였다. DOPG 은 설립 당시 일본에서 분산 객체제품을 공급하는 14개의 컴퓨터 및 소프트웨어 벤더로 구성되었다. DOPG 안에는 4개의 작업그룹 중 NEC가 의장을 맡고 있는 상호운용성 작업그룹은 지금까지 여러 차례에 걸쳐 IOP 상호운용성시험을 수행하였다.

이러한 시험을 통하여 (1) Typecode의

7) <http://www.etsi.org/plugtests/>

8) <http://www.moonv6.com/>



〈그림 8〉 기본적인 상호운용시험 구조

encoding 혹은 간접지칭 그리고 복잡한 Typecode가 지원여부 문제, (2) 명세 버전간의 상호운용성문제, (3) 기타 벤더특정의 행위의 문제점들이 발견되어 정정되었다. OMG 상호운용성시험은 통신분야가 아닌 분산소프트웨어 분야에서도 상호운용성시험의 필요성이 인식되고 적용되기 시작한 중요한 예이다.

2. 상호운용성시험이벤트를 위한 기술 및 절차

ETSI에서는 상호운용성시험이벤트에 실제 적용될 수 있는 체계적이고 따라서 반복적인 시험을 가능하게 하는 절차를 상호운용성시험표준안 [9]에서 제시한다. 그림 8은 가장 단순한 상호운용시험구조를 보여준다. Equipment Under Test (EUT)는 시험대상을 지칭한다. 상호운용성시험에서는 항상 단일한 EUT가 존재한다. EUT는 단말기와 같은 사용자 장비, 라우터와 같은 네트워크장비 혹은 소프트웨어 응용이 된다. Qualified Equipment (QE)는 EUT와 같이 사용자장비, 네트워크장비 혹은 소프트웨어 응용이며 단지 차이점은 이미 다른 유사한 장비와 상호운용하는 것으로 증명된 장비이다. 시험구조에서 복수개의 QE가 존재할 수 있다. 그 밖에도 [9]에서는 Plugtest에서 사용하는 용어들의 정의를 내리고 시험스위트의 템플레이트와 시험절차를 제시한다.

이와 같은 상호운용성시험 절차의 한 사례로

무선랜에 대한 Wi-Fi 인증을 위해서 사용되는 상호운용성시험절차 및 시험스위트를 들 수 있다[12]. 이 시험에서는 성능과 기능이 검증된 기준장비를 지정하고 그들 장비와의 상호운용에서 일정 수준 이상의 성능을 보이면 시험에 통과하는 것으로 규정되어 있다.

시험구조가 위와 같이 QE를 1개 이상 반드시 포함한다면 최초의 QE는 존재하게 되는가 하는 의문이 생긴다. ETSI 방법의 답은 최초의 경우는 QE도 단지 EUT와 다른 하나의 EUT일 뿐이라고 한다. 이 두 개의 EUT가 상호운용에 문제가 없을 때 두개 모두 QE로서 자격을 갖게된다. Test Driver는 Point of Control and Observation (PCO)에 해당되는 개념으로 자연인이 될 수도 있고 자동화된 소프트웨어일 수도 있다.

ETSI 접근방법의 특징은 상호운용성 이론적인 기초를 제공하기 보다는 실용적인 상호운용성시험의 체계화를 제공하는데 있다. 예를 들어 상호운용성시험에서도 EUT는 오직 하나 있는 것으로 보아 한편으로는 시험관정을 용이하게 하고 문제점파악의 효율성을 기하고 있다. 한편 이 경우 적합성시험에서의 Tester와 QE가 어떤 차이점이 있는 것인지 또한 만일 ETSI가 적합성시험을 접속시험으로 본다면 ETSI에서 말하는 상호운용성시험이 개체시험과 무엇이 다른 것이니 명확한 설명이 있어야 할 것이다.

IV. 상호운용성시험과 관련된 오해하기 쉬운 개념들

1. 모호한 시험대상의 문제

시험에서 “시험의 대상(IUT: Implementation Under Testing)”의 개념은 매우 중요하다. 시험

대상이란 “현재시험의 대상이 되는 부분”을 말한다. 시험에서 대개의 경우 시험을 위하여 현재 관심의 대상이 되는 부분뿐만 아니라 다른 구현도 이용하게 된다. 이때 관심의 대상이 되는 부분 뿐만 아니라 시험기와 상호운용에 사용된 전체 시스템을 System Under Test (SUT)라 한다. SUT 내에서 어떤 부분이 관심의 대상이 되는 IUT 인지가 명확하지 않은 경우 그리고 관심의 대상이 되는 부분을 제외한 부분에서 오류가 있을 경우 시험에 실패했을 때 문제발생의 원인을 찾는 작업이 어려워진다. 상호운용성시험은 ITU가 하나가 아니라는 것이 적합성시험과 근본적인 차이점이고 따라서 상호운용성시험의 대상이 되는 IUT를 모두 정확히 열거하는 것은 더욱 중요하다.

상호운용성시험에 있어서 일반적으로 “장비시험(Equipment Testing)”과 같은 모호한 표현이 많이 사용되는데 이것은 장비시험에 많은 종류의 시험이 필요하다는 점을 간과하게 하고, 장비 가운데 실제로 어떠한 것을 시험대상으로 보는 것인지를 불명확하게 한다.

예를 들어 장비시험을 위하여는 물리적인 장치의 연결이 정확한지를 검증한 후에는 기본적인 소프트웨어 통신인프라가 정확히 구현설치가 되었는지를 검증하고 최종적으로 그러한 인프라를 사용하는 응용이 하드웨어와 소프트웨어인프라와 올바르게 상호작용하는지의 검증이 장비시험에 포괄된다. 보통 이러한 상황은 아주 포괄적이어서 우리가 관심을 가지는 IUT가 예를 들어 응용이라고 하여도 올바른 시험절차는 다른 사항에 대한 검토도 요구하게 된다.

그러나 일반적으로 장비시험이라고 할 때에는 너무 모호하고 광범위하여 IUT에 대한 구체적인 언급이 반드시 이어져야 한다.

2. 상호운용성 시험과 혼합시험

시험대상의 기능시험을 위하여 적합성시험과 상호운용성시험 외에도 이 두가지의 시험을 병행하는 시험방법인 “혼합시험”도 가능하다. 실제로 현장에서 상호운용성시험을 수행하며 어느정도 적합성시험을 수행하게 되는 것이 보통이다. 이 때에 수행되는 적합성시험은 엄밀하게는 상호운용성시험으로 볼 수 없으나 보통 이를 분리해내지 않는 경향이 있다. 그 이유는 시험수행자가 상호운용성과 혼합시험을 혼동하는 이유가 있고, 나아가 혼합시험이 효율적이기 때문에 이러한 혼동이 고착화 되는 경향도 있다.

적합성시험이 단일 IUT를 대상으로 하는데 반하여 상호운용성시험이 두 개 혹은 그 이상의 IUT를 대상으로 하므로 엄밀히 말하여 적합성시험항목들의 집합인 적합성시험스위트와 상호운용성시험항목들의 집합인 상호운용성시험스위트는 서로 공통부분이 없게 된다. 이렇게 적합성시험과 상호운용성시험이 서로 분리되어(disjoint) 있음에도 불구하고 이 두 가지의 시험을 포함하는 혼합시험을 사용하게되는 이유는 기본적으로 다음과 같은 자연스런 시험도출방법이 있기 때문이다.

- ① 시스템에 대한 시험항목생성과정에 적합성 시험항목을 버리지 않고 정당한 시험항목으로 보전한다
- ② 상호운용성시험항목을 두 개 혹은 그 이상의 적합성시험항목의 결합으로 본다.

혼합시험의 이론과 그것이 효율적인 이유는 논문 [7]에서 상세히 제시하고 있다.

3. 접속시험과 개체시험의 구별

개체는 독자적으로 동작하거나 혹은 환경으로부터 자극을 받아서 동작하는 소프트웨어 단위이다. 반면에 접속(interface)은 개체가 다른 개체들과 상호동작하는 “창문”이다. 현재 많은 프로토콜의 표준화된 명세가 접속을 정의하고 있으나 실제 구현은 개체로서 존재하게 된다. 개체의 완전한 행위를 알기 위하여는, 그 개체의 모든 관련된 접속에서의 행위와 그들간의 관계를 알아야 한다.

적합성시험의 경우 보통 접속의 명세에 대하여 시험을 수행하는 것으로 충분하나, 상호운용성시험을 위하여는 접속단위에서 시험도출을 고려하는 것으로 충분하지 않고 복수개의 접속으로 이루어지는 개체를 고려하여야 한다. 이 경우 특히 한 개체의 복수개의 접속사이의 상관관계는 시험도출에 어려움을 제기하고, 요구되는 개체행위를 먼저 도출하는 것이 올바른 상호운용성 시험도출의 전제조건이 된다.

논문 [6]은 접속시험과 개체시험에 대한 개념적 구분과 개체시험의 이론적 필요성을 보여주고 있다.

V. 결론

지금까지 상호운용성시험의 이론과 실제에 대하여 논의하였다. 이 논문의 상호운용성의 정의가 매우 광범위한 대상에 적용되듯이, 상호운용성시험은 매우 광범위하게 적용될 수 있는 시험이다. 그러나 지금까지는 한편으로는 통신프로토콜의 영역에서 다른 한편으로는 소프트웨어 시험 중 통합시험이라는 분야로서 독립적으로 연구되어 왔고 따라서 이 두 분야는 서로 상대부

야의 연구결과를 수용하지 못하였고 이 두 분야가 어떤 점에서 공통점이 있고 어떤 점에서 서로 다른지에 대한 이해도 부족한 상태이다.

상호운용성 시험이론은 상호운용성 시험의 실제에서부터 어떤 부분에 대한 체계적인 기술과 절차가 현실적으로 많이 요구되는 지의 정보와 이론적 연구가 필요한 대상의 우선순위에 대한 정보를 얻을 수 있다. 상호운용성시험의 현장적용은 제 IV장에서 논의한 바와 같은 오해나 혼란을 지금도 그대로 갖고 진행이 되고 있다. 상호운용성시험의 이론과 실제의 만남은 세계적으로도 그다지 많은 연구가 이루어지지 않는 분야로 그 동안 국내 연구결과가 국제적으로 비교적 큰 비중을 차지 하고 있고, 우리나라가 비교적 쉽게 선두주자로 나설 수 있는 분야이다. 따라서 앞으로 이 분야의 국내 전문가 그룹간의 협력이 기대되며, 이론과 실제, 연구와 현장적용 간에 활발한 교류가 크게 기대된다.

참고문헌

- [1] Rafiq, O. and Castanet, R., “From Conformance Testing to Interoperability Testing,” The 3rd International Workshop on Protocol Test Systems, 1990.
- [2] Arakawa, N., and Soneoka, T., “A Test Case Generation Method for Concurrent Programs,” Protocol Test Systems, IV, J. Kroon, R.J. Heijink and E. Brinksma (Eds.), Elsevier Science Publishers B. V. (North-Holland), 1992.
- [3] Luo, G., Bochmann, G., and Petrenko, A., “Test Selection Based on Communicating Nondeterministic Finite-State Machines Using a Generalized Wp-Method,” IEEE Transactions on S.E., Vol. 20, No. 2, Feb. 1994.

- [4] Kang, S., and Kim, M., "Test Sequence Generation for Adaptive Interoperability Testing," IFIP TC 6/WG 6.1 The 8th International Workshop on Protocol Test Systems, Evry, France, September 1995.
- [5] ITU-T Recommendations X.290 Series: Conformance Testing Methodology and Framework Parts 1-7, 1995.
- [6] Sungwon Kang, "Relating Interoperability Testing with Conformance Testing," IEEE Global Communications Conference (Globecomm '98), Sidney, Australia, 8-12 November 1998.
- [7] 강성원, 신재휘, 김규형, "적합성과 상호운용성의 혼합시험 접근방법," 한국 정보과학회 논문지.(A) 제 25권 제 8호, pp. 906-916, 1998. 8.
- [8] Sungwon Kang, Jaehwi Shin, Myungchul Kim, "Interoperability Test Suite Derivation for Communication Protocols," Computer Networks - The International Journal of Computer and Telecommunications Networking, Vol. 32, pp. 347-364, March 2000.
- [9] ETSI, Interoperability Test Methods and Approaches: Part 1: Generic Approach to Interoperability Testing, ETSI TS 102 237-1, December 2003.
- [10] 강성원, 신재휘, 성종진, 홍경표, "소프트웨어 상호운용성 시험 체계와 방법론," 정보과학회논문지: 소프트웨어 및 응용 4월호, 제31권 제4호, 2004년 4월.
- [11] Soonuk Seol, Myungchul Kim, Samuel T. Chanson, Sungwon Kang, "Paper title: Interoperability Test Generation and Minimization of Communication Protocols Based on the Multiple Stimuli Principle", To be published in IEEE Journal on Selected Areas in Communications: special issue on Design, Implementation and Analysis of Communication Protocols.
- [12] WECA, "Wi-Fi System Interoperability Test Plan", December 2001.

저자소개



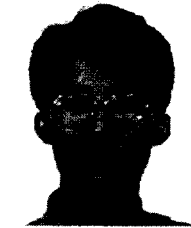
강성원

1982년 서울대학교 사회과학대학 졸업
 1989년 미국 University of Iowa 전산학 석사
 1992년 미국 University of Iowa 전산학 박사
 1993년-2001년 한국통신 연구개발본부 선임연구원
 2001년-현재 한국정보통신대학교 조교수
 2002년-현재 미국 Carnegie-Mellon University 소프트웨어공학석사과정 겸임교수
주관심분야 소프트웨어구조, 소프트웨어시험



장웅

1981년 홍익대학교 전기공학과 학사
 1986년 충남대학교 전자공학과 졸업
 1990년 충남대학교 전자공학과 석사
 1999년 충남대학교 통신공학 박사과정 수료
 1990년-2001년 한국전자통신연구원 근무
 1994년-1995년 미국 NIST 객원연구원
 2001년-현재 한국정보통신기술협회 근무
주관심분야 프로토콜 시험, 유선네트워크장비시험



박용범

1986년 경북대학교 전자공학과 졸업
 1989년 KAIST 전산학 석사
 1989년-2001년 ETRI 팀장
 1995년-1996년 미국 국립표준연구원 객원연구원
 2001년-현재 TTA 디지털통신시험실장
주관심분야 프로토콜공학, 성능 평가, 홈네트워킹