

# 이벤트 기반 서비스 기술 동향

Trend of Event-Driven Service Technology

디지털 홈 특집

이미영 (M.Y. Lee)

데이터베이스연구팀 팀장

김명준 (M.J. Kim)

인터넷서버그룹 그룹장

목 차

- .....
- I . 서론
  - II . 이벤트 기반 아키텍처
  - III . 이벤트 스트림 처리 시스템
  - IV . 결론

이벤트 기반 서비스 기술은 실시간으로 발생하는 이벤트를 감지하고 분석하여 이에 대한 반응으로 서비스가 연동되는 기술로, 실시간 기업 환경 구축이나 유비쿼터스 서비스 환경 구축을 위한 핵심 기반 기술이다. 실시간 기업 환경에서 요구되고 있는 기업내 업무 프로세스에서 발생하는 다양한 정보를 실시간 모니터링, 분석하여 변화에 대한 신속한 대응을 제공하거나, 유비쿼터스 서비스 환경에서 상황에 맞게 적시에 맞춤형 서비스를 제공하기 위해서는 이벤트 기반의 서비스 기술이 요구된다. 본 고에서는 이벤트 기반 아키텍처의 개념, 발전 방향 및 구성 요소에 대해 설명하고, 실시간으로 발생하는 대량의 이벤트를 처리하여 의미있는 상황 정보를 제공하거나 서비스를 연동하는 기술인 이벤트 스트림 처리 기술에 대해 살펴본다.

## I. 서론

유비쿼터스 서비스는 상황에 맞는 적시에 찾아가는 서비스를 제공하는 것으로, 다양한 서비스 모델 발굴과 이를 실현하기 위한 기술들이 연구 개발되고 있다. 유비쿼터스 홈 서비스 예를 보면 사람이 방에 들어 오면 자동으로 점등이 되는 서비스부터 사람이 집에 들어오면 개인의 취향 및 현재의 감정 상태 등을 고려하여 자동으로 집안의 분위기(실내 온도, 조명, 음악 등)를 조정하는 등 다양한 수준의 서비스가 가능하다. 현재 유비쿼터스 서비스는 u-home, u-office, u-shop 등 특정 공간에서 필요로 하는 서비스 제공부터 u-healthcare, u-safety, u-environment, u-education, u-traffic 등 개인 삶의 기반이 되는 각 응용 분야별로 서비스 모델 발굴을 위해 활발한 연구가 진행되고 있다.

이와 같이 상황에 맞는 서비스 혹은 상황을 예측하여 능동적인 서비스를 제공하기 위해서는 개인의 주변에서 발생하고 있는 다양한 정보, 즉 위치 정보, 환경 정보 등과 사용자의 행동 패턴과 같은 이력 정보, 더불어 개인과 직접적인 관계는 없으나 개인에게 영향을 미칠 수 있는 이벤트 등 실제세계에서 발생하는 다양한 이벤트 정보를 종합적으로 분석하여 서비스를 연동하는 기술이 요구되고 있다.

또한 기업이 비즈니스 환경 변화에 유연하고 민첩한 대응을 위해서는 기업 내·외부적인 프로세스의 지연 요소를 제거하고, 비즈니스 과정상에서 발생하는 이벤트와 비즈니스에 영향을 줄 수 있는 외부 이벤트의 조기 감지와 신속한 대응이 가능한 IT 구조를 갖추는 것이 필요하다. 이와 같은 실시간 기업(RTE) 구현을 위해 기업 내부 및 외부에서 발생하는 중요 이벤트에 조직이 신속히 대응하기 위해서는 이벤트 중심으로 비즈니스 프로세스가 구성되어야 한다[1].

본 고에서는 이벤트 기반 아키텍처(EDA)에 대한 개념, 발전 방향 및 구성 요소에 대해 II장에서 설명하고, 차세대 EDA의 핵심 구성 요소가 될 이벤트 스트림 처리 기술에 대해 III장에서 설명하고 결론을

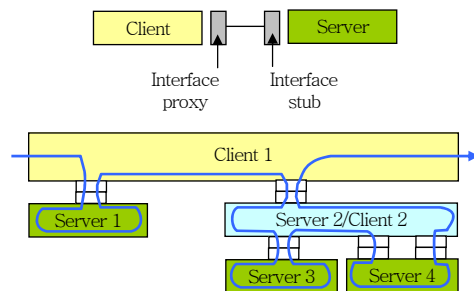
맺기로 한다.

## II. 이벤트 기반 아키텍처

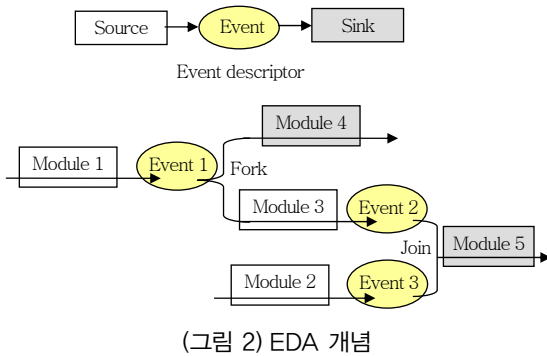
### 1. 이벤트 기반 아키텍처 개념

서비스 구축 방법론으로 서비스 지향 아키텍처(SOA)에 대한 관심이 업계에 널리 퍼져 있어 SOA 관련 제품 개발 및 SOA 기반의 서비스 구축 방법론 정립 등 다양한 활동이 이루어지고 있다. 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[2].

SOA와 EDA는 상호 보완적인 서비스 구축 방법론으로 (그림 1)과 같이 SOA의 기본 개념이 정의된 서비스 인터페이스를 이용하여 요청(request) 및 응답(response)을 통해 서비스를 연동하는 것이라면, EDA는 (그림 2)와 같이 이벤트에 대한 감지(sense) 및 대응(respond) 모델이다[2],[3]. <표 1>의 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 구



(그림 1) SOA 개념



(그림 2) EDA 개념

<표 1> SOA와 EDA 비교

구분	SOA	EDA
상호 규약 정보	서비스 인터페이스 정보	이벤트 규격 정보
연결 방식	1 : 1	N : N
흐름 제어 주체	클라이언트	이벤트 수신자
흐름 제어 방식	순차 경로	동적/병렬/비동기 방식
새로운 입력에 대한 대응	진행중엔 차단	진행중에도 반응

<자료>: Gartner, 2005.

성된다.

양 서비스 구축 방법론의 등장 배경이 구축하고자 하는 서비스 자체의 속성에 따른 차이를 반영한 것 뿐만 아니라, SOA가 서비스의 재활용 및 공유가 주목적이라면, EDA는 실시간 서비스 제공이 주목적이라고 할 수 있다.

## 2. 이벤트 기반 아키텍처 발전 방향

EDA를 지원하는 다양한 기술들이 실제 응용들에서 이미 사용되고 있으며 이벤트 처리 방법에 따라 크게 다음과 같이 나눌 수 있다[3],[4].

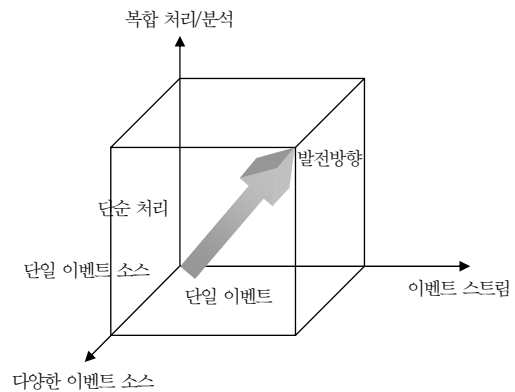
- 단순 이벤트 처리: 발생한 이벤트들은 모두 의미 있는 이벤트로 간주하고 각각의 이벤트 내용에 따라 대응되는 액션을 수행한다. Publish/subscribe 방식이나 중재 방식에 의해 이벤트 처리를 제공한다.
  - Publish/subscribe 방식: Subscribe시 관심 있는 이벤트를 지정함으로써 이벤트 소스로

부터 관련 이벤트들을 직접 수신 받는다. 애플리케이션 서버, MOM, ESB, 웹 서비스 제품 등에서 제공하고 있다.

- 중재 방식: 중재자가 모든 이벤트를 수집하여, 이벤트의 내용에 따라 이벤트 수신자에게 전달한다. Publish/subscribe 방식에 비해 더 세부적인 이벤트 분류가 가능하며, 이벤트 메시지의 변환 기능을 제공한다. 통합 브로커 시스템이 이에 해당된다.

- 스트림 이벤트 처리: 의미 있는 이벤트와 무의미한 이벤트가 같이 발생하는 대량의 이벤트 스트림을 대상으로 하며, 필터링 등을 수행하여 의미 있는 이벤트 정보만 뽑아서 응용에 전달 혹은 서비스와 연동한다. RFID, USN 미들웨어 등이 이에 해당된다.
- 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 패키지에 내장되어 제공되거나 별도의 복합 이벤트 처리 시스템으로 제공되기도 한다.

이벤트 기반 서비스를 위한 기술 발전은 (그림 3)과 같이 각각의 독립적인 이벤트를 대상으로 처리하던 것에서 점차 여러 소스로부터 발생하는 다양한 이벤트, 그리고 대량의 이벤트 스트림을 처리하는



(그림 3) EDA 발전 방향

방향으로 발전하고 있다. 또한 이벤트 처리 방법도 이벤트 내용에 따른 단순 분류에서 이벤트 내용에 대한 세부 분류 및 필터링뿐만 아니라 여러 이벤트 간의 시간상, 공간상, 의미상의 관계를 분석하는 방향으로 발전하고 있다.

### 3. 이벤트 기반 아키텍처 구성 요소

EDA를 지원하기 위한 플랫폼의 구성 요소는 (그림 4)와 같다[4].

- 이벤트 메타 데이터: 이벤트 메시지의 모습을 정의한 이벤트 규격과 이벤트를 처리하기 위한 규칙 등 이벤트 관련 메타 데이터로 이벤트 소스 및 수신자, 이벤트 처리자 등이 공유할 수 있어야 한다.
- 이벤트 프로세싱: 이벤트 규격을 따르는 이벤트 메시지를 대상으로 이벤트 처리 규칙에 따라 실제 처리를 수행하며 이벤트 메시지의 저장 관리를 담당한다.
- 이벤트 구축 도구: 이벤트 규격, 이벤트 처리 규칙 등을 정의하는 것을 지원하는 도구
- 이벤트 관리 도구: 이벤트 처리 상태 모니터링,

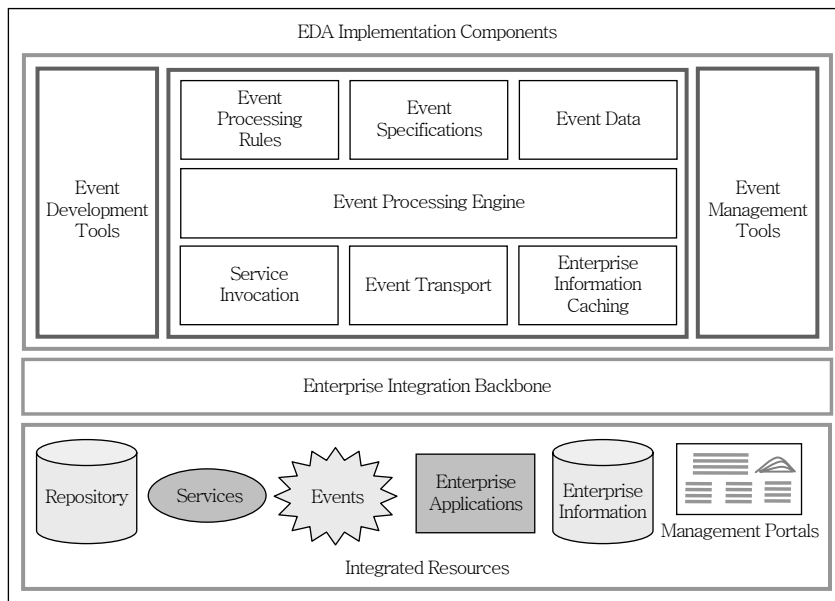
이벤트 흐름에 대한 모니터링 등 전체 시스템 관리 기능을 제공하는 도구

- 엔터프라이즈 통합: 외부 서비스 호출, 이벤트 전송, 엔터프라이즈 정보 접근 등과 같이 이벤트 입·출력시 혹은 이벤트 처리시 외부 자원과의 연동을 담당하기 위한 것으로, 기존 응용 환경에서 사용하고 있는 엔터프라이즈 통합 시스템 (enterprise integration backbone)을 이용하여 연동한다.
- 통합할 외부 자원: 다양한 이벤트를 발생하는 소스와 이벤트 기반의 액션으로서 구동되어야 할 외부 서비스 등을 의미하며, 엔터프라이즈 통합 시스템에 의해 EDA 플랫폼에 통합된다.

## III. 이벤트 스트림 처리 시스템

### 1. 이벤트 스트림 처리 시스템 개요

앞장에서 EDA의 발전 방향에서 제시했듯이 이벤트 처리는 이기종 소스로부터 발생하는 대량의 이벤트 데이터에 대한 처리를 통해 이벤트간의 영향을



(그림 4) EDA 구성 요소

분석할 수 있는 기능을 제공하여야 한다.

이벤트 스트림 처리 시스템은 연속적으로 발생되는 이벤트 데이터를 실시간으로 분석, 판단하여 대응되는 액션을 제공하는 시스템으로 차세대 EDA의 핵심 구성 요소가 될 것으로 보여진다. 이벤트 스트림 처리 시스템이 갖추어야 할 주요 특성은 다음과 같다[5].

① 이벤트 데이터의 흐름에 따른 처리

기존 데이터베이스 관리 시스템과 같이 이벤트 데이터를 디스크에 저장 후 처리하는 방식은 처리 시간의 지연으로 실시간 처리에 적합하지 않으므로 이벤트가 도착하는 즉시 처리할 수 있어야 한다. 또한 응용의 요구에 의해서만 처리하는 수동적인 시스템이 아닌 이벤트 발생에 따라 능동적으로 처리할 수 있어야 하므로 한 번 요구에 의해 연속적으로 처리하는 기능을 제공하여야 한다.

② 고수준의 이벤트 처리 언어 제공

이벤트에 대한 다양한 질의 혹은 분석 요구를 정의할 수 있는 방법이 존재하여야 한다. C++, Java 와 같은 프로그래밍 언어 수준이 아닌 SQL과 같은 고수준의 언어가 제공되어야 다양한 응용에서 이벤트 기반의 서비스 구축이 용이하다. 이벤트 처리 언어에는 처리 이벤트 대상에 대한 영역을 결정하는 원도 개념을 지원하여야 하며 저장 데이터와 이벤트 스트림에 대한 통합 질의 처리와 응용에서 요구하는 분석 기능을 통합할 수 있어야 한다.

③ 이벤트 스트림의 불완전성에 대한 대처

이벤트 도착 지연, 손실, 어긋난 순서로 도착 등 예기치 않은 상황이 발생하여도 이벤트 처리에 문제가 없어야 한다. 예를 들어, 이벤트가 손실되었을 때 손실된 이벤트를 무한정 기다리는 상황이 발생하지 않도록 타임 아웃 등과 같은 대처 방안을 제공하여야 한다.

④ 시간 기반 이벤트에 대해 예측 가능한 처리

처리 결과가 항상 동일하게 나올 수 있도록 시간

에 따른 순서화 및 정해진 처리 방법을 제공하여야 한다.

⑤ 저장 데이터와 이벤트 스트림의 통합 처리

이력 정보와 실시간 발생 데이터를 같이 활용할 수 있어야 하며, 처리 시간 지연을 최소화하기 위해 이력 데이터도 메인 메모리에 관리할 수 있도록 내장 DBMS를 이용하여 관리하여야 한다.

⑥ 고가용성 제공

고가용성을 제공하여야 하며, 백업 시스템과 운영 시스템의 동기화 주기를 단축하여 문제 발생시 실시간으로 대체할 수 있는 수단을 제공하여야 한다.

⑦ 분산 처리 제공

시스템의 확장성을 위해 분산 처리가 가능하여야 하며 응용 시스템에게 투명하게 부하 분산을 지원하여야 한다.

⑧ 실시간 처리

초당 수만에서 수십만 이벤트 메시지를 처리하여 microsec, millisc 안에 처리 결과를 제공할 수 있어야 한다.

상기 설명한 이벤트 스트림 처리 시스템의 특성들은 모두 중요하지만, 무엇보다도 실시간 처리를 위해 이벤트 데이터의 흐름에 따른 처리 기법을 제공하는 것이 가장 중요하다고 볼 수 있고, 이를 제공하기 위해서는 기존의 처리 방법과는 다른 새로운 방법 등이 이용되어야 한다. 예를 들어 데이터베이스 관리 시스템에서는 데이터 색인을 활용하여 데이터의 검색 범위를 줄임으로써 빠른 접근을 제공하였다. 그러나 이벤트 스트림 처리에서는 데이터에 색인을 구성하는 것이 아니라 연속 질의에 색인을 구성하여 입력되는 데이터를 대상으로 수행할 연속 질의의 수를 줄이는 방법을 활용한다. 연속 질의 색인 외에도 연속 질의간에 질의 처리 결과의 공유에 의해 처리 시간을 단축시키는 방법 등 이벤트 스트림 처리에 효과적인 다양한 방법에 대한 연구가 활발히 수행되고 있다.

## 2. 사례 시스템

차세대 EDA 기반 시스템으로 활용 가능한 이벤트 처리 기술은 RFID/USN 미들웨어, 이벤트 스트림 처리 시스템, 데이터 스트림 관리 시스템 등으로 연구 개발이 수행되고 있다.

RFID 미들웨어는 RFID, 센서 등에서 발생하는 연속 데이터에 대한 필터링을 수행하여 의미있는 데이터로 정제하여 응용에 전달하는 시스템[6]으로 Oracle Edge Server[7], SUN Java System RFID Software[8], BEA WebLogic RFID Edge Server (ConnecTerra의 RFTagAware를 통합)[9] 등 다양한 제품이 나와 있다. 국내에서도 RFID 미들웨어에 대한 연구 개발이 활발히 수행되고 있어 ECO, 엘릭스, 한울주식회사, LG CNS 등에서 연구 개발을 수행하고 있으며, 전산원에서 실시하는 시범 서비스 사업에도 국내 제품들이 널리 활용되고 있다. RFID 미들웨어는 RFID 혹은 센서와 같이 특정 소스로부터 발생하는 스트림 데이터 처리가 목적이므로 주로 필터링 처리를 제공하고 있으며 다양한 분석 기능이 미비하다. 그러므로 EDA 기반 시스템으로 활용되기 위해서는 비즈니스 이벤트 처리를 위해 개발된 BAM과 연계가 필요하다. Oracle에서는 Sensor Edge Server와 기존 BAM 솔루션을 통합하여 EDA Suite라는 제품을 발표하고 있다[10].

이벤트 스트림 처리 시스템은 대량의 이벤트 스트림을 대상으로 이벤트들간의 다양한 관계 분석을 통해 대응되는 액션을 수행할 수 있게 하는 시스템으로 iSpheres[11], Progress ESP(Apama를 인수 합병)[12] 등이 있다. 이벤트 스트림 처리 시스템은 네트워크 모니터링, 주식 자동 매매 시스템, 센서 데이터 모니터링 등 특정 응용에서 발생하는 이벤트 데이터를 처리하던 기술들을 기반으로 범용의 이벤트 스트림 처리 시스템으로 발전시킨 것이다.

데이터 스트림 관리 시스템은 기존 데이터베이스 관리 시스템으로는 비즈니스 응용에서 실시간으로 발생하는 대량의 데이터를 처리하는 데 한계가 있으므로 스트림 데이터의 실시간 처리를 필요로 하는

응용에서 활용할 수 있는 시스템을 제공하고자 개발된 것으로 대표적인 시스템으로 Streambase[13]가 있다. 데이터 스트림 관리 시스템은 데이터 스트림으로부터 원하는 정보를 검색하는 기능을 주로 강조하고 있으며, Streambase에서도 SQL과 유사한 StreamSQL이라는 고수준의 연속 질의어를 제공하고 있다.

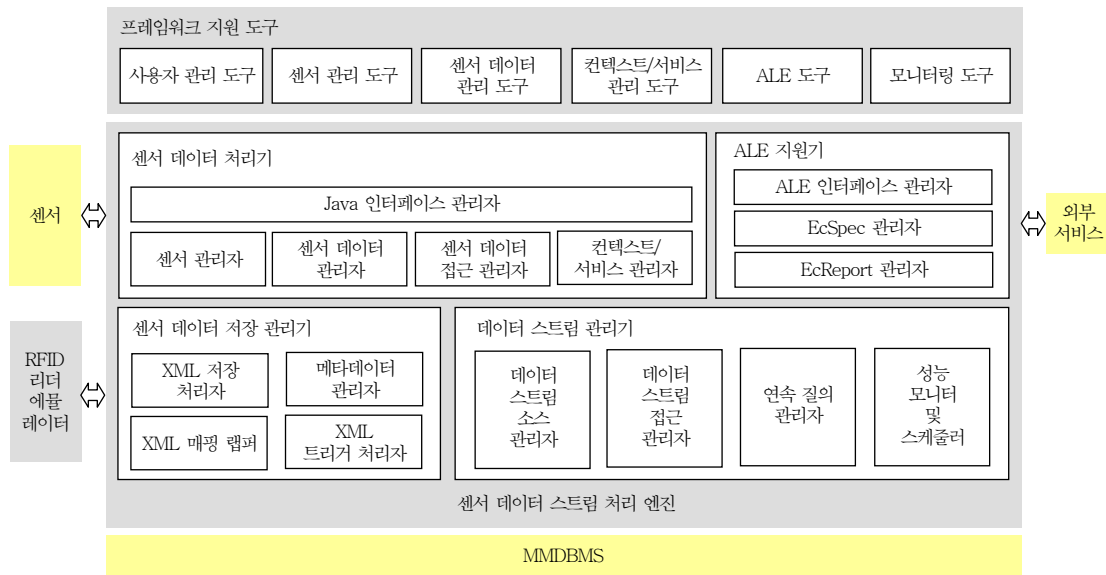
## 3. UbiCore 시스템

ETRI 데이터베이스연구팀에서는 2004년부터 유비쿼터스 서비스 플랫폼 기술 개발의 일환으로 대용량 센서 데이터 스트림 처리 미들웨어인 UbiCore를 개발하고 있다. UbiCore 시스템은 RFID, USN 등에서 발생하는 대량의 센서 데이터를 수집, 필터링, 검색 및 변환 처리 등을 실시하여 응용에서 필요로 하는 센서 데이터를 제공해 주는 센서 데이터 처리용 미들웨어로 다음과 같은 기능을 제공한다.

- UHF 900MHz 대역을 지원하는 RFID 리더와 연동 및 소켓 기반의 범용 스트림 소스 연동 제공
- XML 기반의 스트림 데이터 실시간 처리를 제공하며, 컨텍스트를 표현할 수 있도록 윈도 개념이 통합된 XQuery 기반의 연속 질의어를 제공
- XML 기반 센서 데이터의 필터링, 검색 기능뿐만 아니라 RDBMS를 기반으로 저장 기능을 제공하며, 스트림 데이터와 저장 데이터에 대한 통합 검색을 지원
- 컨텍스트 기반의 서비스가 가능하도록 web service, JMS, http, socket, file system 등 다양한 프로토콜을 통해 외부 응용에 센서 데이터 전달 및 서비스 연동을 지원
- 센서 데이터에 대해 데이터 구조 기반, 데이터값 기반 등 세밀한 수준의 접근 제어 기능을 제공
- 응용 프로그래밍 인터페이스로 Java 인터페이스와 ALE 표준 규격을 따르는 웹 서비스 인터페이스를 지원

UbiCore는 (그림 5)와 같은 시스템 구조를 갖는다. 크게 프레임워크 지원 도구와 엔진으로 구성되





(그림 5) UbiCore 구조

며, 엔진은 ALE 지원기, 센서 데이터 처리기, 데이터 스트림 관리자, 센서 데이터 저장 관리기로 구성된다.

- ALE 지원기: EPCglobal의 표준 규격인 ALE 인터페이스를 지원하는 것으로 ALE 인터페이스를 내부 인터페이스로 변환 처리하는 기능을 제공한다.
- 센서 데이터 처리기: 센서 데이터 발생 소스와 연동하여 데이터를 수집 통합하는 기능, 처리된 결과를 응용에게 전달하기 위해 정보 전달 프로토콜 및 서비스 연동 기능을 제공한다.
- 데이터 스트림 관리자: 데이터 스트림에 대해 연속 질의를 처리하는 모듈로 스트림 데이터가 일련의 과정을 거쳐 처리될 수 있도록 데이터 전달 큐 관리 기능과 XQuery 기반의 데이터 스트림 처리어인 XQueryStream 처리 기능을 제공한다.
- 센서 데이터 저장 관리자: 데이터 스트림 중 필요한 데이터를 주기억 상주 기반의 RDBMS에 저장 관리하고, 데이터 스트림과 저장된 이력 데이터를 통합한 연속 질의를 효과적으로 제공하

기 위해 RDBMS 기반으로 XML 트리거를 제공한다.

UbiCore는 XML 모델 기반으로 범용의 데이터 스트림 처리 기술을 제공하므로 새로운 형식의 이벤트 스트림 처리 수용이 용이하며, 또한 스트림 데이터와 저장된 이력 데이터의 통합 질의가 가능하므로 이벤트 기반 아키텍처를 지원하는 시스템 구축시 핵심 기반 기술로 활용이 가능하다. 앞으로 실시간 처리 성능 개선 및 분산 처리를 위한 확장이 지속적으로 수행될 예정이다.

## IV. 결론

EDA는 실시간 기업 환경 구축을 위해서나 개인의 상황에 맞는 맞춤형 서비스 제공을 위해서 널리 활용될 수 있는 서비스 구축 모델이다.

본 고에서는 EDA의 기술 발전 방향을 분석하며, 향후 이벤트 스트림 처리 시스템이 EDA에서 중요한 역할을 할 것임을 제시하였다. 그러므로 국내에서도 이벤트 스트림 처리 시스템에 대한 연구 개발을 적극적으로 수행하여야 할 것으로 보여진다. 현

재 국내에서는 RFID/USN 등 유비쿼터스 컴퓨팅에 대한 많은 관심과 연구 개발 투자가 이루어지고 있어, 이미 RFID 미들웨어, 센서 데이터 스트림 처리 미들웨어 개발 등 관련 기술을 확보하고 있으므로 이를 기반으로 범용의 이벤트 스트림 처리 시스템의 확장 개발을 통해 EDA 분야의 기술 선도 및 시장 확보가 필요하다.

### 약어 정리

ALE	Application-Level Events
BAM	Business Activity Monitoring
BPM	Business Process Management
DBMS	Database Management System
DSMS	Data Stream Management System
EDA	Event-Driven Architecture
ESB	Enterprise Service Bus
ESP	Event Stream Processing
MOM	Message-Oriented Middleware
RFID	Radio Frequency Identification
RTE	Real-Time Enterprise
SOA	Service-Oriented Architecture
USN	Ubiquitous Sensor Network

### 참고 문헌

- [1] 백한진, "RTE 구현을 위한 전략," SDS Consulting Review, 2004.
- [2] Daryl C. Plummer, "Software Architecture Will Evolve from SOA and Events to Service Virtualization," Gartner, Mar. 2005.
- [3] Roy W. Schulte, "Event-Driven Applications: Definition and Taxonomy," Gartner, July 2003.
- [4] Brenda M. Michelson, "Event-Driven Architecture Overview," Patricia Seybold Group, Feb. 2006.
- [5] Mike Stonebraker, Uguur Cetintemel, and Stanley Zdonik, "The Eight Rules of Real-Time Stream Processing," Streambase, 2006.
- [6] Michael J. Liard, "A White Paper on: Radio Frequency Identification(RFID) Middleware Solutions: Global Market Opportunity," VDC, Aug. 2004.
- [7] [http://www.oracle.com/technology/products/sensor\\_](http://www.oracle.com/technology/products/sensor_edge_server)  
[edge\\_server](http://www.oracle.com/technology/products/sensor_edge_server)
- [8] <http://www.sun.com/software/products/rfid>
- [9] <http://www.bea.com>
- [10] <http://www.oracle.com/technology/soa/events.html>
- [11] <http://www.ispheres.com>
- [12] [http://www.progress.com/realtime/products/esp\\_plat-](http://www.progress.com/realtime/products/esp_platform)  
[form](http://www.progress.com/realtime/products/esp_platform)
- [13] <http://www.streambase.com>