

유비쿼터스 환경의 서비스 융·복합 아키텍처[†]

충남대학교 이규철*
한국정보사회진흥원 이호경

1. 서 론

정보통신부에서 기존의 IT 839를 변경하여 2006년 부터 추진되는 u-IT839 전략[1]에서는 각 서비스와 인프라들이 독자적으로 발전하는 단계에서 서비스 간 상호운용성 제공과 연동을 통한 융복합 서비스 창출의 단계로 발전하고 있다.

특히 유비쿼터스 환경에서의 다양한 서비스 및 단말에 대한 연동 및 융합을 위한 요구사항이 증가함에 따라, 전 세계적으로 이에 따른 관련 표준 및 기술 개발이 시작되고 있는 상황이다.

또한 전자정부 등의 공공분야는 차기 목표로서 이음새 없는 서비스제공을 추진 중에 있으며, 이를 위하여 다양한 정부 시스템들을 연계, 통합하기 위한 서비스 융복합 인프라에 대한 수요가 매우 높다고 할 수 있다.

민간분야에서도 대기업을 중심으로 신규 비즈니스 창출과 실시간기업(RTE)[2]으로의 발전을 위하여, ERP, CRM 등 자사나 파트너사 들의 다양한 정보시스템들을 연계 통합하는 사업이 증가하고 있다.

서비스 융복합을 달성하는 기술들을 살펴보면, 2000년 이후로 국제표준을 따르는 웹서비스[3] 기술이 급부상하고 있다. 현재 웹서비스 기술은 기업 내의 정보시스템 통합을 위한 서비스 지향 아키텍처(SOA: Service Oriented Architecture)[4] 실현 기술로서 발전하고 있다. 그러나, 현재의 SOA는 향후에는 유비쿼터스 서비스 및 디바이스 통합, RFID[5] 등의 외부 단말의 통합의 방향으로 발전하여 궁극적으로 이음새 없는 서비스 컨버전스가 실현될 전망이다.

따라서 이음새 없는 서비스 컨버전스가 실현되기 위해서는 현재의 웹서비스 기반 SOA 기술 외에 유비쿼터스 환경에서의 다양한 서비스와 디바이스 통합 및

RFID 등의 외부 단말을 융복합하는 기술개발이 요구된다.

이 중 웹서비스와 유비쿼터스 서비스와 디바이스 연동 기술로 새롭게 각광 받고 있는 유비쿼터스 웹서비스(UWS: Ubiquitous Web Service) 기술은 현재 세계적으로도 초기단계에 있는 실정이다[6].

또한 유비쿼터스 서비스 컨버전스의 실현을 위해서는 RFID 등을 이용하여 외부 환경의 변화를 감지하고 이를 적시에 대응하는 실시간 서비스구축을 위한 핵심 기술로서 요사이급부상하고 있는 EDA(Event Driven Architecture) 기술도 같이 결합되어야 한다.

본 고에서는 이와 같은 유비쿼터스 환경의 서비스 융복합에 필요한 기반 기술인 SOA, UWS, EDA를 살펴보고, 이를 가능하게 하는 아키텍처를 정의한다.

2. 서비스 융복합 기반 기술 개요

2.1 SOA

SOA(Service Oriented Architecture)[4]는 비동기 방식의 메시지 교환을 통해 약하게 결합된 분산 소프트웨어 아키텍처이다. SOA는 웹서비스를 비롯하여 ebXML, RosettaNet 등과 같은 e-비즈니스 프레임워크를 포괄하는 아키텍처이다.

분산 시스템은 의도하는 기능을 실현하기 위해서 협력하는 소프트웨어 에이전트들로 구성되게 되는데, 이러한 에이전트가 서비스를 단위로 형성된 특별한 유형의 분산 아키텍처를 SOA라고 하는 것이다. 실제로 서비스는 컴포넌트들이 모여서 제공할 수도 있고, 큰 응용 프로그램의 일부 기능을 노출하여 구현될 수도 있는데, 이러한 서비스는 일반적으로 네트워크상에서 참조될 수 있는 인터페이스를 가지고 있으며, 표준 프로토콜과 데이터 형식을 사용한다. 서비스를 이용하는 에이전트는 오직 서비스의 인터페이스에만 주의를 기울이면 된다.

그림 1은 W3C에서 제시하고 있는 SOA에 대한 일

[†] 본 고의 내용은 한국정보사회진흥원에서 수행한 "유비쿼터스 환경의 서비스 융복합 로드맵 연구"의 일부임.

* 종신회원

반적인 체계를 보여주는데[3], 여기서 SOA의 핵심 요소로는 교환되는 메시지, 서비스 요청자와 서비스 제공자로 행동하는 에이전트, 메시지의 흐름을 허용하는 공유된 전송 메커니즘, 컴퓨터에 의해 처리될 수 있는 서비스나 메시지에 관한 서술 등이 있다. 이런 정의에 의하면, COM이나 CORBA와 같은 분산 객체 기술이나 웹 아키텍처도 SOA의 일종이라고 할 수 있다. 그러나 진정한 의미의 SOA는 웹서비스가 크게 활성화되면서 부각되기 시작했는데, 이것은 MS, Sun, IBM 등 대형 소프트웨어 개발 회사가 각기 색채가 다른 솔루션을 제공하면서, 통합적인 관점을 제공하기 위한 용도로 많이 사용되었다. 물론 W3C나 OASIS와 같은 표준화 기관에서도 이러한 통합적 개념이 필요하게 되어 활발히 사용하고 있다.

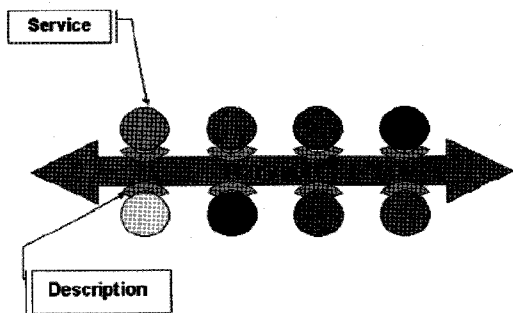


그림 1 W3C의 SOA 정의

SOA는 서비스를 중심으로 프로그램 코드를 공유하고 재활용하기 위한 것이다. SOA는 크게 3가지의 역할을 하는 구성요소와 이들 간의 관계로 규정될 수 있는데(그림 2), 역할로는 서비스 제공자 (Service Provider), 서비스 소비자 (Service Consumer), 서비스 등록기 (Service Registry) 가 있다.

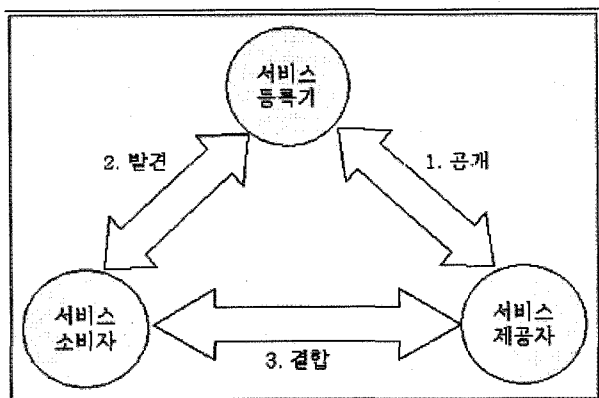


그림 2 SOA의 구조

먼저, 서비스 제공자는 자신의 어떤 응용의 일부분 또는 전부를 어떻게 네트워크상의 서비스로서 사용할 수 있는 것인지를 정의한다. 이러한 내용을 서비스 등록기

에 공개하여 서비스 소비자가 활용할 수 있도록 제공한다. 서비스 소비자인 응용은 자신이 필요한 서비스를 발견하고, 그 내용을 서비스 등록기로부터 추출하여 이해하고, 이것을 바탕으로 서비스 제공자의 서비스를 결합하여 사용할 수 있게 된다. OASIS에서는 기능을 중심으로 SOA에서 서비스 기술(Service Description), 서비스 발견(Service Discovery)과 함께 메시징(Messaging)과 조정(Orchestration/Coordination)을 중요한 기능으로 간주하고 있다[9]. 메시징은 결합 과정에서 사용되는 것으로 보안 문제를 포함하고 있으며, 조정은 서비스들을 결합하여 비즈니스 목표를 달성하기 위한 것으로 관리 문제와 연관된다. 이러한 기능들은 공통의 전송 표준과 공통의 언어 표준을 기반으로 하고 있다.

2.2 UWS

초기의 웹서비스 기술은 주로 비즈니스 분야에서 다양한 응용에 대한 통합의 도구로서 이용되어 왔으나, 현재는 광대역통합망(BcN: Broadband convergence Network) 기반의 유무선 통합 응용, 방송/통신 융합, 정보 가전/홈네트워킹, 임베디드 환경 등 IT839 전략의 다양한 분야에서 핵심 연동 기술로 그 활용 범위가 빠르게 확산 되고 있다.

이처럼 웹서비스는 단순한 비즈니스 응용 분야의 확장뿐만 아니라 단말의 다양성, 네트워크의 다양성, 사용 환경의 다양성과 서비스 환경의 변화/융합을 포괄하는 유비쿼터스 환경을 고려한 유비쿼터스 웹서비스 기술로의 발전이 필수적이다.

유비쿼터스 웹서비스 (Ubiquitous Web Services)는 어떠한 단말/네트워크 환경에서도 다양한 응용 서비스를 연계하여 이용할 수 있도록 하는 미래형 웹서비스 기술로 정의될 수 있다[6].

그동안 유비쿼터스 컴퓨팅 소프트웨어 요소기술로서 기존의 MS의 DCOM(Distributed Component Object Model)이나 CORBA(Common Object Request Broker Architecture) ORB(Object Request Broker), SUN의 Java RMI(Remote Method Invocation) 등의 분산객체 기술들을 활용하는 연구가 추진되어 왔으나, 기존의 분산객체 기술들은 많은 이기종 시스템, 다중 클라이언트, 다중 네트워크 환경, 저전력/저성능 컴퓨팅 환경 등을 지원하기 어려운 한계를 나타내었다.

유비쿼터스 웹서비스 기술은 이러한 기존의 분산객체 미들웨어 기술들의 한계를 극복하며, 플랫폼 독립적인 애플리케이션 통합을 지원함으로써, 타 웹 기술들과 함께 실질적인 유비쿼터스 컴퓨팅의 소프트웨어 통합 기반구조로서의 역할을 수행할 수 있다.

유비쿼터스 웹서비스 기술 중 중요한 하나의 영역인 모바일 웹서비스에 대해서는 무선인터넷 사업자, CP, 서비스제공자 등의 73%가 모바일 단말에서 이용할 수 있는 웹서비스 기능을 요구하고 있으며, 이에 대한 표준 부재가 가장 큰 문제점으로 대두되고 있다.

향후 유비쿼터스 웹서비스는 단순한 유선 기반의 통상적인 컴퓨팅 환경에서 벗어나, 다양한 장치와 단말을 지원하는 기술, 다양한 컴퓨팅 환경(홈네트워크, 텔레매틱스, 광대역통합망, 모바일, 방송 등)을 지원하는 기술 등으로 분야별로 발전하면서 연계/융합될 것으로 예상되고 있다.

2.3 EDA

SOA와 더불어 최근에는 EDA에 대한 관심 또한 커지고 있어 SOA와 EDA를 합하여 SOA 2.0이라고도 명명하고 있다. 정보 조사 기관에 의하면, 현재는 대기업과 중소기업의 10% 미만이 SOA와 EDA 기반으로 응용을 구축하고 있고, 패키지 응용에서도 5% 미만에서 적용되고 있지만, 2009년엔 새로운 패키지 응용과 비즈니스 응용의 25% 이상이 SOA와 EDA 기반으로 구축될 것으로 전망하고 있다[10].

SOA와 EDA는 상호 보완적인 서비스 구축 방법론으로 SOA의 기본 개념이 정의된 서비스 인터페이스를 이용하여 요청(request) 및 응답(response)을 통해 서비스를 연동하는 것이라면, EDA는 그림 3과 같이 이벤트에 대한 감지(sense) 및 대응(respond) 모델이다. 표 1의 비교처럼 SOA는 클라이언트에 의해 서비스가 제어되며 순차적으로 실행되는 반면, EDA는 이벤트 수신자가 대응 여부를 결정하며, 이벤트가 동시에 여러 곳으로 전달이 가능하고 또한 비동기 방식으로 전달이 가능하므로 이벤트 발생에 의한 대응이 동적으로 구성된다.

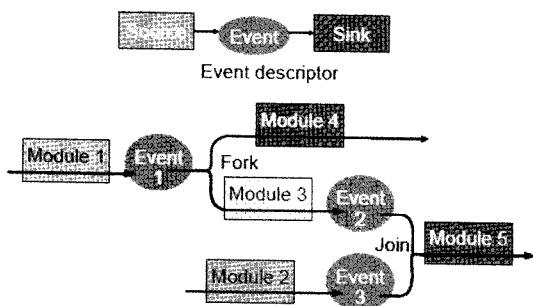


그림 3 EDA 개념

양 서비스 구축 방법론의 등장 배경이 구축하고자 하는 서비스 자체의 속성에 따른 차이를 반영한 것뿐만 아니라, SOA가 서비스의 재활용 및 공유가 주목적

이라면, EDA는 실시간 서비스 제공이 주목적이라고 할 수 있다.

표 1 SOA와 EDA 비교

구분	SOA	EDA
상호규약	서비스 인터페이스 정보	이벤트 규격 정보
연결방식	1 : 1	N : N
흐름제어 주체	클라이언트	이벤트 수신자
흐름제어 방식	순차경로	동적/병렬/비동기 방식
새로운 입력에 대한 대응	진행 중에는 차단	진행 중에도 반응

EDA를 지원하는 다양한 기술들이 실제 응용들에서 이미 사용되고 있으며 이벤트 처리 방법에 따라 크게 다음과 같이 나눌 수 있다[11].

- 외부 단순 이벤트 처리: RFID와 같은 외부 이벤트 소스로부터 발생한 이벤트들은 모두 의미있는 이벤트로 간주하고 각각의 이벤트 내용에 따라 대응되는 액션을 수행한다. 또는 의미있는 이벤트와 무의미한 이벤트가 같이 발생하는 대량의 이벤트 스트림을 대상으로 하며, 필터링 등을 수행하여 의미있는 이벤트 정보만 뽑아서 응용에 전달 혹은 서비스와 연동한다. RFID, USN 미들웨어 등이 이에 해당된다.
- 내부 복합 이벤트 처리 : 여러 이벤트 소스로부터 발생한 이벤트를 대상으로 이벤트들의 영향을 분석하여 대응되는 액션을 수행한다. 단순 이벤트 처리가 하나의 이벤트를 대상으로 한 반면, 복합 이벤트 처리는 여러 이벤트간의 다양한 관계를 분석한다. 복합 이벤트 처리는 BAM(Business Activity Monitoring)에 내장되어 제공되거나 별도의 복합 이벤트 처리 시스템으로 제공되기도 한다.

본 고에서는 외부에서 발생하고 있는 다양한 정보, 즉 위치 정보, 환경 정보 등과 같이 실세계에서 발생하는 다양한 이벤트 정보를 RFID, USN 미들웨어를 통해 수집하고 이를 종합적으로 분석하여 서비스와 연동하는 외부 단순 이벤트 처리 부분만을 EDA의 범위로 정하였다.

한편 기업이 비즈니스 환경 변화에 유연하고 민첩한 대응을 위해서 기업내/외부적인 프로세스의 지연 요소를 제거하고, 비즈니스 과정에서 발생하는 이벤트와 비즈니스에 영향을 줄 수 있는 이벤트의 조기 감지와 신속한 대응을 할 수 있는 실시간 기업(RTE) 구현을 위해 필요한 내부 복합 이벤트 처리 부분은 SOA의 범

주에 포함시켜 구분하였다.

3. 서비스 융복합 아키텍처

본 장에서는 유비쿼터스 환경의 서비스 융복합에 필요한 기술인, SOA, UWS, EDA를 모두 지원할 수 있는 통합 아키텍처를 정의한다. 이를 위해 우선 전체 아키텍처의 구성요소를 개괄적으로 정의하고, SOA, UWS, EDA 각 부분의 아키텍처를 상세히 기술한다.

3.1 서비스 융복합 아키텍처의 구성요소

서비스 융복합 아키텍처의 핵심 구성요소는 그림 4에서 보듯이 SOA를 지원하는 SOA Federation, EDA를 지원하는 Event-Driven Service Broker, 유비쿼터스 웹 서비스 네트워크(Ubiquitous Web Service Network)과 연결하는 UWS Gateway이다.

사용자들은 개인 컴퓨터나 무선 디바이스를 이용하여 포털(Portal)을 통해 SOA Federation에 접근할 수 있으며, 협업시스템(Federated System)은 표준 서비스 게이트웨이(Standard Service Gateway)를 통해 접근할 수 있다. 이 게이트웨이는 SOA Federation과 협업시스템 간의 안전한 통신과 외부 협업을 지원한다. 표준 서비스 게이트웨이는 외부의 프로토콜이나 협업을 지원하기 위해 Administrative와 Collaborative 두 종류의 메시지 타입을 제공한다. Administrative 메시지 타입은 SOA Federation 서버 관리에 사용되며, Collaborative 메시지 타입은 SOA Federation 협업 기능에서 사용된다. SOA Federation 서버 관리

에 해당하는 것은 SOA 참여자의 프로파일 정보의 수정 및 저장, 모든 협업 엔티티 타입의 생성 및 정의, 협업 문서 정의 등이 있다. SOA Federation 협업 기능에는 협업 행위 결과물을 Federation 서버에 등록하고 보안 점검 등이 해당된다.

RFID와 같은 외부 이벤트 소스들이 발생시키는 이벤트들은 Event-Driven Service Broker에 의해 이벤트의 의미가 분석되어, 처리되어야 할 서비스와 연동되게 된다. 이와 같은 연동은 이벤트와 관련된 서비스 시나리오를 관리하고 해석함으로써 가능하게 된다.

전등 스위치나 시계, 문손잡이 등 다양한 디바이스들이 연결된 유비쿼터스 웹 서비스 네트워크는 이들 디바이스와 SOA의 서비스를 연계하기 위해 UWS Gateway를 사용하게 되며, 이들 디바이스가 발생시키는 이벤트는 Event-Driven Service Broker로 직접 연결될 수 있다.

SOA Federation은 서비스 융복합 아키텍처의 중심에 위치하고 있으며, FERA(Federated Enterprise Reference Architecture)[12]의 표준을 따라 Federation 서버, 에이전트 프레임워크(Agent Framework), 협업 프로세스 플로우 컨트롤러(Collaborative Process Flow Controller)와 Built-in 서비스로 구성된다. Federation 서버는 SOA의 내·외부 환경을 연결시켜주며, 에이전트 프레임워크는 협업 과정이 일어나는 동안 행위를 수행하고 판단하는 에이전트 집합소이다. 협업 프로세스 플로우 컨트롤러는 협업 프로세스 내의 협업 프로세스 플로우를 관리한다. 각각의 협업 프로세스는 하나 이상의 협업 프로세스 플로우를 가질 수 있는데, 협업 프로세스 플로우에는 행위,

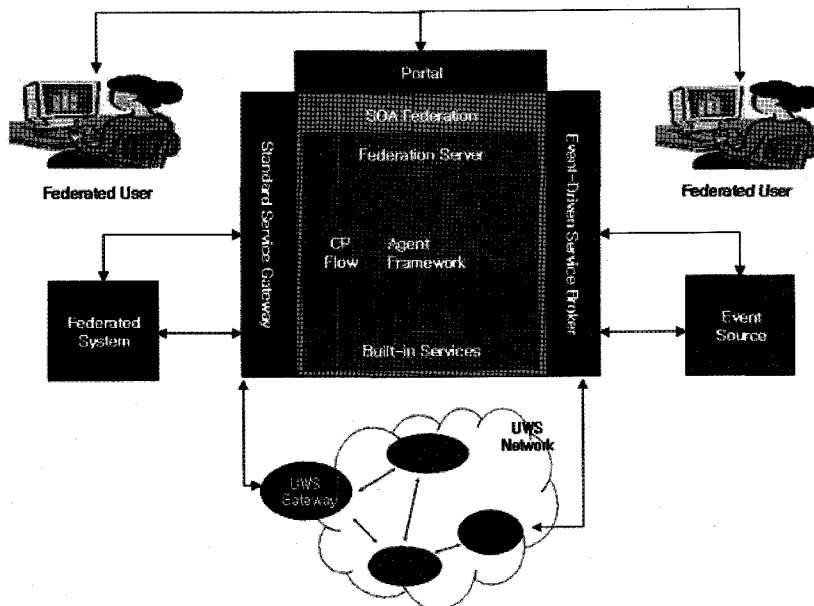


그림 4 서비스 융복합 전체 아키텍처

결정, 입출력 참조와 프로세스 내의 특정 값을 평가하는 규칙에 대한 유효성과 입·출력의 확인과 같은 프로세스를 수행하는 동안 발생하는 이벤트가 포함된다.

3.2 SOA Federation

SOA Federation의 자세한 구조는 그림 5와 같다. SOA Federation은 오케스트레이션(Orchestration) 서버 역할을 담당한다. 협업 프로세스와 협업 프로세스 수행 전체에서 관여되는 웹 서비스를 관리하는 SOA Federation은 다음과 같은 컴포넌트로 구성된다.

- Federation 서버(Federation Server)
- 에이전트 프레임워크(Agent Framework)
- 협업 프로세스 플로우 컨트롤러(Collaborative Process (CP) Flow Controller)
- 이벤트 관리자(Event Manager)
- 비즈니스 행위 관리자(BAM: Business Activity Manager)
- 비즈니스 규칙 엔진(Business Rule Engine)
- Built-in 서비스(Built-in Services)

3.2.1 Federation 서버

협업 내용과 문맥은 Federation 서버에 저장되고 정의된다. Federation 서버는 SOA Federation 서버와 외부 연결시켜 주며, 역할은 다음과 같다.

- 협업 프로세스를 수행하는 동안의 모든 협업 행위에 대한 조정
- 협업 참여자 프로파일, Gateway 프로파일, 보안 프로파일, 비즈니스 프로세스 표준, 협업 문서, 웹 서비스 정보의 저장 및 버전 관리
- 협업 프로세스의 흐름을 정의한 메타 스키마 저장

및 버전 관리

- Gateway와 Portal과의 통신

이러한 역할을 수행하는 Federation 서버의 구성 요소는 다음과 같다.

3.2.1.1 Federation 관리자

Federation 관리자는 SOA Federation과 협업 시스템 간의 중앙에 위치하는 조정자로서 다음과 같은 역할을 수행한다.

- 협업시스템과 관계된 요청 및 응답을 관리
- 보안 인증과 권한에서 필요한 모든 것을 보안 제공자(Security Provider)를 사용하여 수행
- Federation 레지스트리에 메타데이터를 저장하고 질의
- 에이전트 프레임워크 인터페이스를 관리하는 에이전트 인터페이스 관리자가 필요로 하는 정보를 제공하고 그와 통신하는 기능

3.2.1.2 에이전트 인터페이스 관리자

에이전트 인터페이스 관리자는 에이전트 프레임워크 인터페이스를 관리한다. 에이전트 인터페이스 관리자가 제공하는 서비스로는 다음과 같이 3가지가 있다.

- Agent delegation: 에이전트 및 그와 관계된 프로토콜 상세 정보를 포함하는 에이전트 인터페이스를 생성
- Agent monitoring and escalation: SOA Federation 에이전트를 모니터링하고, 에이전트가 사용되지 않을 때, 수정작업 수행
- Agent invocation(binding): 어떤 결정이나 행위 수행 시 필요한 에이전트를 호출

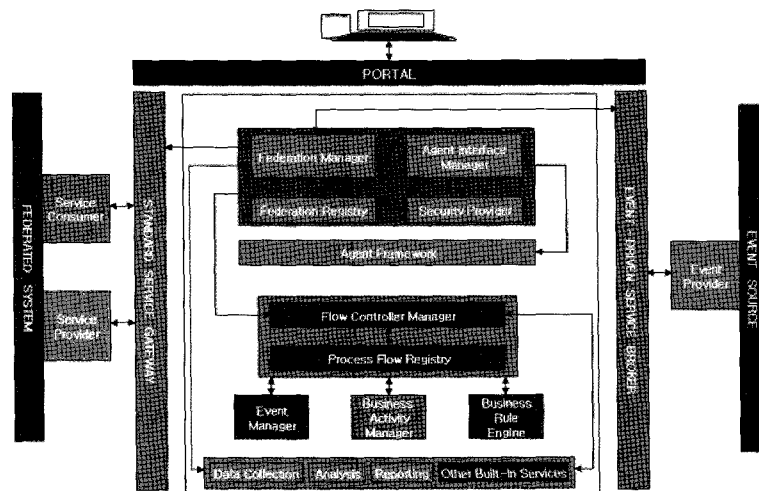


그림 5 SOA Federation의 상세 구조

3.2.1.3 Federation 레지스트리

Federation 레지스트리는 Federation과 관련된 모든 정보를 저장, 버전 관리, 질의 및 관리 기능을 제공한다. Federation과 관련된 모든 정보라는 것은 협업 참여자 프로파일, Gateway 프로파일, 보안 프로파일, 비즈니스 프로세스 표준, 협업 문서, 웹 서비스 정보를 포함한 내용과 협업 메타데이터이다.

Federation 레지스트리로는 UDDI와 ebXML 레지스트리가 모두 사용될 수 있다.

3.2.1.4 보안 제공자

보안 제공자는 CP Flow Security Manager, Federation Identity Manager, Federation Security Policy Manager를 제공한다. CP Flow Security Manager는 다른 2개의 Manager를 조정하고 관리하며, Federation Identity Manager는 Federation 서버에 접근하는 사용자나 시스템을 인증하고, Federation Security Policy Manager 보안 정책에 따라 권한 체크를 수행한다.

3.2.2 에이전트 프레임워크

에이전트 프레임워크는 에이전트를 설계, 구현, 관리하는 서비스들의 집합이다. SOA Federation에서의 웹 서비스는 어떤 결정을 내리고 행위를 수행하는 에이전트로부터 응답을 받고, 에이전트로 요청을 보낼 때 필요한 에이전트 인터페이스를 제공하고 배치하는 기능을 수행한다.

에이전트는 원격의 협력시스템이나 SOA Federation 내에서 동작할 수 있다. 원격의 에이전트는 고정된 위치나 이동형 모두가 될 수 있으며, SOA Federation 에이전트는 그림 5의 에이전트 프레임워크에 의해 지원된다.

에이전트 프레임워크는 에이전트 프레임워크 표준과 비즈니스 룰 표준이 필요하다. 그러나 현재, 에이전트 프레임워크에 대한 표준은 존재하고 있지 않은 실정이다.

3.2.3 협업 프로세스 플로우 컨트롤러

협업 프로세스 플로우 컨트롤러는 Flow Controller Manger와 Process Flow Registry를 포함한다. SOA 시스템에서 수행되는 각각의 협업 프로세스는 협업 프로세스 플로우와 협업 프로세스 모델링을 지원하는 다른 협업 엔티티의 집합이다. Flow Controller Manager는 협업 프로세스 플로우를 관리한다. 협업 프로세스 플로우는 협업 이벤트, 행위와 역할 사이의 협업 방법을 정의한 집합을 뜻한다.

FERA 기반의 SOA에서 행위와 결정은 에이전트나 사람 혹은 사람이 사용하는 시스템에서 SOA Federation으로 보내지는 입력에 의해 수행된다. 에이전트가 수행

하는 행위와 결정은 모두 자동적으로 수행된다.

Flow Controller Manager는 협업 프로세스의 행위와 결정에 대한 선택, 입출력의 가능성과 협업 프로세스 플로우를 관리한다. 예를 들어 입력 값을 취할 수 있게 되자마자, Flow Controller Manager는 모든 행위와 결정, 이벤트에 알려주게 된다. 정보 기반에서 Flow Controller Manager는 비즈니스 행위 관리자나 비즈니스 규칙 엔진, 또는 이벤트 관리자에게 메시지를 보낼 것이다.

Process Flow Registry는 협업 프로세스 플로우 정보 엔티티를 관리한다. 정보 엔티티에 포함되는 것은 CollaborativeProcessFlow, Activity, Decision, Event, InputOutput 등이 있다.

3.2.4 이벤트 관리자

이벤트 관리자는 이벤트를 관리한다. 여기서 이벤트란 협업 프로세스 플로우가 수행되는 동안 발생하는 것들을 표현한 협업 엘리먼트를 뜻한다. 각각의 이벤트는 이벤트 및 이벤트 결과에 따른 하나 이상의 행위를 생성하는 트리거(trigger)를 가진다. 이벤트는 클러스터를 구성하거나 조합된 이벤트를 생성할 수 있다. 이러한 이벤트들은 참여자가 각 상태의 의미를 바꾸는 생명 주기 내에서 발생할 수 있다. 이벤트는 SOA Federation 문맥이나 연합된 시스템 내에 위치할 수 있다.

이벤트 관리자는 트리거 상태가 되거나 트리거에 기반한 이벤트 상태를 조정할 때 이벤트 인스턴스를 생성한다.

3.2.5 비즈니스 행위 관리자

행위는 SOA 협업시스템이나 로컬 SOA Federation 에이전트에 의해 수행되는 오퍼레이션이나 태스크를 의미한다. 이러한 행위 중 Flow Controller Manager와 Process Flow Registry와 통신하는 행위들은 Business Activity Manager가 관리한다.

행위에 필요한 모든 입력 값이 만족되었을 때, Flow Controller Manager는 그 사실을 BAM에게 통보한다. 그러면 BAM은 Process Flow Registry에 저장되어 있는 정보(행위와 관련된)를 바탕으로 행위를 관리한다. 행위가 수행되면 결정에 대한 기준이나 다른 행위의 입력이 되는 출력물이 생성된다.

또한 BAM은 행위가 수량 정보를 생성한다면 그 정보를 수정한다. 수량 정보에 해당하는 것은 비즈니스 규칙에서 참조되어지는 특정 논의에 포함되는 정량의 값을 포함한다. 비즈니스 규칙은 협업 프로세스가 수행되는 동안 협업 플로우를 자동으로 조정하는 기능을 수행한다. 또한, 수량 정보에는 협업 프로세스를 측량하기도 하며, 그들의 패턴을 생성하기도 한다.

3.2.6 비즈니스 규칙 엔진

행위와 유사하게 결정권은 협업시스템이나 로컬 SOA Federation 에이전트에 의해 수행된다. 이러한 결정 중 Flow Controller Manager와 Process Flow Registry와 통신을 위한 결정 관리는 비즈니스 규칙 엔진이 한다.

결정을 위해 필요한 모든 조건들이 만족되었을 때, Flow Controller Manager는 그 사실을 비즈니스 규칙 엔진에게 통보한다. 그 사실을 통보 받은 비즈니스 규칙 엔진은 Process Flow Registry에 저장되어 있는 정보 중 결정과 관련된 정보를 기반으로 결정 수행을 관리한다. 결정이 수행될 때, 선택 사항이 생성되며, 이는 다른 행위의 입력이나 다른 결정의 조건이 된다.

3.2.7 Built-in 서비스

Built-in 서비스는 SOA Federation, SOA 참여자, SOA 관리자가 사용가능한 기능을 제공한다. 이들은 서로 다른 종류의 태스크를 수행하게 되는데, 이들은 다음과 같다.

- 모니터링
- 시스템과 협업 보고
- 영향 분석
- 데이터 분석
- 데이터 품질 분석
- 협업 패턴 분석

3.3 Event-Driven Service Broker

다양한 이벤트 소스로 부터의 이벤트를 해석하여 SOA 와 Federation과 연동시켜 주는 역할을 하는 Event-Driven Service Broker의 상세구조는 그림 6과 같다.

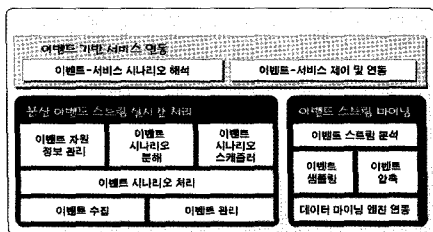


그림 6 Event-Driven Service Broker

Event-Driven Service Broker는 크게 다음과 같은 세 부분으로 구성된다.

- 분산 이벤트 스트림 실시간 처리
- 이벤트 스트림 마이닝
- 이벤트 기반 서비스 연동

분산 이벤트 스트림 실시간 처리 부분은 다양한 외부 소스에서 발생하는 이벤트 스트림을 수집, 필터링,

통합, 변환 등을 통해 발생 상황을 실시간으로 감지하고 이벤트 정보를 제공하는 부분으로, 시스템의 확장성 및 가용성을 위해 분산 처리를 제공하며 다음과 같은 모듈로 구성된다.

- 이벤트 자원 정보 관리: 분산되어 있는 이벤트 처리 노드들에 대한 정보(연동된 이벤트 소스, 제공하는 기능, 워크 로드 등)를 관리하며 이벤트 시나리오 분해 및 스케줄러에 정보 제공
- 이벤트 시나리오 분해: 연속 질의어로 표현된 이벤트 시나리오를 분석, 여러 개의 서브 시나리오로 분해
- 이벤트 시나리오 스케줄러: 분해된 이벤트 서브 시나리오를 수행하기에 가장 적합한 이벤트 처리 노드로 분배 및 이벤트 시나리오의 실시간 처리에 대한 QoS 보장을 위해 처리 순서 등 제어
- 이벤트 시나리오 처리: 이벤트 시나리오에 따라 이벤트 통합, 이벤트 필터링, 이벤트 검색, 이벤트 정렬, 이벤트 변환 등 일련의 이벤트 처리 기능을 제공
- 이벤트 수집: 정해진 프로토콜에 따라 이벤트 소스로부터 이벤트를 수집하는 기능을 제공하며 다양한 이벤트 소스를 수용하기 적합하도록 기본 인프라 제공
- 이벤트 관리: 수집된 이벤트가 이벤트 시나리오에 따라 처리되어 외부로 전송 혹은 삭제되기 전까지 시스템 내 저장 관리

이벤트 스트림 마이닝 부분은 대량의 이벤트 스트림으로부터 내재되어 있는 상황을 추론하여 예측하는 부분으로 다음과 같은 기능을 갖는 모듈로 구성된다.

- 이벤트 스트림 분석: 이벤트 스트림 분석 규칙을 해석하여 이벤트 샘플링 방법, 데이터 마이닝 알고리즘 선정 등 분석 규칙 처리를 위한 전체 제어 및 실행
- 이벤트 샘플링: 이벤트 스트림 분석 규칙을 효과적으로 수행하기 위해 이벤트 스트림으로부터 이벤트 정보를 샘플링하는 기능을 제공
- 이벤트 압축: 대량의 이벤트를 효과적으로 관리하기 위한 이벤트 정보 압축 방법을 제공
- 데이터 마이닝 엔진 연동: 상용 데이터 마이닝 엔진과 연동하여 다양한 마이닝 기법을 제공

이벤트 스트림 기반 서비스 연동 부분은 다양한 이벤트에 의해 서비스 실행이 제어되도록 이벤트 기반 서비스 시나리오를 표현 및 처리하는 부분

- 이벤트-서비스 시나리오 해석: 이벤트-서비스 시나리오 정의어 혹은 표현 방식을 입력으로 받아 이를 해석하여 내부 실행 구조로 변환
- 이벤트-서비스 제어 및 연동: 이벤트-서비스 내부 표현 방식을 기반으로 이벤트 정보 전송, 외부 서비스 호출 등 이벤트 기반 서비스 실행을 제어으로 구성되어 있다.

3.4 UWS Gateway

UWS Gateway는 유비쿼터스 환경에서 단말-단말, 단말-서비스 간 연동을 가능케 하는 웹서비스 기반의 서비스 게이트웨이로서 소규모, 저전력의 단순한 기능을 가진 프로세서를 탑재한 시계, 전등 스위치, 문 손잡이, 심장박동기 등으로 구성된 유비쿼터스 웹 서비스 네트워크의 서비스 요구를 SOA Federation과 연계하는 기능을 갖는다(그림 7). 또한 이들 디바이스가 발생시키는 이벤트는 Event-Driven Service Broker로 직접 연결될 수도 있다.

UWS Gateway가 필요한 이유는 이들 디바이스의 프로세서 능력이 한계가 있기 때문에 정상적인 웹 서비스 메시지를 생성하거나 처리할 수 없기 때문이며, 따라서 표준화된 웹 서비스와 연동을 하기 위해서는 이를 중계 및 변환해 주는 Gateway가 반드시 있어야 한다.

UWS Gateway는 이런 역할을 수행하기 위해 다음과 같은 기능을 갖는 컴포넌트로 구성된다(그림 8).

- u-웹 서비스 브로커 모듈: SOA Federation과 표준 서비스 Gateway를 통해 연계될 수 있는 역

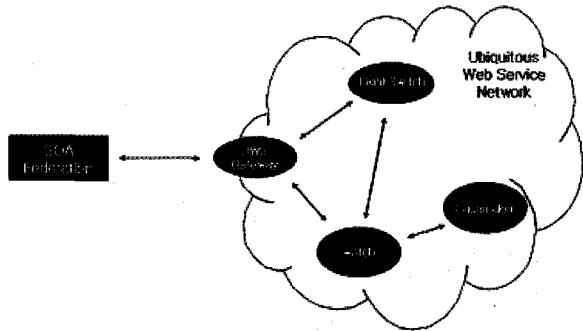


그림 7 유비쿼터스 웹 서비스 네트워크

할을 담당하며 유비쿼터스 디바이스의 서비스 요구를 표준화된 웹 서비스의 형태로 변환하고 응답 메시지를 디바이스가 이해할 수 있는 형태로 변환하는 기능을 담당

- 동적 서비스 처리 모듈: 기존의 UDDI를 유비쿼터스 환경에서 사용할 수 있는 동적인 기능을 갖

는 서비스 레지스트리로 구성하여 활용하며, 새로 나타난 디바이스가 연계될 수 있는 상대 서비스를 적은 비용으로 빨리 찾을 수 있는 동적 서비스 발견 기능을 제공

- u-프로토콜 연동 모듈: 유비쿼터스 디바이스는 여러 가지 제약에 의해 표준화된 SOAP 프로토콜의 사용이 어렵기 때문에 이를 효율적으로 변형한 단순한 프로토콜을 사용하게 되어 이들 다중 프로토콜을 바인딩하는 기능이 필요하며 따라서 여러 가지 프로토콜에 대한 어댑터를 갖추고 있어야 함
- u-웹 서비스 관리 모듈: 유비쿼터스 웹 서비스 관리를 위해 서비스 레지스트리를 관리하며, 새로 나타난 디바이스 및 서비스를 capture하고 디바이스 사이의 서비스 호출 및 브로커를 통한 서비스 호출/응답을 모니터링하는 기능을 제공
- u-웹 서비스 인터페이스: u-웹 서비스를 접근하기 위해 필요한 인터페이스를 제공하며, 미들웨어 연동, 비동기식 웹 서비스의 지원, 메시지 콘텐츠의 변환 등의 기능을 제공

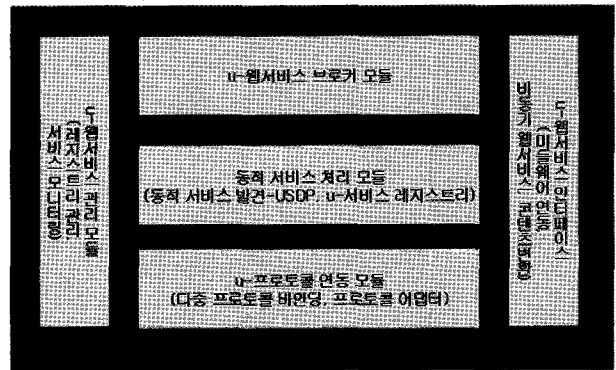


그림 8 UWS Gateway의 구조

4. 결 론

정보통신부에서 2006년부터 추진되는 u-IT839 전략에서는 서비스 간 상호운용성 제공과 연동을 통한 융복합서비스 창출의 단계로 발전을 꾀하고 있다. 특히 유비쿼터스 환경에서의 다양한 서비스 및 단말에 대한 연동 및 융합을 위한 요구사항이 증가함에 따라, 전 세계적으로 이에 따른 관련 표준 및 기술 개발이 시작되고 있는 상황이다.

서비스 융복합을 달성을 위해 이음새 없는 서비스 컨버전스가 실현되기 위해서는 현재의 웹서비스 기반 SOA 기술 외에 유비쿼터스 환경에서의 다양한 서비스와 디바이스 통합 및 RFID 등의 외부 단말을 융복합하는 기술개발이 요구된다. 이 중 웹서비스와 유비쿼터

스 서비스와 디바이스 연동 기술로 새롭게 각광 받고 있는 유비쿼터스 웹서비스(UWS: Ubiquitous Web Service) 기술은 현재 세계적으로도 초기단계에 있는 실정이다. 또한 유비쿼터스 서비스 컨버전스의 실현을 위해서는 RFID 등을 이용하여 외부 환경의 변화를 감지하고 이를 적시에 대응하는 실시간 서비스구축을 위한 핵심기술인 EDA(Event Driven Architecture) 기술도 같이 결합되어야 한다.

본 고에서는 이와 같은 유비쿼터스 환경의 서비스 융복합에 필요한 기반 기술인 SOA, UWS, EDA를 살펴보고, 이를 가능하게 하는 아키텍처를 정의하였다. SOA, UWS, EDA를 지원하는 서비스 융복합 아키텍처의 핵심 구성요소는 SOA를 지원하는 SOA Federation, EDA를 지원하는 Event-Driven Service Broker, 유비쿼터스 웹 서비스 네트워크와의 연결을 위한 UWS Gateway이다.

FERA를 따르는 SOA Federation은 서비스 융복합 아키텍처의 중심에 위치하며, Federation 서버, 에이전트 프레임워크, 협업 프로세스 플로우 컨트롤러와 Built-in 서비스로 구성된다. Federation 서버는 SOA의 내·외부 환경을 연결시켜주며, 에이전트 프레임워크는 협업 과정이 일어나는 동안 행위를 수행하고 판단하는 에이전트 집합소이다. 협업 프로세스 플로우 컨트롤러는 협업 프로세스 내의 협업 프로세스 플로우를 관리한다. 각각의 협업 프로세스는 하나 이상의 협업 프로세스 플로우를 가질 수 있는데, 협업 프로세스 플로우에는 행위, 결정, 입출력 참조와 프로세스 내의 특정 값을 평가하는 규칙에 대한 유효성과 입·출력의 확인과 같은 프로세스를 수행하는 동안 발생하는 이벤트가 포함된다.

Event-Driven Service Broker는 다양한 이벤트 소스로 부터의 이벤트를 해석하여 SOA와 Federation 과 연동시켜 주는 역할을 한다. UWS Gateway는 유비쿼터스 환경에서 단말-단말, 단말-서비스 간 연동을 가능케 하는 웹서비스 기반의 서비스 게이트웨이로서 소규모, 저전력의 단순한 기능을 가진 프로세서를 탑재한 시계, 전등 스위치, 문 손잡이, 심장박동기 등으로 구성된 유비쿼터스 웹 서비스 네트워크의 서비스 요구를 SOA Federation과 연계하는 기능을 갖는다.

본 서비스 융복합 아키텍처는 IT839 전략의 각 분야간 및 다양한 디바이스 간의 서비스 호환성을 제공함으로써 IT839 전반에 걸친 서비스 컨버전스를 실현하는 기반을 만들고, 온라인 도메인 상의 풍부한 콘텐츠와 비즈니스를 IT839 전략과 접목시켜 고부가가치 응용비즈니스를 창출하고 새로운 블루오션 시장을 창

출하는 기대효과를 얻을 수 있을 것이다.

참고문헌

- [1] 양청삼, “u-IT839 전략으로 Dynamic u-KOREA 실현”, 정보통신부, KTOA 2006, 37호, pp.66~72
- [2] 백한진, “RIE 구현을 위한 전략”, SDS Consulting Review 2004
- [3] W3C, “Web Services Architecture”, 2004. 2, <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>
- [4] Adobe, “Service Oriented Architecture”, http://www.adobe.com/enterprise/pdfs/Services_Oriented_Architecture_from_Adobe.pdf
- [5] Nilchael J. Liard, “A White Papter on: Radio Frequency Identification (RFID) Middleware Solutions: Global Market Opportunity, VDC, Aug, 2004.
- [6] Malcolm attard, “Ubiquitous Web Services”, <http://www.cs.um.edu.mt/~csaw/CSAW03/Proceedings/UbiquWebServices.pdf>
- [7] Brenda M. Nilchel son, “Event-Driven Architecture Overview”, Patricia Seybold Group, Feb. 2006.
- [8] W3C, “Web Services at W3C”, <http://www.w3.org/2003/Talks/1211-xml2003-wssoa/slide6-0.html>
- [9] OASIS, “SOA”, <http://www.w3.org/2003/Talks/1211-xml2003-wssoa/slide6-0.html>
- [10] Daryl C. Hunmer, “Software Architecture Will Evolve From SOA and Events to Service Virtulization”, Gartner, March 2005.
- [11] Roy W Schulte, “Event-Driven Applications: Definition and Taxonomy”, Gartner July 2003.
- [12] CPDA, “ebXML and Federated Enterprise Reference Architecture(EFERA)”, 2003.07.

이 규 철



1984 서울대학교 컴퓨터공학(학사)
1986 서울대학교 컴퓨터공학(석사)
1990 서울대학교 컴퓨터공학(박사)
1989. 2~충남대학교 교수
관심분야: 데이터베이스, XML,
e-비즈니스, 시맨틱 웹, SOA
E-mail : kclee@cnu.ac.kr

이 호 경



2002 인하대학교 환경공학(학사)
2004 인하대학교 전자계산공학(석사)
2004. 4~한국정보사회진흥원 ITA·표준팀
연구원
관심분야: 웹서비스 품질관리, 시맨틱 웹,
시맨틱 웹서비스
E-mail : machi@nia.or.kr

• 제19회 영상처리 및 이해에 관한 워크샵 •

- 일 자 : 2007년 2월 7 ~ 9일
- 장 소 : 제주그랜드호텔
- 내 용 : 워크샵 등
- 주 최 : 컴퓨터 비전 및 패턴인식 연구회
- 상세안내 : <http://ipiu.or.kr>