

소프트웨어 컴포넌트 시험기술 동향

김남희 | TTA IT시험연구소 S/W시험인증센터
전임연구원

신석규 | TTA IT시험연구소 S/W시험인증센터
센터장

1. 개요

컴포넌트 기반의 S/W 개발은 산업계에서 성장하고 있는 중요한 분야 중의 하나이다. 재사용성과 유지 보수성이 우수한 컴포넌트는 e-Business가 활성화 되면서, 급속하게 변화하는 환경과 복잡한 비즈니스 로직 구현 등의 요구사항을 만족 시켜줄 수 있는 대안으로 산업계에서 활성화되어 사용되고 있다. 그러나 컴포넌트를 구입하여 기존의 레거시 시스템과 상호 운용하거나 컴포넌트 기반의 시스템을 구축하는데 있어 구입한 컴포넌트의 품질을 평가하기 위한 인증제도가 부족한 상태이다.

현재 국내·외적으로 컴포넌트 개발 산출물의 종류 및 양식, 개발 산출물 품질평가 기준, 컴포넌트 제품 품질시험 및 인증방안 등 여러 분야에서 연구가 활발히 진행되고 있지만, 아직은 초기 연구 단계로서 컴포넌트 품질 평가 시험 방안에 대해 제시된 표준이 부족한 상태이다.

이에 본고는 최근의 S/W 컴포넌트 기술 및 시험기술을 분석하고, TTA에서 ISO/IEC 9126 표준과 S/W 컴포넌트의 고유한 특징을 반영하여 개발한 S/W 컴포넌트 품질 평가 모델을 소개하고자 한다.

2. S/W 컴포넌트 기술

S/W 컴포넌트에 대한 효과적인 평가를 위해 먼저 S/W 컴포넌트 기술의 파악이 선행되어야 한다. 이에 최근까지 제시된 주요 컴포넌트를 분류하여 소개하고자 한다.

○ CORBA COMPONENT MODEL(CCM)

CCM은 CBD(Component-based Development) 컴포넌트를 위한 특성들을 제공하며, 다른 대표적인 참조 모델을 포함하는 포괄적인 모델이다. 각 컴포넌트는 레퍼런스 및 인터페이스 집합을 갖게 된다. CCM 컴포넌트에서의 Port는 컴포넌트의 클라이언트와 정의된 Port들을 연결 짓는 통신 채널의 개념으로 Facet, Receptacle, Event, Attribute의 4가지 타입의 매커니즘을 제공한다.

CCM 컴포넌트 모델의 주요 특징은 (1)분산 컴포넌트 지향 모델이며, (2)산업계 최초의 다중 언어 컴포넌트 표준이라는 것이다. 즉 CCM 모델은 다중언어와 다중 운영체제, 다중 벤더들의 여러 플랫폼에서 동일하게 사용된다.

○ ENTERPRISE JAVABEANS(EJB) 모델

엔터프라이즈 자바빈즈(EJB)는 자바 프로그래밍 언어로 컴포넌트 기반의 분산된 비즈니스 애플리케이션을 개발하고 배포하기 위한 컴포넌트 아키텍처로서, EJB 아키텍처를 사용해서 개발된 애플리케이션은 대규모적이고 사용자가 증가할 경우에 안정성을 책임진다.

EJB의 주요 특징은 (1)분산된 트랜잭션을 지원한다는 것이며, (2)빈 제공자가 비즈니스 메소드로부터 보안 문맥에 접근하기 위해서 사용할 수 있는 간단한 인터페이스를 제공하며, (3)콜백이 타임기반의 이벤트들에 대해 스케줄되어지는 것을 가능하게 하는 타이머 서비스를 제공한다는 것이다.

○ COMPONENT OBJECT MODEL(COM)

COM은 소프트웨어 컴포넌트들이 객체의 자격으로 다른 객체들과 대화할 수 있도록 하는 바이너리 표준

으로 Microsoft의 모든 객체지향기술의 기반이 되는 모델이다. COM 구조는 Interface와 Implementation이 저장된 장소로서 Persistent Storage, Intelligent Names(Monikers), Uniform Data Transfer로 구성되어 있다.

Microsoft가 제시한 다른 컴포넌트 모델로는 네트워크 상에서 다른 컴퓨터에 있는 서버 프로그램 객체에 서비스를 요청할 수 있도록 하는 DCOM, 트랜잭션 서비스 등을 강화시킨 COM+, 미들웨어나 기업시장을 위한 다중 계층 프로그램 개념의 MTS, 객체지향 프로그래밍 기술 및 도구모임인 ActiveX 등이 있다.

○ .NET 모델

Microsoft .NET 플랫폼은 개발자에게 XML 웹 서비스를 만들고 이들을 서로 통합하는 데 필요한 기술을 제공한다. .NET 플랫폼은 비주얼 스튜디오 닷넷, 닷넷 엔터프라이즈서버, 닷넷 프레임워크, 닷넷 구축 블록 서비스, 서버/데스크탑/디바이스에서 실행되는 운영체제로 구성된다.

.NET은 몇 가지 핵심적인 특성을 제안하였는데, 먼저 현재 있는 코드를 최대한 활용하기위해 기존에 있는 COM 바이너리를 새로 만든 .NET 바이너리와 합칠 수 있게 하였다. 또한 예전의 COM과는 달리 교차언어 상속(Cross Language Inheritance), 교차언어 예외처리(Cross Language Exception Handling), 교차언어 디버깅(Cross Language Debugging)이 가능하도록 설계되었다.

3. S/W 컴포넌트 시험 기술 동향

3.1 시험 기술

국내 · 외적으로 컴포넌트의 품질을 검증하기 위해 많은 연구들이 수행되어져 왔다. 그러나 컴포넌트 개발 표준이 정의되어 사용되고 있는 반면 컴포넌트 시험을 위한 표준은 아직 정의되지 않았다. 또한 많은 컴포넌트 자동화 시험도구들이 개발되어 상용화되고 있으나 컴포넌트의 포괄적 품질을 검증하기 위한 도구는 거의 없다. 대표적인 컴포넌트 시험기술 연구를 정리하면 다음과 같다.

○ NIST/ITL(National Institute of Standard and Technology/IT Laboratory)

NIST에서는 컴포넌트를 상태(State)와 행위(Behavior)를 갖는 객체들(Objects)로 구성된 객체지향적 개념으로 정의하고, 별도로 개발된 컴포넌트들이 하나로 통합 운영될 시스템을 효율적으로 시험하기 위한 방법을 연구하였다. 컴포넌트의 상태 명세를 위해 유한상태도(Finite State Machine)를 사용하며 유한상태도는 데이터 흐름과 제어 흐름으로 구성되는데, 이 모든 흐름정보를 이용하여 테스트 케이스를 자동 생성하여 한 컴포넌트가 상호작용하는 다른 컴포넌트와 정상적으로 기능을 수행하는가를 검증한다.

○ CVC(Component Vender Consortium)

1998년 설립 이래 컴포넌트의 품질을 보증하기 위한 품질 인증 프로그램(CVC Logo Program)을 운영하고 있다. 이 프로그램은 컨소시엄 멤버가 제출한 컴포넌트를 평가하고 그 시험을 패스하는 경우 품질 로고를 부여한다. CVC에서 제시한 품질 규격은 (1)컴포넌트로 인한 Memory Corruption 문제가 없을 것, (2)컴포넌트로 인한 Memory Leak 현상이 없을 것, (3)해결할 수 없는 오류/예외상황이 없을 것, (4)컴포넌트 내의 모든 코드에 대하여 80% 이상의 시험을 완료

할 것 등을 그 내용으로 한다. 시험에는 Compuware사의 DevPartner 도구가 사용된다.

○ University of Western Australia and Murdoch University

호주에서는 컴포넌트를 잘 정의된 인터페이스를 갖는 소프트웨어로 정의하고, 제 3자 기관에서 컴포넌트를 시험하기 위해 개발사가 제공해야 하는 문서 및 도구를 제안하였다. 이 연구에서는 개발사가 제공해야 할 시험 명세서의 표준안과 시험 결과를 제시하고, 시험 명세서를 제 3자 기관이 직접 실행할 수 있도록 구현된 시험 패턴 검증기(Test-Pattern Verifier)를 개발하였다.

○ 한국소프트웨어컴포넌트컨소시엄

단위 컴포넌트에 대하여 개발자와 제 3자 시험기관에 의한 테스트로 이루어지는 2단계 테스트 평가 체계를 가정하고 시험기관이 개발자로부터 넘겨받아야 하는 과정에서 검토해야 할 테스트 자료 목록 및 내용, 그리고 시험기관에 의해 수행될 테스트 절차를 제시하였다. 또한 단위 컴포넌트 테스트 케이스 설계를 위하여 동치영역 분할 테스트 기법, 상태 전이 테스트 기법, 불변식 기반 테스트 기법을 이용하고, 단위 컴포넌트 통합 테스트 케이스 설계를 위하여 유즈케이스 분석기법을 이용하고 있다.

○ 한국전자통신연구원

컴포넌트 시험을 수행하기 위해 필요한 품질 요소들을 정의하고 효율적인 시험을 위해 필요한 자동화 도구를 개발하였다. 컴포넌트 평가를 위한 품질 요소는 기능성(품질 항목: 설치, 기능의 존재, 정확성, 상호운용성), 사용성(품질 항목: 사용의 용이성), 효율성(품질

항목: 시간 행동, 자원이용), 이식성(품질항목: 적응성)으로 정의하였으며, 개발된 자동화 도구(Componenet Test Manager)는 EJB 빈을 시험하기 위해 시험 데이터를 작성하고 시험 결과를 비교하는 과정을 자동화한 도구이다.

3.2 자동화 시험도구

현재 상용화되어 출시되고 있는 컴포넌트 자동화 시험도구를 알아본다. 자동화 시험도구를 특징별로 분류하여 소개하면 다음과 같다.

○ 테스트 케이스 및 드라이버 자동 생성 도구

- Interactive Development Environments사 TestMaster: OOA/D의 저장소로부터 시험사례들을 생성하는 명세 기반 시험 사례 생성 도구
- Quality Checked Software사 Cantata: 시험 드라이버 메시지 코드를 자동적으로 생성하는 시험 드라이버 생성 도구
- Rational사 RQA: 테스트 스텝과 테스트 드라이버를 자동으로 생성하는 자동화도구
- Parasoft사 Jtest: 테스트 케이스와 테스트 스텝을 자동으로 생성하여 Java 클래스, JSP, 컴포넌트를 시험

○ EJB 시험도구

- Empirix사 Bean-Test: EJB Middle-tier 애플리케이션의 시험을 위한 도구로써 기능성 시험과 함께 성능 및 부하 시험의 기능을 제공
- Segue Software사 SilkPilot : Capture/Playback 방식의 도구를 기본 모델로 하는 상호작용이 용이한 시험도구이며 Middle-tier 서버

컴포넌트의 시험을 위한 도구로 CORBA, Java RMI, EJB 컴포넌트 모델 등을 지원

- Mercury Interactive사 LoadRunner: EJB 성능 시험을 위한 시험도구

○ 동적분석 도구

- Nu-Mega Technology사 BoundChecker: C++소스에서 초기화되지 않은 메모리를 자동적으로 찾아주며 메모리 부족현상을 검증하는 도구

대부분의 상용화된 자동화 도구들은 소스코드를 바탕으로 시험을 수행하기 때문에 제 3자에 의한 컴포넌트 시험에 적용할 수가 없으며, 컴포넌트의 고유 특성(예: 인터페이스의 성능 및 정확성 등)을 지원하는 시험도구가 거의 없다.

4. 범용적인 S/W 품질 모델

S/W품질을 측정하고 표현하는 모델을 S/W 품질 모델이라고 하는데 대표적인 품질 모델에는 ISO/IEC 9126, McCall의 품질모델, Boehm의 품질모델 및 Dromey의 품질 모델 등이 있다. ISO/IEC 9126은 범용 S/W의 품질 측정에 가장 많이 사용되는 품질 모델로써 S/W 컴포넌트 품질 측정을 위한 기본 모델로 사용되었다. ISO/IEC 9126은 다음 그림과 같이 소프트웨어의 품질 속성들을 6가지의 특징과 24개의 하위특징으로 정의하고, 각 하위특징들은 외부와 내부 메트릭에 의해서 측정된다.

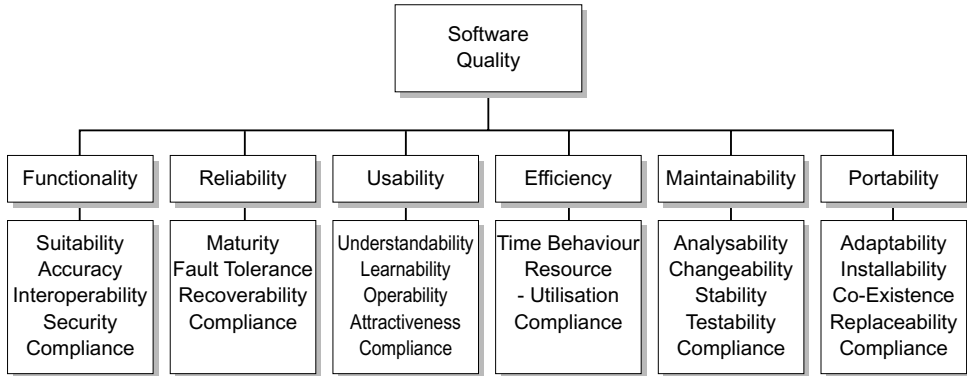


그림 1. ISO/IEC 9126 품질 모델

하지만 이 품질 모델은 재사용이 가능한 컴포넌트 보다는 일반적인 소프트웨어 애플리케이션을 평가하기 위해 제안되었기 때문에 S/W컴포넌트의 품질 측정을 위해서는 품질 모델의 특화가 필요하다.

5. S/W 컴포넌트 품질 평가 모델

5.1 S/W 컴포넌트의 특징

컴포넌트 S/W에 특화된 품질 모델을 정의하기 위해 고려되어야 할 기본적인 컴포넌트의 특징에 대하여 알아본다. 기본적으로 5가지의 특징으로 구분할 수 있으며, 이러한 특징들을 기반으로 해서 S/W 컴포넌트의 품질 요소 및 품질 항목들을 정의한다.

○ 도메인 공통 기능 평가

컴포넌트는 컴포넌트를 사용할 조직 내에서의 컴포넌트 재사용성을 향상시키기 위해 이를 구성하는 패밀리 멤버들 간의 동일한 도메인 내에서의 공통된 기능을 제공해야 한다. 해당 컴포넌트에 대하여 동일한 도메인 내에서 지켜져야 하는 컴포넌트 표준 도메인

모델이 존재한다면 컴포넌트는 해당 표준을 준수해야만 한다.

○ 특화를 위한 Require 인터페이스 평가

컴포넌트 S/W에 있어서 공통성(Commonality) 및 가변성(Variability) 분석은 모델로부터 컴포넌트를 추출하는 주요한 액티비티이다. 패밀리 멤버들 간의 공통된 기능성에 대하여 로직, 어트리뷰트, 워크플로우에 있어서의 가변성이 존재할 수 있다. 컴포넌트는 이러한 가변성을 구현해야 하며 Require 인터페이스를 통하여 로직을 특화할 수 있는 장치를 제공해야 한다.

○ 컴포넌트 인터페이스 및 Granularity 평가

컴포넌트는 전형적으로 복수개의 객체와 클래스로 구성된다. 따라서, 컴포넌트는 OOP에서의 객체 보다는 좀 더 큰 규모의 단위가 된다. CCM에 있어서의 Facet 인터페이스 뿐 아니라 Equivalent 인터페이스와 같은 컴포넌트 수준에서의 인터페이스를 제공함으로써 컴포넌트 구매자들은 더 적은 수의 API 만을 알아도 컴포넌트의 기능들을 이용할 수 있게 된다. 이를 통하여 컴포넌트를 통합하는 과정에서의 컴포넌트와

애플리케이션 간의 결합도를 낮출 수 있다. 이러한 이유로 블랙박스 컴포넌트가 더 선호되며, 컴포넌트의 인터페이스는 컴포넌트 단위의 재사용에 있어서의 Granularity와 중요한 요소이다.

○ 이진 파일 수준의 조합 및 대체성 평가

CBD에서 컴포넌트 간의 조합 및 컴포넌트와 목적 애플리케이션 간의 통합은 이진 파일 또는 바이트 코드 수준에서 이루어져야 한다. 또한 컴포넌트간 또는 컴포넌트와 목적 애플리케이션 간의 조합은 전형적으로 해당 컴포넌트 참조 모델을 통해서 정의되는데, 통합되는 컴포넌트들은 애플리케이션의 다른 부분들을 수정하지 않고도 새로운 버전의 컴포넌트들과 대체가 가능해야 한다.

○ 컴포넌트 참조 모델 준수성 평가

컴포넌트 참조 모델을 통해서 컴포넌트간 또는 컴포넌트와 목적 애플리케이션 간의 통합은 컴포넌트 플랫폼과 같은 컴포넌트 구현 환경을 통하여 구현된다. 따라서 컴포넌트 구매자들은 목적 컴포넌트의 컴포넌트 참조 모델을 고려해야 하며, 참조 모델을 어느 정도 준수하는지도 고려해야 한다.

5.2 S/W 컴포넌트 품질 평가 모델

ISO/IEC에서 제시한 범용적 품질모델을 기반으로 하여 S/W 컴포넌트에 적용할 수 있는 특화된 품질모델을 정의하기 위하여, 앞에서 분석된 S/W 컴포넌트의 특성을 반영하여 품질요소와 그 품질요소를 구성하는 품질항목을 다음과 같이 제시한다.

○ 품질요소

- 기능성(Functionality): 일반 S/W 제품의 품질에 있어서의 기능성은 요구사항에 명시되어 있는 기능들이 충분히 구현이 되었는지를 측정한다. 그러나 소프트웨어 컴포넌트 개발에 있어서는 컴포넌트의 특성상 여러 가지 요구사항의 공통된 집합인 패밀리 요구사항에 명시되어 있거나 또는 암시된 기능들을 만족하는 속성이다.
- 재사용성(Reusability): 재사용 단위의 컴포넌트가 동일한 도메인을 대상으로 하는 각 패밀리 멤버들 간에 범용적으로 이용될 수 있는지에 대한 정도를 측정하는 품질요소로 세부 품질 항목으로써 특화성, 결합성, 범용성을 제시한다.
- 준수성(Conformance): 준수성은 소프트웨어 컴포넌트를 개발함에 있어서 지켜져야 하는 일종의 규약이나 표준을 준수했는지를 측정하기 위한 품질요소로 도메인 표준 준수성과 플랫폼 준수성으로써 판단된다.

○ 품질 항목

- 적합성(Suitability): FRS(Family Requirement Specification)에서 요구된 기능들이 구현된 정도
- 결합성(Coupling): 컴포넌트 각각이 다른 컴포넌트와의 연동 시에 컴포넌트 간에 발생하는 상호작용의 정도
- 범용성(Generality): 컴포넌트가 가지는 기능성들이 FRS의 기능성 및 표준 도메인 모델에 대하여 잘 맞는지의 정도
- 특화성(Customizability): 컴포넌트가 가지는 기능을 수행하는데 필요한 로직, 어트리뷰트, 워크플로우에 대한 가변성이 얼마나 구현 되었는지의 정도
- 도메인 표준 준수성(Standard Conformance):

컴포넌트가 주어진 도메인에 존재하는 표준들을 얼마나 준수하는지의 정도

- 플랫폼 준수성(Platform Conformance): 컴포넌트가 미리 정해진 플랫폼을 얼마나 준수하는지의 정도

이러한 품질요소와 품질항목으로 구성된 S/W컴포넌트를 위한 품질모델과 범용적 품질모델과의 관계를 정의하면 다음과 같다. 또한, 각 품질 항목의 측정을 위해 메트릭이 정의되었다.

6. 결론

구매자들은 S/W 컴포넌트를 이용하여 개발함으로써 전체 소프트웨어에 대한 비용과 시간을 단축할 수 있기를 바란다. 구매자들은 제 3의 제조자들에 의해 개발된 컴포넌트에 대하여 'Not-Invented-Here' 신드롬을 가지고 있기 때문에 구매하기 이전에 컴포넌트에 대한 철저한 평가를 원한다. 이러한 업계의 요구에 대응하기 위해 TTA에서는 실용적이며 구체적인 S/W 컴포넌트 평가 품질모델을 개발하였으며, 개발된 품질모델은 S/W 컴포넌트 기술의 변화에 대응하여 계속 개선되고 있다.

이와 더불어 주어진 시간과 비용 내에서 보다 정확

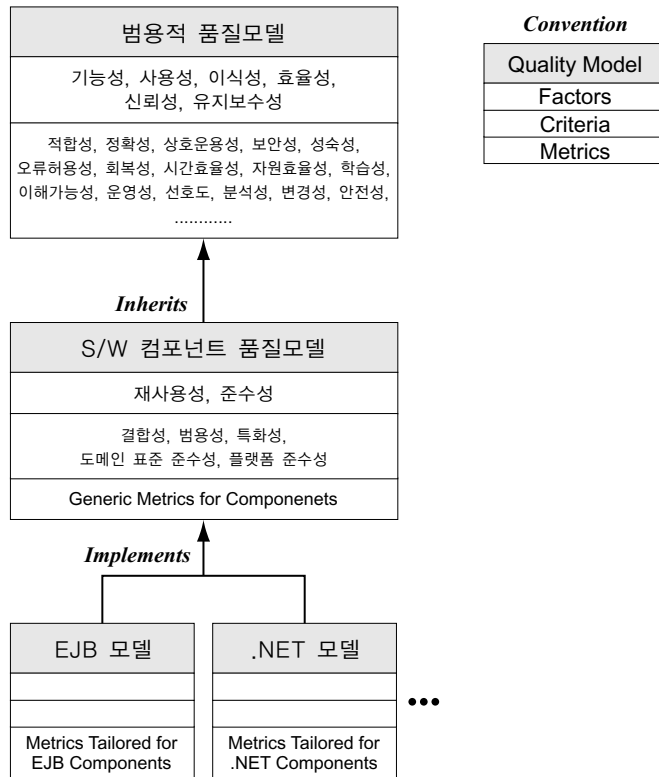



그림 2. S/W 컴포넌트에 특화된 품질모델

하고 신속하게 시험을 수행하기 위해서는 자동화된 컴포넌트 시험 환경과 이를 뒷받침할 수 있는 도구의 개발이 필요하다. 또한 안정된 S/W 컴포넌트의 배출 및 컴포넌트 시장의 활성화를 위하여 S/W 컴포넌트 품질 모델의 표준화 및 시험·인증의 의무화 등을 포함한 제도적 장치 마련이 필요하다.

7. 참고문헌

- [1] ISO/IEC, FCD 9126 Information Technology – Software product quality.
- [2] J.Morris, G.Lee, K.Parker, and G.A.Bundell, “Software Component Certification,” IEEE Computer, v.34 No.9,

September 2001, pp. 30–36.

- [3] L.Gallagher, “Conformance Testing of Object-Oriented Components Specified by State/Transition Classes,” NIST technical report, April 1999.
- [4] M.schroeder, “A Practical Guide to Object-Oriented Metrics,” IEEE IT Professional, November/December 1999, pp. 30–36.
- [5] 컴포넌트, Mobile S/W 테스트/인증 방안에 관한 연구, 한국정보통신기술협회, 2002.
- [6] 컴포넌트 시험 평가 가이드라인 개발에 관한 연구, 한국소프트웨어컴포넌트컨소시엄, 2001.
- [7] 컴포넌트 시험 환경 설계 및 구축에 관한 연구, 한국전자통신연구원, 2000. 

DMB 미들웨어 표준화 착수

지상파 디지털 멀티미디어 방송(DMB)의 표준화 작업을 주도하고 있는 차세대방송표준포럼(위원장 김용한 서울시립대 교수)이 향후 DMB가 이동통신과 결합할 것을 전제로 DMB의 미들웨어 표준화 작업에 나설 예정이다. 특히 차방포럼은 DMB의 양방향 데이터서비스용 미들웨어로 유럽형 데이터방송 플랫폼인 MHP(Multimedia Home Platform)과 한국형 무선인터넷 플랫폼인 WIPI(Wireless Internet Platform for Interoperability)를 복수로 고려하고 있어 관련 업계의 비상한 관심을 끌고 있다. 이는 휴대폰과 결합함으로써 양방향의 방송·통신 융합서비스가 될 DMB의 플랫폼을 결정하는 것이어서 표준화 방향에 따라 방송 이동통신 미들웨어 수신 단말기 및 방송 통신의 애플리케이션 업계에 상당한 영향을 끼칠 것으로 보인다. 차방포럼 관계자는 “최근 지상파 DMB 표준의 하나로 MPEG4 BIFS(Binary Format for Scene)를 채택했으나, 이 규격으로는 향후 이동통신과 DMB가 결합된 단말기에서 완벽한 양방향 데이터 서비스를 기대하기가 힘들다”며 “이에 따라 MHP와 WIPI 등을 DMB의 미들웨어로 채택하는 방안을 앞으로의 표준화 작업에서 적극 검토할 예정”이라고 말했다. 차방포럼은 최근 지상파 DMB의 비디오 및 오디오 압축규격 등을 확정했으며, 한국정보통신기술협회(TTA)의 단체표준화 절차를 마치는 대로 양방향의 데이터서비스용 미들웨어 표준화 작업에 착수하기로 했다.