

# e-비즈니스 응용 시스템을 위한 컴포넌트 개발에 관한 연구

김 행 곤<sup>†</sup>

## 요 약

인터넷과 인트라넷 기반의 전자적 서비스를 통해 비즈니스 거래가 전자상거래 공간에서 보다 유용하고 확장된 비즈니스 개념들을 제공하게 하였다. 특히, 컴포넌트 기반 e-비즈니스 기술은 시스템과 응용 레벨 모두에서 e-비즈니스 이슈들을 해결하는 방향이 되고 있다. 컴포넌트 기반 시스템은 네트워크 간의 상호동작성, 여러 네트워크 환경들간의 이동성 등의 특성을 포함한다. 따라서, 전통적인 프로세스와 개발 모델들은 빠르게 변화하는 환경에 부적절한 구조를 가지므로, e-비즈니스 응용 개발자들은 컴포넌트 기반 웹응용을 개발하기 위한 매커니즘으로 웹서비스 기술을 시도하고 있다. 본 논문에서는 e-비즈니스 시스템을 구축하기 위하여 웹서비스 기반의 e-비즈니스 시스템 아키텍처에 특히 초점을 맞추고, 다양한 상황에서 e-비즈니스 도메인에 적용하기 위한 설계패턴을 식별하고 명세화한다. 또한 제안된 프로세스를 적용하여 e-비즈니스 응용을 위한 컴포넌트들을 개발하도록 프로토타입과 프레임워크를 연구하고 CAS(Customer Authentication system)을 개발하는 사례를 보여준다. 끝으로, e-비즈니스에서 컴포넌트 기반 개발 방향을 보여줄 것이다.

## A Study on the Component Development For e-Business Application Systems

Haeng-Kon Kim<sup>†</sup>

### ABSTRACT

An electronic services which are based on Internet/Intranet business transactions are available to e-market places and get the broader business concepts. Component-based e-commerce technology is a recent trend towards resolving the e-commerce challenge at both system and application levels. The basic capabilities of component based systems should include the plug and play features at various granularities, interoperability across networks and mobility in various networking environments. E-business application developers are attempting to more towards web-service as a mechanism for developing component-based web-applications. Traditional process and development models are inadequately architected to meet the rapidly evolving needs for the future of scalable web services. In this thesis, we focus specifically on the issues of e-business system architecture based on web service for establishing e-business system. We specifies and identifies design pattern for applying e-business domain in the context of multiple entities. We investigate prototype and frameworks to develop components for e-business application based suggested process. We present a worked example to demonstrate the behavior of Customer Authentication System(CAS) development with component and recommend process. Finally, we indicate and view on future directions in component-based development in the context of electronic business.

키워드 : e-비즈니스(e-Business), CBD(Component Based Development), MDA(Model Driven Architecture), Web Service

### 1. 서 론

비즈니스를 위한 수많은 자원들과 요소들이 복잡하게 연결되어 있고 전자상거래 관리 시스템은 처리해야할 자료와 데이터가 인터넷을 통해 서로 상호작용하고 있다. 보편화된 인터넷과 웹은 전통적인 비즈니스에서 벗어나 인터넷의 효율적인 사용을 개발하여 자신들의 업무에 맞추고 또한 경제 분야에도 새롭고 더 나은 서비스를 고객에게 제공하는 e-비즈니스 형태의 접근이 이루어지고 있다[1]. 이 흐름에

따라 웹을 사용한 새로운 모델로써 XML(eXtensible Markup Language)기반으로 등장한 웹서비스는 이미 표준화된 기술들 즉, SOAP(Simple Object Access Protocol), WSDL(Web Service Description Language), UDDI(Universal Description, Discovery, and Integration) 등을 기반으로 발전하였으며 각 표준들과 상호작용하는 응용들이라고 할 수 있다. 현재 e-비즈니스에는 XML을 기반으로 웹 표준들을 서로 상호 작용시킴으로써 고객이 원하는 서비스를 제공하는 웹서비스 개념을 포함시키는 추세이다. e-비즈니스 응용은 비즈니스 트랜잭션의 제공자와 소비자 측면에서 단말 사용자의 자동화를 위한 플랫폼이 기반되며 이 새로운 모

<sup>†</sup> 종신회원 : 대구가톨릭대학교 컴퓨터정보통신공학부 정교수  
논문접수 : 2004년 2월 25일, 심사완료 : 2004년 5월 21일

델로 비즈니스 응용간의 자동화를 위해 실시간 응용인 웹 서비스가 제공하는 핵심기술이 요구된다[2].

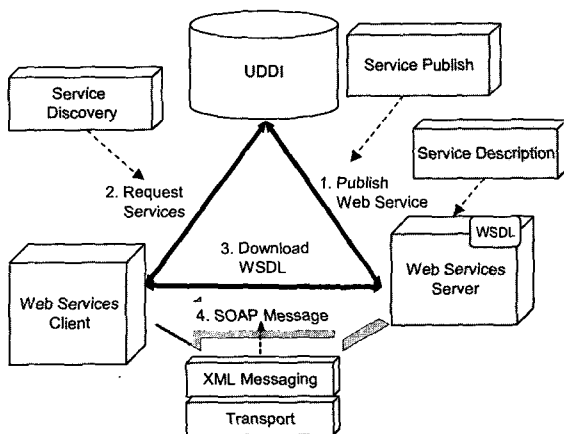
본 논문은 웹서비스를 이용한 차세대의 e-비즈니스 어플리케이션 개발하기 위한 필수적인 기본 기술들을 계층적으로 표현한 재사용 가능한 아키텍처를 기술하고 e-비즈니스 도메인의 공통적인 설계패턴을 식별 및 명세화한다. 또한, e-비즈니스 응용을 위한 개발 프로세스를 제안하고 Customer Authentication system(CAS)을 통해 사례 연구한다.

## 2. 관련 연구

### 2.1 웹서비스

웹서비스는 XML기반으로 웹 표준들의 상호작용을 통해 고객이 원하는 서비스를 제공하는 응용이다. W3C에서는 “웹서비스는 공개된 인터페이스와 바인딩을 XML을 사용하여 정의, 기술하므로 발견이 가능해지고 인터넷 기반의 프로토콜을 거쳐 XML 기반의 메시지를 사용하는 다른 소프트웨어 응용들과 직접적인 상호작용이 가능한 URI(Uniform Resource Identifier)에 의해 식별된 소프트웨어 시스템이다.”라고 정의하였다[3].

웹서비스에 기반이 되는 표준들로는 SOAP, UDDI, WSDL이 있다. SOAP은 웹 서비스에 접근하기 위해 사용되는 XML 기반의 통신 프로토콜이다[3]. WSDL은 웹서비스 기술 언어로써 오퍼레이션들과 관련된 데이터 타입들, 지원되는 전송 프로토콜 등을 포함한다. UDDI는 웹서비스들을 출판하고 발견하기 위한 SOAP 기반의 API(Application Program Interface)이다. (그림 1)은 웹서비스 기본 아키텍처를 나타낸다.



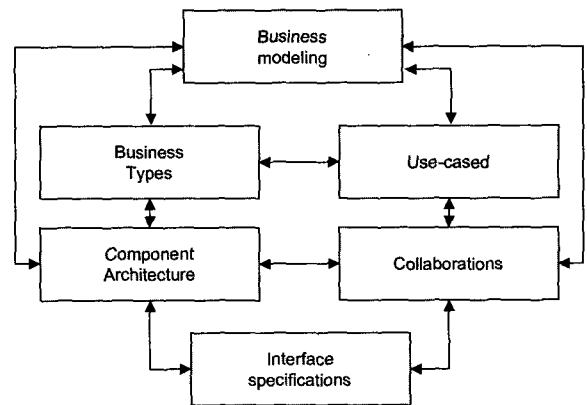
(그림 1) 웹 서비스 구조

### 2.2 컴포넌트 기반 개발(CBD : Component Based Development)

컴포넌트 기반 개발은 독립적인 기능을 담당하는 다양한 컴포넌트 소프트웨어의 집합으로부터 해당 업무의 수행에

필요한 기능을 담당하는 하나 이상의 컴포넌트를 결합함으로써 해당 업무를 위한 소프트웨어를 개발하는 기술을 말한다[4]. 컴포넌트 기반 개발 기술은 과거 구조적 방법이나 객체 지향 기술이 제대로 해결하지 못한 소프트웨어 생산성, 소프트웨어 재사용성, 시스템 유지 보수성을 향상시킬 수 있는 대안으로 주목받고 있다[5].

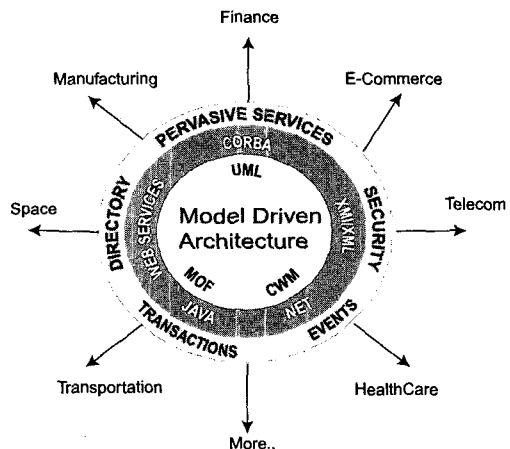
CBD 모델링 기술은 객체지향의 UML(Unified Modeling Language) 분석 및 설계 모델링에 대한 표기법과 의미를 이용한다. 컴포넌트 기반 개발 모델링에는 비즈니스 모델링, 비즈니스 타입 모델링, 유즈 케이스 모델링, 컴포넌트 아키텍처 모델링, 상호협동 모델링이 있다[4-6]. (그림 2)는 컴포넌트 기반 개발 모델링 맵을 보여준다.



(그림 2) 컴포넌트 기반 개발 모델링 맵

### 2.3 MDA(Model Driven Architecture)

최근 OMG(Object Management Group)에서 소개된 MDA는 모든 컴포넌트 기술요소의 표준 메타 모델을 정의하고, 이를 기반으로 각 구성요소를 정의함으로써 모든 컴포넌트 기술 요소들의 호환성 및 시스템간 동작성을 보장하고 있다. (그림 3)은 언어와 미들웨어에 중립적인 MDA를 나타낸다.



(그림 3) Model driven Architecture

MDA에서 PIMs(Platform Independent Models)는 UML과 같은 플랫폼 독립 모델링 언어로 표현되고, 메타모델을 기반으로 PIM을 자동으로 각 구현 환경에 적합한 PSM(Platform Specific Model)으로 변환할 수 있는 구조를 정의한다. <표 1>은 MDA 기술의 핵심으로 OMG에서 정의한 모델링 표준인 UML, CWM(Common Warehouse Meta-model), MOF(Meta Object Facility), XMI(XML Metadata Interchange)에 대해 기술한다[7].

<표 1> 표준 모델링 유형

UML	객체 및 컴포넌트 시스템을 표현하기 위한 표준 언어로서 시스템의 아키텍처 및 구성 객체, 객체간의 상호작용 등을 모델링
CWM	데이터 웨어하우징과 비즈니스 분석 도메인에서 기초가 되는 비즈니스와 기술적인 메타데이터 모듈을 표현하는 메타모델
MOF	시스템의 객체지향 모델을 구성하기 위해 사용되는 메타모델들의 본질적인 요소와 구문, 구조를 정의
XMI	XML 태그가 XML에서 순서화된 MOF 중심 모델을 표현하기 위해 사용되는 방법을 정의

### 3. 컴포넌트 기반 e-비즈니스 시스템 설계 및 구현

e-비즈니스 시스템은 트랙잭션의 제공자와 소비자 측면에서 단말 사용자의 자동화를 위한 플랫폼을 기반하여 제안한 모델로 비즈니스 응용간의 자동화를 위한 실시간 응용 웹서비스를 목표로 한다. 본 절에서는 e-비즈니스 시스템의 아키텍처를 기술하고 공통적인 설계패턴을 식별 한다. 또한 e-비즈니스 시스템 개발을 위한 프로세스를 제안한다.

#### 3.1 e-비즈니스 시스템 아키텍처

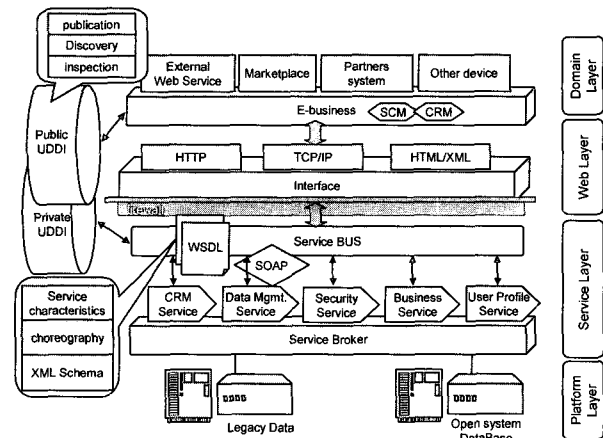
웹서비스 기반의 e-비즈니스 시스템 아키텍처의 논리적인 뷰를 (그림 4)에서 나타낸다. 최상위 계층은 전통적인 엔터프라이즈 아키텍처 n 계층을 표현한다. 비즈니스 활동을 위한 단말장치들, 마켓플레이스 서비스 요구자들의 서비스 기술을 포함하며 동적인 e-비즈니스를 위해 기업간의 웹서비스를 Public UDDI를 통해 통합된 디렉토리 통신을 가능하도록 한다.

다음 하위 계층은 요구자와 공급자 컴포넌트들이 상호작용하기 위한 물리적인 정보교환 활동을 한다. 인터페이스는 개념적으로 웹서비스 계층의 기초 네트워크로 전송, 교환되는 정보의 패키지를 위해 패키징, 문맥, 라우팅, 정책 등을 웹서비스 메시지에 부착될 수 있도록 하는 프레임워크 확장으로 계층화된다.

다음 계층의 서비스 버스는 전체 비즈니스의 활동 경로가 되며 다른 비즈니스 컴포넌트들과도 통합하게 한다. 서비스 버스는 비즈니스 프로세스 요소들을 분류하고, 자료화하며 기업간 또는 기업 내의 정보교환을 위한 Public/Private UDDI

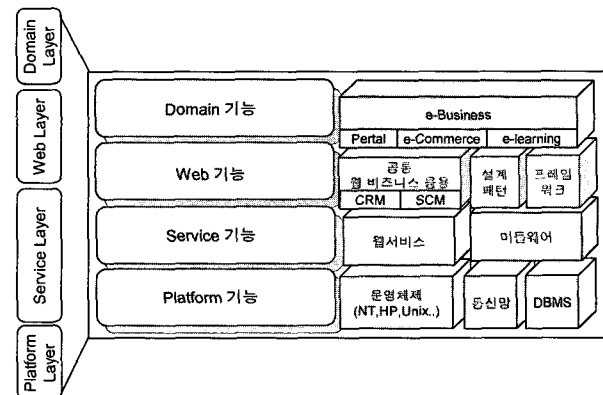
표준으로 웹서비스들을 출판하기 위한 아케텍처 계층이 된다.

서비스 브로커는 여러 미들웨어 기술들과 APIs 또는 전통적인 데이터베이스 연결자 등으로 구성된다. 이들 기술들은 기존 데이터, ERP(Enterprise Resource Planning), 올라클과 같은 개방 데이터베이스 시스템의 접근을 제공한다. 서비스 브로커는 웹 서비스 표준들과 이종의 비즈니스 응용들 간의 상호 동작성을 제공한다. 이 전반적인 아키텍처는 데이터 관리, 웹 사이트 관리 등의 전체 비즈니스 관리 기능과 보안 관리 기능, QoS 관리 기능에 걸쳐져 있다.



(그림 4) e-비즈니스 시스템 아키텍처

정의된 e-비즈니스 시스템 아키텍처를 기반으로 실제적인 응용 개발을 위해 필요로 하는 기술들의 계층이 정의어진다. (그림 5)는 e-비즈니스 시스템 개발에 요구되어지는 컴포넌트 기술들을 계층적으로 표현한 것이다. 이들 계층들은 웹 응용 개발에 필요한 기술의 적용 영역과 범위 및 구현 환경과의 제약사항들을 기준으로 분류되어졌고 각 계층별 기능은 도메인 기능, 웹 기능, 서비스 기능, 플랫폼 기능으로 나누어지며 이들 기능적 모듈들은 컴포넌트 아키텍처 정의, 컴포넌트 식별을 위한 분류 조건으로 적용될 수 있다.



(그림 5) e-비즈니스 시스템 컴포넌트 계층 구조

플랫폼 계층은 분산 컴퓨팅 환경에서의 다양한 통신망과 물리적 플랫폼 구성 요소의 기능을 제공하고 실제 통신망과 통신서비스 및 응용 계층과 연계되는 하부적인 물리적 플랫폼이다.

서비스 계층은 다양한 웹 응용에서 이종의 통신 프로토콜과 데이터베이스 연동을 위한 미들웨어 기능을 제공하고 또한, 응용에 하부적인 기능 수행에 필요한 공통 컴포넌트와 서비스들을 포함한다.

웹 계층은 웹 응용 개발에 공통적으로 요구되는 컴포넌트와 통신 서비스 응용을 제공하는 계층으로 확장적인 컴포넌트들의 집합과 웹 응용 프레임워크 및 설계패턴까지도 포함하고 작은 규모의 비즈니스 영역에서 공통으로 사용되는 공통 컴포넌트와 커스터마이징함으로써 도메인 컴포넌트로 이용 가능한 형태의 컴포넌트들을 포함한다.

도메인 계층은 다양한 웹 응용 도메인 영역별 수행되어질 수 있는 기능을 포함하며 다양한 요구사항이나 주체 등의 외부 변수에 따라 서로 다르게 전개 가능하다.

3.2 e-비즈니스 시스템 개발을 위한 설계패턴

e-비즈니스 도메인에 웹서비스를 적용한 시스템 개발에 필요한 시스템 정보와 자원을 패턴화하기 위한 시스템 설계패턴 추출 방법과 추출 과정에 따라 시스템 요소들 간의 관련성을 일반화함으로써 재사용가능한 설계패턴을 식별하고 분류한다.

3.2.1 e-비즈니스 설계패턴의 식별

e-비즈니스 환경에 적용하고 사용 가능하도록 하는 비즈니스 환경 변화에 따라 설계패턴들을 <표 2>와 같이 식별

한다.

eCRM(electronic Customer Relationship Management)은 온라인 상에서 고객의 행동과 성향을 분석해 고객만족을 극대화하고 실시간에 1대 1 마케팅을 실현을 포함하고 eSCM(electronic Supply Chain Management)는 웹기반으로 구매자와 공급자를 연결해 주는 역할을 하며 많은 기업이 인터넷을 통해 공급과 구매를 할 수 있도록 하고 공급자에게는 많은 매출을, 구매자에게는 좋은 거래선을 만들 수 있도록 한다. eWorkplace는 업무에 보다 효율적인 시스템을 구축하고 이러한 시스템을 시간과 장소에 구애없이 활용하는 Company Portal, Business Intelligence(BI), e-Learning을 포함하며 ePlatform은 기존 네트워크는 물론 인터넷과 모바일 등 멀티채널을 지원하여 고객, 공급자, 직원이 다양하게 활용할 수 있도록 플랫폼을 구축한다.

<표 2> 설계패턴의 식별

eCRM	<ul style="list-style-type: none"> <li>고가치의 1:1 마케팅</li> <li>관계 마케팅</li> </ul>
eSCM	<ul style="list-style-type: none"> <li>공급자 관리 효율성</li> <li>자원 비용과 시간 절감</li> </ul>
eWorkplace	<ul style="list-style-type: none"> <li>효율적인 시스템 구축</li> <li>기업 포탈, BI(Business intelligence), e-Learning</li> </ul>
ePlatform	<ul style="list-style-type: none"> <li>다중 채널을 지원</li> <li>EAI(Enterprise Application Integration), 최적화된 아키텍처</li> </ul>

3.2.2 e-비즈니스 시스템의 설계 패턴 정의

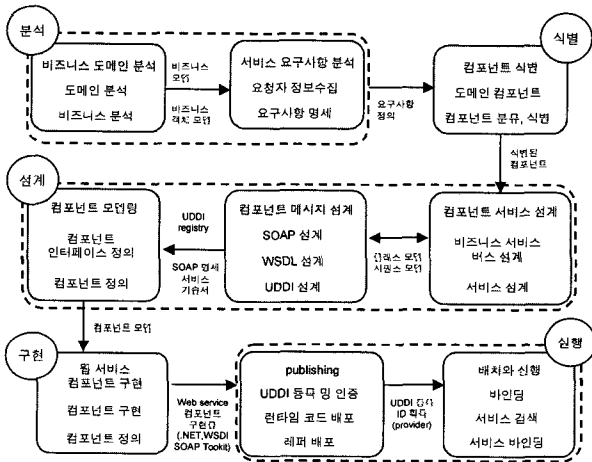
기술한 시스템 아키텍처를 기반으로 식별한 웹서비스 기반의 e-비즈니스 시스템 설계패턴 내용과 구조를 정의한다 <표 3>.

<표 3> e-비즈니스 시스템의 설계패턴

패턴 종류	패턴 이름	패턴 내용	패턴 구조
eCRM	비즈니스 서비스 도메인 패턴	<ul style="list-style-type: none"> <li>비즈니스 액터에 의해 요구되는 성공적인 목적을 반영하여 여러 서비스들은 UDDI 표준에 따른 서비스구현과 인터페이스의 세부사항을 포함하는 서비스 기술서가 작성되고 등록됨</li> <li>추상 서비스는 하위 서비스 클래스의 속성을 위임</li> <li>서비스는 비즈니스 맥락에 의존</li> </ul>	
eSCM	UDDI 등록기 패턴	<ul style="list-style-type: none"> <li>비즈니스 컴포넌트 요소들이 가지는 여러 UDDI의 상호작용 패턴</li> <li>UDDI는 public/private로 구성</li> <li>기업간 또는 기업내 서비스 정보를 SOAP을 통해 메시지를 교환하여 통합 가능</li> </ul>	
eWorkplace	웹서비스 액터 패턴	<ul style="list-style-type: none"> <li>공급자 측면 또는 소비자 측면의 웹서비스 업체 기능에 따른 분명한 역할 배분 필요</li> <li>특히, 웹서비스 브로커는 웹 비즈니스 시장 필수요소로써 서비스 제공자와 웹서비스 사용자간의 연결을 교차역할</li> </ul>	
	서비스 발견 엔진 패턴	<ul style="list-style-type: none"> <li>서비스 발견 엔진을 이용하여 클라이언트가 원하는 서비스를 UDDI에서 발견 패턴</li> <li>UDDI 비즈니스 등록기는 여러 WSDL 서비스 기술서로 서비스등록</li> <li>서비스 서버는 보안과 관리기능</li> </ul>	

### 3.3 e-비즈니스 시스템 개발 프로세스

본 논문에서 제안하는 웹서비스 기반의 e-비즈니스 응용 개발 프로세스는 e-비즈니스 응용들의 원활한 상호작용을 위한 웹서비스를 가능케하여 웹서비스 장점을 최대한 활용할 수 있는 프로세스이다. 이 프로세스는 기존의 컴포넌트 기반의 e-비즈니스 응용 개발 프로세스에 웹서비스 개발을 위한 부분을 추가한다. (그림 6)와 같이 크게 분석, 식별, 설계, 구현, 실행의 5단계로 정의하고 좀더 세분화하여 9개의 단계로 나누어지며 21개의 활동들로 구성된다. 프로세스를 살펴보면 분석단계는 전체적인 비즈니스를 위한 요구사항 분석단계로 설계를 위한 전처리 단계이며 식별 단계에서는 비즈니스 도메인에서의 컴포넌트를 식별하고 설계단계는 서비스 설계와 메시지 설계로 나누며 구현 후 구축된 UDDI에 배포하고 최종적으로 배치 후 실행하게 된다.



(그림 6) e-비즈니스 시스템 개발 프로세스

#### 단계 1 : 비즈니스 도메인 분석

도메인 분석에는 시스템을 보다 효율적으로 개발하고 유지하기 위한 방법인 독특한 특성들을 인식하는 것에 기초한 방법으로 FODA(Feature Oriented Domain Analysis)를 사용한다. 비즈니스 분석은 비즈니스의 전체 조건인 규칙과 목적과 문제로 나누어서 표현하며 비즈니스 룰은 개념적인 제약사항과 조건으로 표현할 수 있으며 비즈니스 목적과 문제는 PFD(Process Flow Diagram)으로 표현된다. 식별된 비즈니스 객체들과 관계를 나타내주는 비즈니스 객체 다이어그램과 시스템 아키텍처 기반을 설계하는데 사용되는 비즈니스 다이어그램을 설계하게 된다.

#### 단계 2 : 비즈니스 요구사항 분석

웹서비스를 설계하기 위한 가장 초기의 작업이며 웹서비스 아키텍처 요청자 측면의 요구사항을 정의하고 명세한다. 이들 요구사항은 요청자 측면의 비즈니스 정보, 요청자 정보 등을 수집하여 정의하고 요구사항 도출과 분석 작업을

통해 얻어진 정보를 기반으로 유즈케이스 다이어그램을 표현한다.

#### 단계 3 : 컴포넌트 식별

특정 도메인에서 시스템 개발에 사용되어진 재사용 가능한 컴포넌트들의 집합을 식별한다. 식별된 컴포넌트는 다양한 의미로 여러 응용에 사용 가능하며 컴포넌트간의 조립을 통해 새로운 응용을 생성하는 자원이 된다.

#### 단계 4 : 컴포넌트 서비스 설계

컴포넌트 서비스 설계에서는 MDA 기술을 기반으로 개발 컴포넌트 정보들을 반영한 컴포넌트 기반 모델을 설계하고 개발하는 단계이다. 모델들은 컴포넌트의 특정 플랫폼에 독립적인 모델이며 이 모델들은 원하는 특정 플랫폼에 맞는 모델로 변환된다. OMG에서 제공하는 표준 모델로 MOF, CWM, UML을 각각 설계하고 각 응용들 간의 상호관계를 순차적으로 표현하는 시퀀스 다이어그램을 생성한다. 또한, 서비스의 클래스 다이어그램과 이전 단계에서 생성된 유즈케이스 다이어그램을 기반으로 인터페이스를 설계하고 설계패턴과 매칭되는 클래스 다이어그램을 도출하고 정련화한다.

#### 단계 5 : 컴포넌트 메시지 설계

컴포넌트 메시지 설계단계는 컴포넌트의 XML기반 웹서비스 표준 기술들을 설계하는 단계로써 SOAP를 사용하여 프로토콜을 구현하고 인터페이스 기술 언어인 WSDL로 기술하며 WSDL로 기술된 웹서비스 문서를 가지고 있는 서비스 디렉토리인 UDDI를 설계한다. 또한 요구사항에 따라 설계된 메시지 간의 매핑이 요구된다.

#### 단계 6 : 컴포넌트 모델링

컴포넌트 모델링 단계에서는 클래스와 초기 인터페이스를 통해 컴포넌트를 정의하고 컴포넌트간의 관계성, 컴포넌트 책임과 흐름을 정의한다. 정의된 컴포넌트 인터페이스를 정련화한다. 이 단계에서는 컴포넌트들을 식별하고 이들 컴포넌트간의 관계를 보여주는 컴포넌트 다이어그램을 생성한다.

#### 단계 7 : 웹서비스 컴포넌트 구현

웹서비스를 IBM의 WSTK, private UDDI Registry 1.0, AXIS, UDDI4J, WSDL4J, Microsoft의 .NET, WSDK, SOAP Toolkit, BEA의 WebLogic Workshop 의 등을 이용하여 컴포넌트를 구현한다.

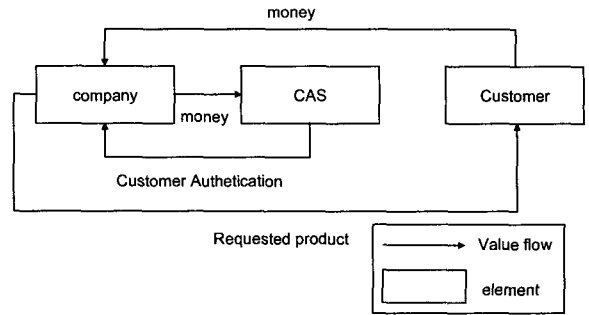
#### 단계 8 : 배 포

웹서비스의 검색과 배포를 위한 부분으로 서비스 요청자는 UDDI 서버의 UDDI 등록기를 통해 검색하고 서비스 제공자는 구현된 서비스 WSDL 문서들을 등록한다. 서비스 제공자가 UDDI에 등록을 하면 유일한 아이디를 비즈니스

또는 서비스 등록에 대해서 발행된다. UDDI 등록 데이터는 화이트, 옐로우, 그린 페이지들로 비즈니스 등록 정보를 분류해서 저장되며 그밖에도 서비스 타입들에 대한 프로그램머들, 비즈니스 등록자 정보 등을 수집한다.

단계 9 : 배치와 실행

웹서비스는 웹서비스 아키텍처의 동작에 참여하는 UDDI, 요구자 그리고 공급자들이 각 플랫폼에 배치되고 요청자는 UDDI의 원하는 서비스를 검색하여 발견하면 서비스를 제공자에게 요청하고 제공자로부터 서비스를 제공받게 된다.

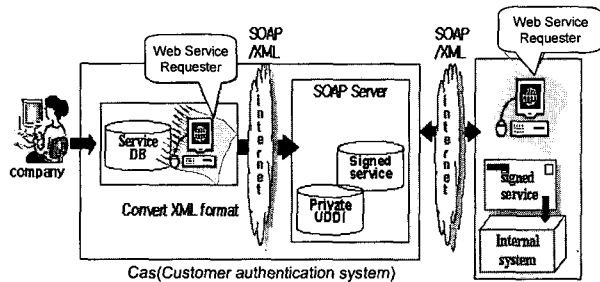


(그림 8) CAS 비즈니스 모델

4. 사례 연구

4.1 CAS 시스템 구조

개발 시스템 CAS(Customer Authentication system)는 전자 상거래에 있어서 필수적인 사용자 정보 인증을 담당하는 시스템이다. CAS 시스템의 전체적인 구조도는 (그림 7)과 같다. 제품회사의 요청에 따라 CAS 시스템이 public UDDI로부터 서비스를 받아서 XML 문서로 변환한뒤 SOAP을 통해 SOAP 서버에 전달한다. 전자서명이 된 서비스가 고객 서버에 전달되고 고객은 전자서명된 서비스를 DB에 자동 저장된다.

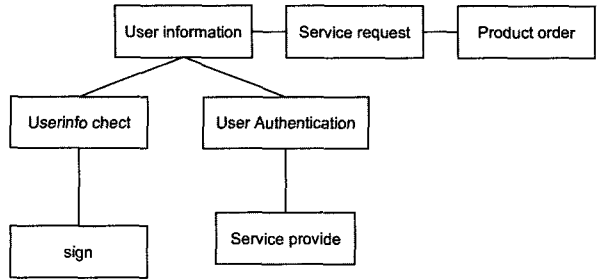


(그림 7) CAS 시스템의 전체 구조도

4.2 CAS 시스템 설계

단계 1 : 비즈니스 도메인 분석

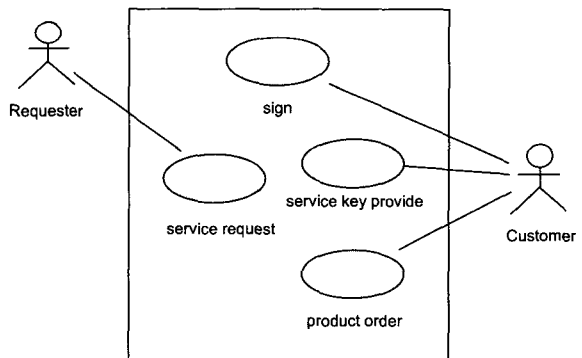
첫 번째 단계로 비즈니스 도메인 분석은 이 CAS 시스템의 비즈니스 전체조건인 규칙과 목적과 문제를 분석하여 CAS 시스템의 비즈니스모델을 (그림 8)에서와 같이 표현한다. 고객은 필요한 제품의 회사에게 제품을 요구하면 제품회사는 CAS 시스템에게 제품을 요청한 고객의 신상 데이터를 기반으로 CAS 시스템은 고객의 신원 인증을 한다. 제품회사는 이에 따른 가치를 지불하게 된다. 다음 산출물로 식별된 객체들 간의 관계성을 보여주는 CAS 시스템의 비즈니스 객체 모델을 (그림 9)에서 표현한다. CAS 시스템에서 식별되는 객체들로 고객 정보, 서비스 요청, 제품 주문, 고객 정보 체크, 사인, 고객 인증, 서비스 제공이 있고 이 객체들의 관계를 보여준다.



(그림 9) CAS 시스템의 비즈니스 객체 모델

단계 2 : 비즈니스 요구사항 분석

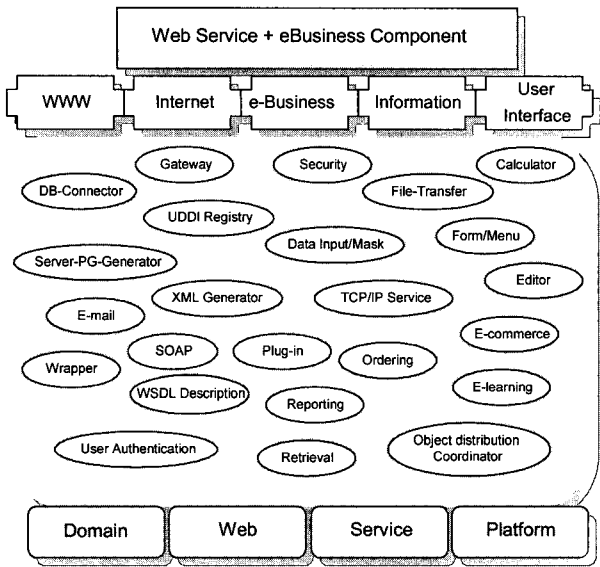
웹서비스 응용을 설계하기 위한 밑바탕으로 웹서비스 요청자 측면의 비즈니스 정보, 요청자 정보등을 수집하여 정의하고 이를 기반으로 유즈케이스 모델 (그림 10)을 정의하였다.



(그림 10) CAS 유즈케이스 모델

단계 3 : 컴포넌트 식별

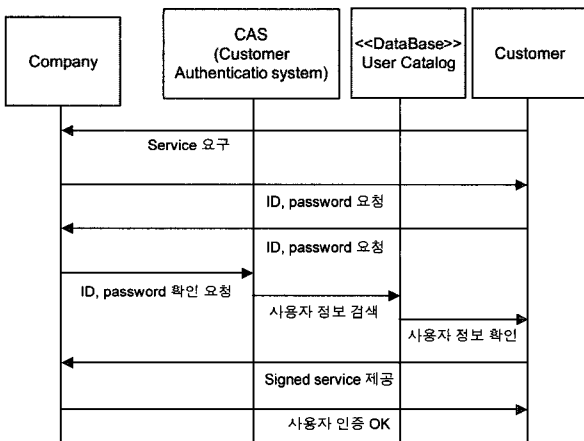
웹서비스와 e-비즈니스 영역에서 시스템 개발에 사용되는 재사용 가능한 컴포넌트들의 집합을 식별한다. 이들 컴포넌트는 웹서비스 기반 응용 개발을 위해 계층별로 커스터마이징하여 조립하고 확장가능하다. (그림 11)은 현재 인터넷 상에서 유통되고 있는 컴포넌트 기반 웹서비스 지원 e-비즈니스 도메인 영역에서 식별된 컴포넌트들을 나타낸다.



(그림 11) e-비즈니스 영역 컴포넌트

단계 4 : 컴포넌트 서비스 설계

비즈니스 서비스 버스 설계와 서비스 설계 활동 후 시퀀스 모델 (그림 12)을 산출한다. 사용자가 특정 회사의 제품을 주문하면 그 회사는 먼저 CAS에 소비자 인증을 요구한다. 소비자의 유일한 ID를 이용하여 CAS의 private UDDI에 있는 소비자의 정보 즉 서비스를 검색하여 일치하면 제품회사에 소비자의 사인된 정보를 보내준다. 인증된 소비자 정보를 바탕으로 전자상거래가 이루어지게 된다.

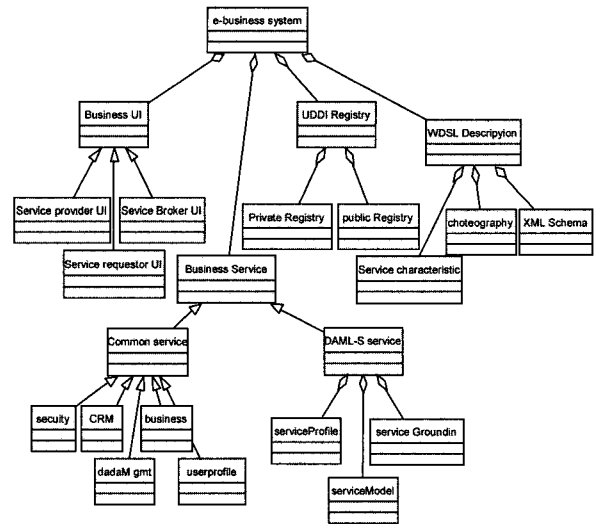


(그림 12) CAS 시스템의 순차 다이어그램

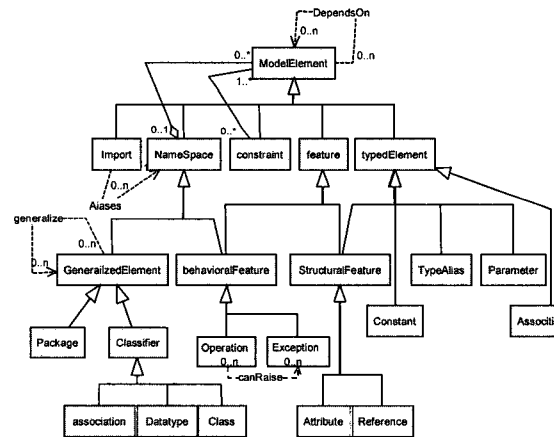
(그림 13)은 e-비즈니스 시스템 영역에서의 CAS 시스템에 적용되는 UML 메타 모델인 개략적인 클래스 다이어그램이다. 클래스 다이어그램은 e-비즈니스 시스템의 일련의 클래스, 인터페이스 및 협동과 그들 간의 관계를 보여준다.

(그림 14)는 e-비즈니스 시스템의 객체 지향 모델을 구성하기 위해 사용되는 메타모델들의 요소들과 구분, 구조를 정의한 MOF 모델을 보여준다. (그림 15)는 e-비즈니스 시

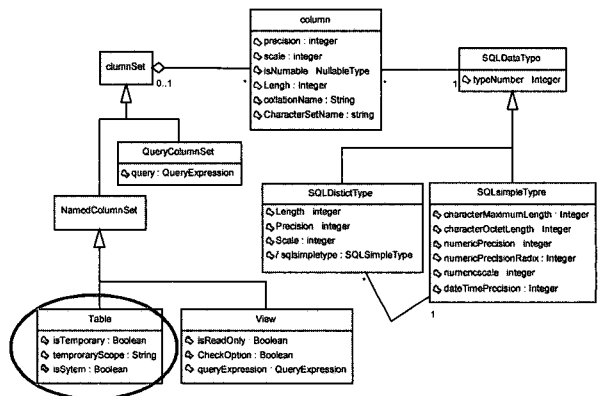
스템 도메인에서 사용되어지는 데이터 웨어하우징의 기초가 되는 비즈니스와 기술적인 메타데이터를 표현한 메타모델이며 이 메타모델을 기반으로 CAS 시스템에 적용한 일부 데이터 테이블 부분을 (그림 16)에서 보여준다.



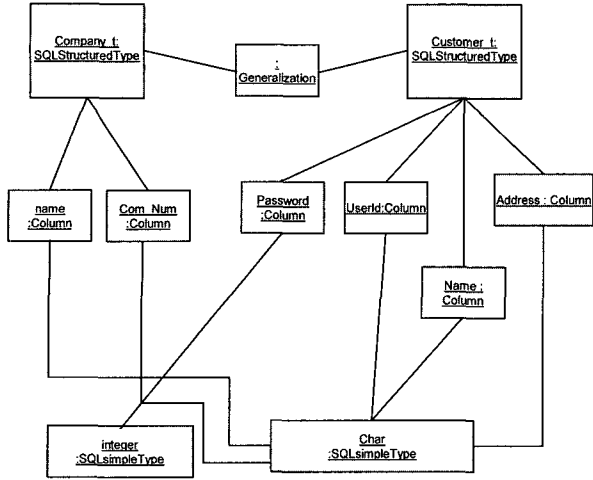
(그림 13) e-비즈니스 시스템의 개략적인 클래스 다이어그램



(그림 14) e-비즈니스 영역의 MOF 모델



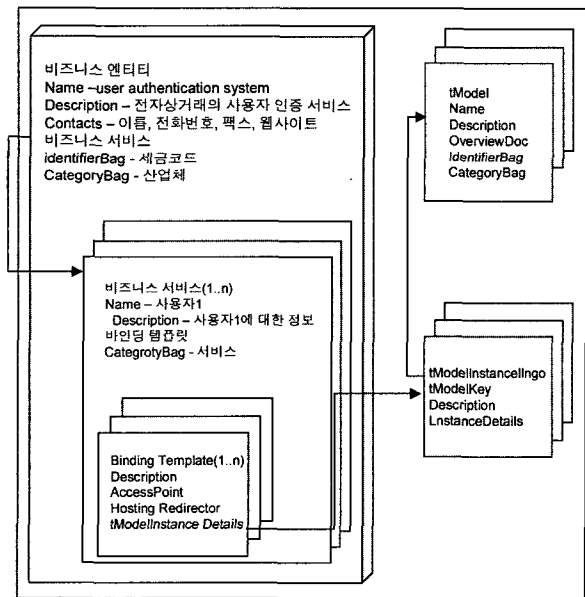
(그림 15) e-비즈니스 영역의 CWM 모델(개략적인 DB모델)



(그림 16) CAS 시스템에 적용된 CWM 모델(테이블 일부)

단계 5 : 컴포넌트 메시지 설계

SOAP 프로토콜을 설계하고 WSDL표준을 이용하여 서비스를 기술한다. WSDL은 서비스 인터페이스의 작성, 매개 변수 설정, SOAP 바인딩과 관련된 내용 기술, 구현 파일 작성 순으로 UDDI에 등록된 파일을 생성한다. UDDI 레지스트리는 화이트, 옐로우, 그린 페이지로 구성되어 있으며 CAS의 단일 UDDI 레코드를 (그림 17)에서 보여준다.

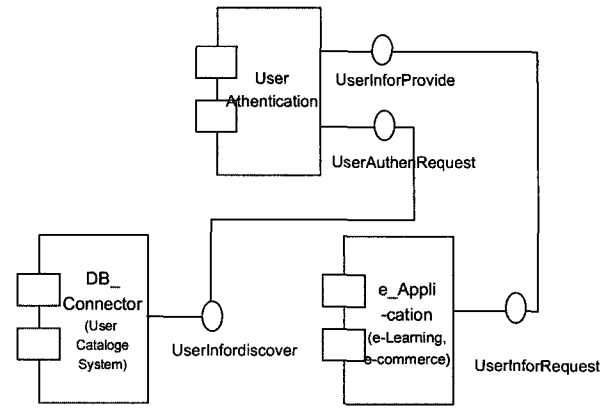


(그림 17) CAS UDDI 단일 레코드

단계 6 : 컴포넌트 모델링

e-비즈니스 도메인에서 식별된 컴포넌트들 간의 관계성 및 인터페이스, 컴포넌트 책임과 흐름을 정의한다. CAS 시스템의 컴포넌트를 식별하고 이들 관계를 보여주는 컴포넌트 모델을 설계한다. (그림 18)는 CAS시스템과 사용자 카탈로그 시스템, 기존 사이트 컴포넌트들 간의 관계를 보여

주는 컴포넌트 다이어그램이다.

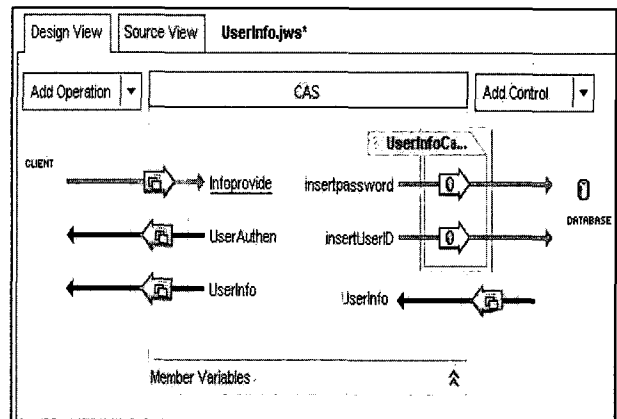


(그림 18) CAS 시스템의 컴포넌트 다이어그램

단계 7 : 웹서비스 컴포넌트 구현

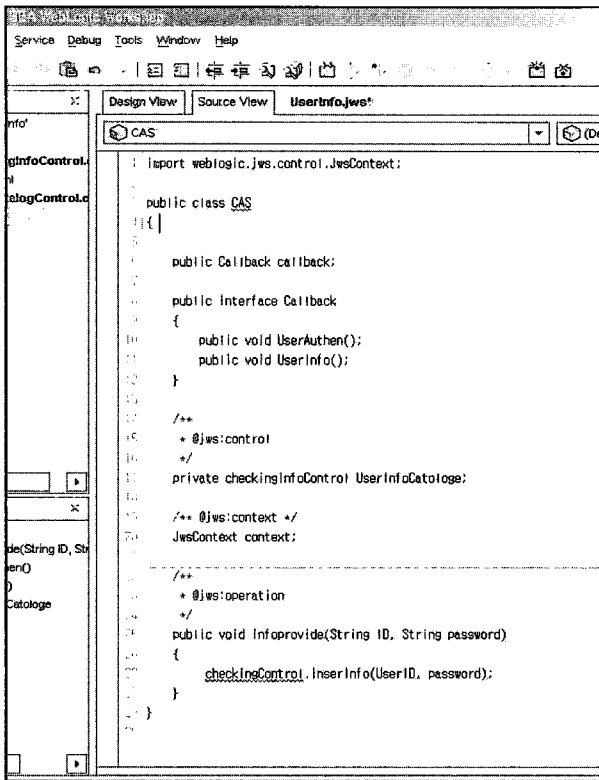
CAS 시스템을 BEA의 WebLogic WorkShop 7.0을 이용하여 웹서비스 기술을이용한 구현화면은 다음과 같다. (그림 19)은 WebLogic WorkShop 7.0에서의 개발자가 이해하기 쉽도록 제공해주는 사용자 인터페이스를 보여주고 (그림 20)은 사용자 인터페이스 구성에 따라 자동으로 생성된 코드를 보여준다.

설계과정에서 설계된 WSDL에 따라 Java로 작성된 파일을 WSDL 파일로 자동으로 바꾸어주는 WSDL 자동화틀인 Java2WSDL을 사용하여 WSDL 파일(그림 21)을 생성한다. 사용자 카탈로그 시스템은 고객의 정보를 저장하고 있는 데이터베이스를 총괄 관리하는 시스템이다. 이 데이터 베이스에는 고객의 이름, 고유의 ID, 주소, 비밀번호가 들어간다. 사용자 카탈로그 시스템에 의해 관리되어지는 사용자 정보인 데이터베이스를 생성하고 이렇게 생성된 데이터 베이스에서 필요한 사용자 정보를 SQL문을 이용하여 추출한다. 이들 데이터베이스 생성 코드와 SQL문은 컴포넌트 서비스 설계 단계에서 생성된 CWM 모델을 바탕으로 구현된다.

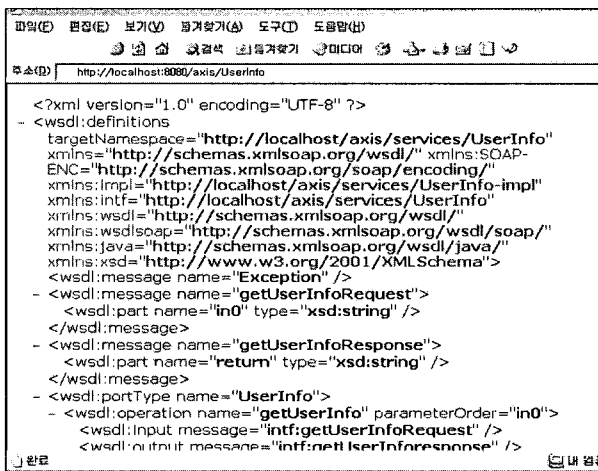


(그림 19) WebLogic WorkShop 7.0 구현 인터페이스





(그림 20) WebLogic WorkShop 7.0 구현 코드



(그림 21) WSDL 파일

### 5. 결론 및 향후 연구

인터넷과 웹은 빠른 성장과 함께 물리적 시장을 보완하여 기존의 전통적인 비즈니스 모델에서 제품과 서비스를 전달하는 새로운 방법을 형성하는 동적인 e-비즈니스로의 혁신적인 변화를 가져왔다. 이에 최근 급부상하고 있는 XML기반의 표준 기술을 바탕으로 응용간의 자동화를 가능하게 하는 웹서비스가 등장하였고 이 웹서비스 기술은 e-비즈니스 도메인에서 기업 내 또는 기업간의 통합 서비스 문제를

해결해주기 위한 기술 핵심이 되고 있다. 따라서 e-비즈니스 시스템은 소비자와 제공자 측면에서의 자동화를 위한 플랫폼에 웹서비스가 제공되는 응용이라 할 수 있다.

현재 인터넷 상에서 유통되고 있는 e-비즈니스 도메인의 컴포넌트들 간의 상호운영성과 통합성을 지원하는 MDA 기술은 모든 컴포넌트 기술요소의 표준 메타 모델을 정의하고 이를 기반으로 각 구성요소를 정의함으로써 모든 컴포넌트 기술 요소들의 호환성 및 시스템 간 동작성을 보장하고 있다.

본 논문에서는 특정 언어와 미들웨어에 종립적인 OMG의 MDA 기술을 적용하고 공통적인 기술적 접근에 따라 웹서비스 기반으로 한 e-비즈니스 시스템의 아키텍처를 보여주고 빠르게 변하는 IT 경향에 잘 적용할 수 있도록 e-비즈니스 도메인에 적용된 웹서비스 설계패턴을 식별하고 명세하였다. 또한, e-비즈니스 응용의 컴포넌트들을 개발하기 위한 개발 프로세스를 단계별로 제안하였고 이 제한된 프로세스에 기반하여 CAS 시스템 구축에 적용하였다.

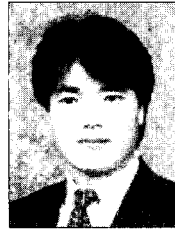
재사용가능하고 응용 도메인의 한계점을 극복하여 제안된 아키텍처와 식별된 설계패턴은 특정 도메인 개발자들을 지원하며 제안 개발 프로세스는 다양한 플랫폼과 도메인에서 개발된 응용들의 개발 시간과 노력을 최소화 시켜주고 양질의 시스템 구축이 가능하다.

향후 연구로 식별된 설계패턴을 기반한 e-비즈니스 컴포넌트 개발과 제안된 아키텍처의 상세화가 필요하며 개발된 컴포넌트 조합에 대한 연구와 제안된 개발 프로세스의 검증 등을 통한 표준화가 요구된다.

### 참고 문헌

- [1] Harvey M. Deitel, Paul J. Deitel, TemNieto and T. R. Nieto, e-Business & e-Commerce, Prentice Hall, 2001.
- [2] Reash Trivedi, "Web Service Architecture Models," RCG IT, Apr., 2002.
- [3] Michael Champion, "Web Services Architecture," at URL : <http://dev.w3.org/cvsweb/~checkout~/2002/ws/arch/wsa/wd-wsa-arch.html>, 2002.
- [4] 차정은, "컴포넌트 기반 개발 프로세스 지원을 위한 컴포넌트 저장소의 설계 및 구현," 대구가톨릭대학교 전산통계학 전공 박사학위논문, Feb., 2001.
- [5] 조완수, UML 객체 지향 분석·설계, 홍릉과학출판사, Apr., 2000.
- [6] Alan Brown, "Using service-oriented architecture and component-based development to build web service applications," Rational Software white pater from IBM, Apr., 2002.

- [7] Simeon Simeonove, 자바를 이용한 웹서비스 구축(XML, SOAP, WSDL, UDDI의 이해), 인포북, pp.32-48, 2002.
- [8] 프리먼, 존스, .NET XML 웹서비스, 정보문화사, Mar., 2003.
- [9] Richard Soley and the OMG Staff Strategy Group, "Model Driven Architecture," OMG Whit Paper Draft 3.2, at URL : <http://www.omg.org/~soley/mda.html>, Nov., 2000.
- [10] John D. Poole, "Model Driven Architecture : Vision, Standards and Emerging Technologies," European Conference on Object-Oriented Programming, at URL : [http://www.omg.org/mda/mda\\_files/Model-Driven\\_Architecture.pdf](http://www.omg.org/mda/mda_files/Model-Driven_Architecture.pdf), Apr., 2001.
- [11] Erich Gamma, Design Patterns : Abstraction and Reuse of Object-Oriented Design, European Conference on Object-Oriented Programming, 1993.



**김 행 곤**

e-mail : [hangkon@cu.ac.kr](mailto:hangkon@cu.ac.kr)

1985년 중앙대학교 전자계산학과(공학사)

1987년 중앙대학교 대학원 전자계산학과  
(공학석사)

1991년 중앙대학교 대학원 전자계산학과  
(공학박사)

1978년~1979년 미 항공우주국 객원 연구원

1987년~1989년 한국전기통신공사 전임연구원

1988년~1989년 AT&T 객원 연구원

2001년 Central Michigan University 방문교수

1990년~현재 대구가톨릭대학교 컴퓨터정보통신공학부 교수

관심분야 : CBSE, 소프트웨어 재공학, CASE, 유지보수 자동화  
툴, 사용자 인터페이스, 요구공학 및 도메인 공학