분산 컴포넌트 명세를 통한 e-비즈니스 컴포넌트 구축

김 행 곤[†] · 최 하 정[††] · 한 은 주[†††]

요 약

오늘날의 컴퓨팅 시스템은 인터넷을 사용하여 비즈니스 거래와 분산 업무 처리로 확대되어가고 있으며 정보 기술은 점차적 으로 재사용성과 독립성 그리고 이식성을 가진 컴포넌트를 기반으로 한 응용 개발이 확산되고 있다. 컴포넌트 개발 형태는 코드의 재사용이나 클래스 라이브러리보다 좀 더 발전된 형태의 부품개발 형태로서, CBD(Component Based Development)를 기초로 한다. 그러나, CBD를 이용하여 새로운 컴포넌트를 구축하는 비용의 증가와 함께 비즈니스 요구사항에 맞는 컴포넌트 개발을 위한 노력이 필요하다. 또한 빠르고 정확한 컴포넌트 정보를 웹 상에서 지원할 수 있도록 시스템 측면에서 정규화 형태의 컴포넌트 모델이 요구되고 있다. 본 논문에서는 사용자의 요구사항에 접근하고 웹 상에서 빠르고 신속하게 어플리케이션이 개발되는데 목적을 두고 있다. 네트워크상에서 비즈니스 도메인을 기반한 가장 소규모 단위의 분산 컴포넌트를 대상으로 인터페이스 명세를 제공한다. 컴포넌트 내부와 외부 관계를 담고 있는 명세는 사용자의 요구 사항을 정확하게 분석되도록 구성하며 이러한 명세는 비즈니스 도메인에서 재사용 가능한 정보 크기인 EJB(EnterpriseJavaBean)로 서블릿 시스템 내에서 세션과 엔티티 형태의 정보로 나누어 저장된다. 비즈니스 컴포넌트를 제공하기 위한 질의를 사용하여 비즈니스 컴포넌트를 이용할 수 있으며, 시스템은 차후에 등록, 자동 재배치, 조회, 테스트, 그리고 다운로드하여 컴포넌트를 제공받을 수 있는 환경 구축을 목표하며 이는 컴포넌트 재사용성을 증대시키며 비용을 절감하고 사용자가 분산 컴포넌트를 쉽게 사용할 수 있도록 하는데 목적을 둔다.

The e-Business Component Construction based on Distributed Component Specification

Haeng-Kon Kim[†] · Ha-Jung Choi[††] · Eun-Ju Han[†††]

ABSTRACT

The computing systems of today expanded business trade and distributed business process Internet. More and more systems are developed from components with exactly reusability, independency, and portability. Component based development is focused on advanced concepts rater than passive manipulation or source code in class library. The primary component construction in CBD. However, lead to an additional cost for reconstructing the new component with CBD model. It also difficult to serve component information with rapidly and exactly, which normalization model are not established, frequency user logging in Web caused overload. A lot of difficult issues and aspects of Component Based Development have to be investigated to develop good component-based products. There is no established normalization model which will guarantee a proper treatment of components. This paper elaborates on some of those aspects of web application to adapt user requirement with exactly and rapidly. Distributed components in this paper are used in the most tiny size on network and suggest the network-addressable interface based on business domain. We also discuss the internal and external specifications for grasping component internal and external relations of user requirements to be analyzed. The specifications are stored on Servlets after dividing the information between session and entity as an EJB (Enterprise JavaBeans) that are reusable unit size in business domain. The reusable units are used in business component through query to get business component. As a major contribution, we propose a systems model for registration, auto-arrange, search, test, and download component, which covers component reusability and component customization.

키워드 : 컴포넌트 기반 개발(Component Based Development), 분산 컴포넌트(Distributed Component), e-비즈니스 컴포넌트(e-Business Component), 엔터프라이즈 자바빈즈(EJB), 서블릿(Servlet)

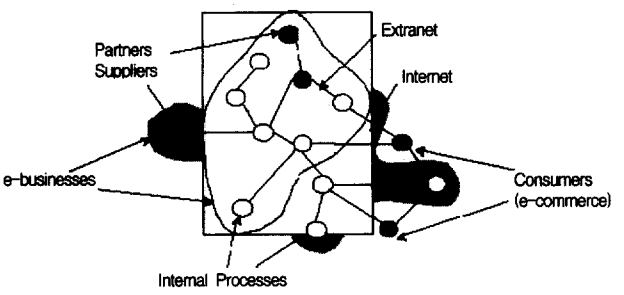
1. 서 론

웹 사용자들의 요구와 최근의 사용자들 요구의 많은 변화가 있으며 점차 기술과 응용은 B2B의 기술로 변화하였다. 이것은 비즈니스 측면에서의 전자상거래와 더불어 인터넷에서의 제공된 정보와 서비스를 확대하여 고객에게 많은 기회를 부여하도록 한 것이다(그림 1)[1]. 현재의 웹 기술은 사용자의 접속에 따른 과부하와 성능저하로 정보 관리 및 애플리케이션에 많은 제약을 초래할 수 있으므로, e-비즈니스 측면의 정보를 다루는 형태가 필요로 한다.

본 논문에서는 사용자가 요구하는 웹 애플리케이션이 정

† 종신회원 : 대구가톨릭대학교 컴퓨터정보통신공학부 교수
†† 준 회 원 : 대구가톨릭대학교 대학원 전산통계학과
††† 준 회 원 : 경일대학교 컴퓨터공학과 초빙교수
논문접수 : 2001년 10월 5일, 심사완료 : 2001년 12월 19일

확하고 빠르게 구축될 수 있도록 비즈니스 모델을 기반으로 컴포넌트를 사용자 입장에서 분류할 수 있도록 한다. 이를 위해 네트워크 상에서 접근 가능한 인터페이스를 제공하는 가장 작은 단위의 분 컴포넌트의 요소들을 분류하고 이를 비즈니스 컴포넌트들로 구성하기 위해 분산 컴포넌트 명세 개발 프로세서를 제안한다. 이는 각각의 역할과 기능들을 정확하게 분석/설계 명세로 정의하고 여기서 얻어진 명세를 기반으로 최종의 비즈니스 컴포넌트를 목표로 한다.



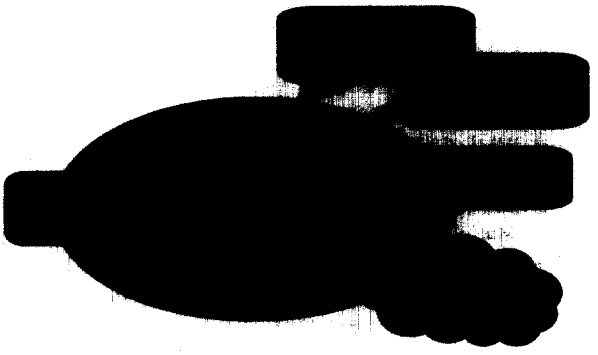
(그림 1) e-비즈니스의 환경

서버 등록을 위해 일반적으로 재사용 가능한 정보형태인 EJB형식의 컴포넌트 명세를 작성하며 연속적인 사용자와의 인터페이스를 위해 서버내의 애플릿인 서블릿을 구축하여 트랜잭션 관리 및 재사용성을 증대시킬 수 있도록 한다. 또한 사용자 요구에 따라 요구 명세를 통해 부품에 대해 질의를 작성하고 결과를 빠르고 쉽게 얻을 수 있는 인터페이스를 제공한다. 이는 기존의 컴포넌트 접근을 용이하게 할 뿐만 아니라, 인터넷 기반의 비즈니스 업무 처리 서비스를 폭넓게 하고 효율적으로 관리함으로써 과다한 정보 제공을 위한 시간적, 공간적 비용을 줄이고 일시적이고 컴포넌트 사용자들이 빠르고 효과적으로 이들 컴포넌트를 활용할 수 있도록 한다.

2. 관련 연구

2.1 기존 컴포넌트 명세

컴포넌트 명세는 공급자_중계자_고객간의 약속을 명시하는 것으로 약정 기반 기법, 모델링 언어를 사용하여 시각적으로 표현하는 모델링 기법 그리고 수학적 개념으로 시스템을 명확하게 표현 정형 명세 기법 등으로 나눈다[3, 12, 14]. 약정기반은 (그림 2)에서와 같이 기본, 행위, 동기화 서비스등으로 분류해서 명세화 하며 기본약정에서는 컴포넌트 개발자가 개발언어, IDL등으로 오퍼레이션, I/O 인자, 그리고 예외 사항등을 명시하고 행위약정에서는 결과를 예측하기 위해 선, 후, 불변 조건을 사용한 검증으로 오퍼레이션 행위 검증과 일관성 있는 조건의 명시 그리고 클라이언트와 서버에게 약정의 책임 부가 서비스를 명세화 한다. 동기화 약정에서는 뷰잉/메소드 사이의 동기화를 고려한 객체 행위를 기술하며 컴포넌트가 제공하는 순차성, 병행성, 혼잡성 관계 기술행위들을 정량화 한다. 서비스 약정에서는 서버 객체의 준수 성질 나열함으로써 서비스 질 약정을 정적(때로는 동적인 협상)으로 명시서비스 질의 인자 즉, 최대 응답 시간, 자료 스트림을 사용한 결과 처리량 그리고 서비스 질 명세 등을 추가한다[18].

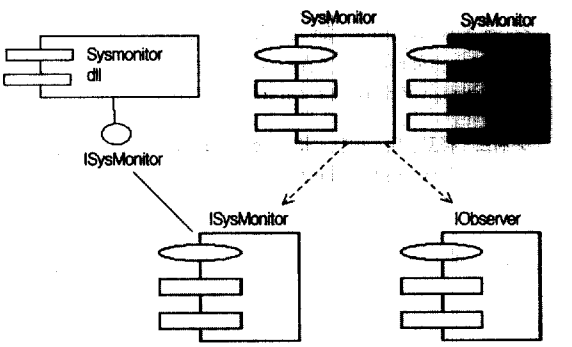


(그림 2) 약정 기반에 의한 컴포넌트 명세 분류

interfaces	: F Interface
iids	: F IID
firstInterface, identity	Interface
{ firstinterface, identity } ⊆ I interfaces ∀i : interfaces; d:IID | (i, d) I dom QI° QI(i, d) ⊆ I interface iids = IIDsOfInterface { interface }	

(그림 3) 정형 명세에 의한 COM 프레임워크 컴포넌트 명세

또한 컴포넌트 기능의 명료화를 위한 정형 명세 기법은 이해와 작성이 어렵지만 컴포넌트 프레임워크나 모델링 언어에서의 모호성을 제거할 수 있는 방법이다(그림 3). 기존의 UML을 이용한 명세는 패키지의 스테레오 타입으로 컴포넌트 정의하며 UML에 의한 컴포넌트 명세는 기존 모델링 기술에 스테레오 타입(컴포넌트 타입), 인터페이스 클래스, 실현관계를 추가하고 이를 이용하여 컴포넌트를 기술하게 된다(그림 4)[17].



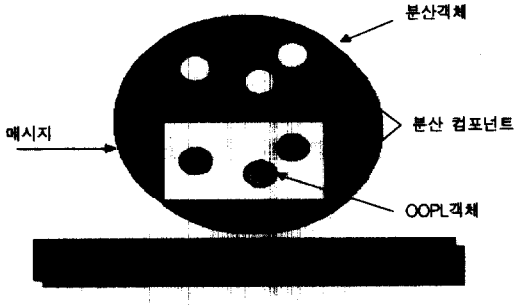
(그림 4) UML에서의 컴포넌트 다이어그램 명세

2.2 분산 컴포넌트(Distributed Component : DC)

분산 컴포넌트는 네트워크 접근 가능한 인터페이스를 제

공하는 가장 작은 단위의 컴포넌트로서 상업적으로 유용한 컴포넌트 구현 기술을 사용하여 생성되는 설계패턴의 응용 분야로, 전 개발단계동안 사용된다.

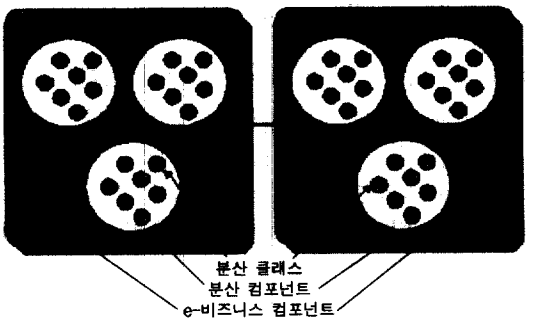
e-비즈니스 컴포넌트 계층은 분산 컴포넌트로 구현되어 질 수 있으며, 독립적으로 개발되어 수행 가능한 모듈이 된다. 이러한 분산 컴포넌트는 직접 e-비즈니스 컴포넌트 계층의 독립적인 개발을 제공하고 가능케 하며, 개발과 수행 시간에서는 결국 e-비즈니스 컴포넌트 그 자체가 된다. 스레드 관리와 통신 프로그래밍, 그리고 병행 관리 등과 같은 복잡한 수행에 대해서는 은닉되어지며, 실행시간동안 ORB (Object Request Broker), 운영체제 등과 같은 소프트웨어를 기반으로 기술적인 분산 컴포넌트 독립성을 이루게 되므로, 코드에 영향 없이 개발 가능케 한다[20].



(그림 5) 분산 컴포넌트

이러한 분산 컴포넌트 접근을 통해 재구축 또는 변경 없이 또 다른 컴포넌트를 서브 클래스 함으로써 상세화 할 수 있으며, 다수의 프로그래밍 가능한 언어들 중 어느 것으로 작성하여도 무방한 중간적인 응용 코드로 표현된다. 또한, 기술적인 바인딩을 통해 다른 플랫폼으로 이식이 이루어지며, 메시지내의 메타 데이터를 사용하여 클라이언트의 변화 가능한 요구사항과는 독립적으로 인터페이스가 개발되어지므로 최소화된 의존성을 가진다[2].

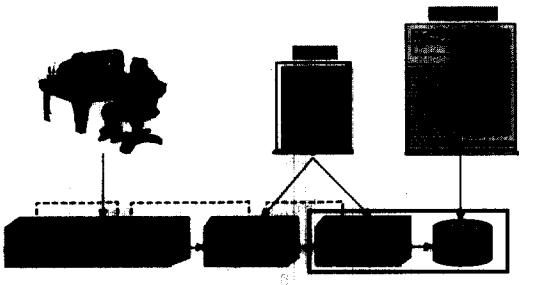
(그림 5)에서는 분산 컴포넌트로 조합된 집합은 결국 e-비즈니스 컴포넌트인 분산 객체의 구현을 의미함을 나타낸다. 분산 객체들간의 메시지는 메시지 파라미터들에 대한 메타 데이터로 운반되어지며 각 분산 객체들은 메시지를 번역하고 설계 때 정의된 인터페이스로 접근되어진다.



(그림 6) e-비즈니스와 분산 컴포넌트와의 관계

(그림 6)에서는 분산 컴포넌트와 e-비즈니스 컴포넌트간의 관계를 나타내고 있다. 분산 컴포넌트는 분산 객체로 이루어지며 여러 단계에서의 비즈니스 개념들을 자동적으로 가지며 단일 단계에서의 e-비즈니스 컴포넌트 상에서 분산 객체를 구현하는 기술적인 컴포넌트라 할 수 있다. 반면에 e-비즈니스 컴포넌트는 다수의 분산 클래스로 구성되며 각각 역할을 가지고 있고, 큰 입자로 된 비즈니스 개념의 패키지 형태로 개발된 것이다. 분산 컴포넌트의 4가지 카테고리는 (그림 7)과 같이 구분된다[3].

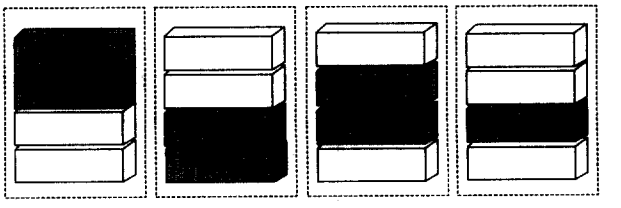
- UDC(User Distributed Component) : 비즈니스 컴포넌트를 위한 사용자 인터페이스를 표현하기 위해 특별히 생성되는 분산 컴포넌트들로 윈도우 크기와 특징 등의 정보를 제공한다.
- WDC(Workspace DC) : 단일 사용자와 관련 있는 응용 논리의 필요성에 의해 제공되는 분산 컴포넌트로, 실제 컴포넌트의 작업공간을 이룬다.
- EDC(Enterprise DC) : 비즈니스 컴포넌트들간의 상호동작과 검증, 비즈니스 규칙을 적용하는 계층으로 데이터 무결성을 관리한다.
- RDC(Resource DC) : 데이터베이스 액세스를 제공하는 실제 물리적인 처리부분의 컴포넌트 저장소이다



(그림 7) 분산 컴포넌트의 4가지 카테고리

이러한 분산 컴포넌트들은 여러 가지 조합방법에 따라서 두 개 혹은 그이상의 분산 컴포넌트들끼리 묶여서 사용자의 필요에 따라서 e-비즈니스 컴포넌트로 사용되어진다. (그림 8)과 같이 4가지 카테고리는 서로 연결되어 e-비즈니스 컴포넌트로 사용되어진다.

따라서, 비즈니스 도메인의 컴포넌트 측면에서 가장 기본적이면서 모든 비즈니스 컴포넌트의 주요 요소인 분산 컴



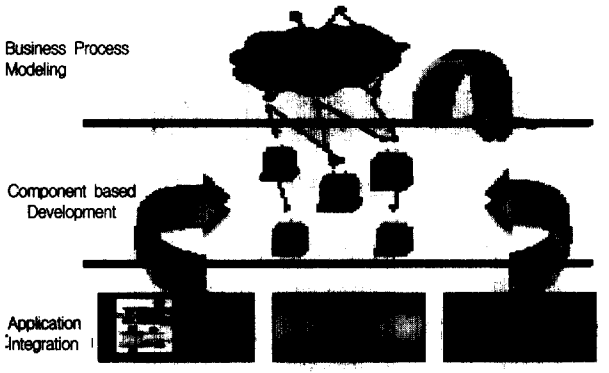
(그림 8) 다양하게 결합된 e-비즈니스 컴포넌트들

포넌트를 정확히 명세함으로써 명확하고 용이하게 비즈니스 컴포넌트 시스템을 구축할 수 있으며, 기존의 컴포넌트의 재사용성이 용이하고 비용절감과 인터넷 기반의 비즈니스 업무 처리 서비스를 폭넓게 확장시킬 수 있다[4].

2.3 CBD와 e-비즈니스와의 관계

CBD(Component Based Development)는 객체지향 패러다임에 근거하여 비즈니스 요구를 수용한 복합적인 소프트웨어 방법론으로 비즈니스 로직을 포함하는 생산성 있는 컴포넌트 개발과 이들 컴포넌트의 인터페이스를 통한 의미있는 결합으로써 새로운 소프트웨어를 개발한다는 개념이다. 아키텍처 기반의 컴포넌트 명세화와 구현 그리고 패키지, 이들 생산된 컴포넌트들의 재사용성 관리 및 배포, 조립에 의한 응용 생성에 이르는 체계적인 프로세스가 컴포넌트 저장소를 중심으로 병행되어져야 한다[6].

한편, e-비즈니스 컴포넌트를 구축하기 위해서 아키텍처는 재사용성과 기존의 소프트웨어 부산물들의 통합에 대한 프레임워크를 제공한다. 대부분의 성공적인 e-비즈니스 컴포넌트는 효율적으로 비즈니스 변화에 빠르게 응답하는 형상과 기존 소프트웨어 가용성으로 제시한다. 아키텍처는 동적으로 움직이며 비즈니스 프로세스 모델링을 통해 얻어진 비즈니스 측면의 로직과 기존에 운영하던 외부 컴포넌트와 기존의 설계, 데이터베이스, 그리고 ERP(Enterprise Resource Planning) 인터페이스를 통해서 컴포넌트로 전환함으로써 비즈니스 변화와 개혁을 가져다준다(그림 9)[7].

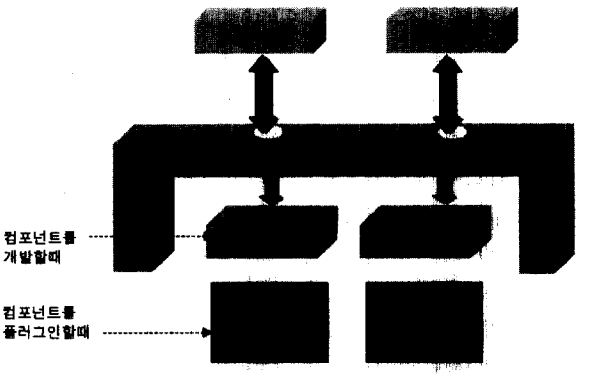


(그림 9) 통합된 e-비즈니스 컴포넌트 구조

2.4 EJB의 구동

EJB 애플리케이션은 클라이언트와 서버의 EJB 컴포넌트들로 이루어져 있다. EJB 컴포넌트들은 서버에서 동작하는 업무를 수행하며 관련 데이터를 처리하는 소프트웨어 모듈이다. 클라이언트들은 EJB 컴포넌트들을 호출하며 그 결과값을 받을 수 있다. 또한, 실행된 EJB 컴포넌트들은 EJB 컨테이너의 관리하에서 생성, 실행, 소멸과정이 수행된다.

(그림 10)과 같이 클라이언트들은 EJB 애플리케이션 서버를 통해 EJB 컴포넌트와 연결하여 메소드들을 호출할 수



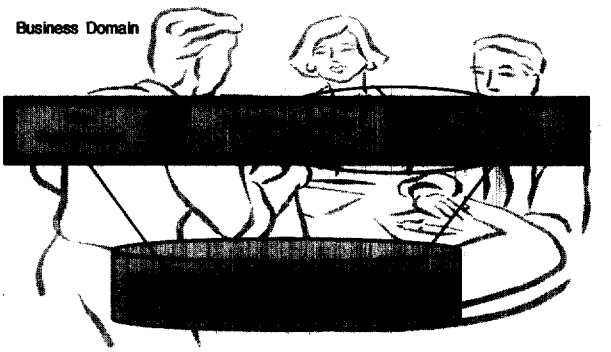
(그림 10) EJB 애플리케이션 서버의 구조

있다. EJB 컴포넌트들이 실행될 때 자동으로 관리되어야할 트랜잭션, 보안, 자원관리 등의 내용을 XML문서로 작성하여 EJB 애플리케이션 서버에 컴포넌트와 같이 플러그 인(Plug-in)한다[7].

3. 분산 컴포넌트 명세를 통한 e-비즈니스 컴포넌트 생성

3.1 개발 프로세스

분산 컴포넌트는 곧 분산 설계 패턴 컴포넌트로서 (그림 11)과 같은 분산 컴포넌트 명세 개발 프로세스를 통해 분석되어 최종 조합된 비즈니스단계의 컴포넌트로서 사용 가능하다. 명세 개발 프로세스 단계는 다음과 같이 3단계로 이루어져 있다.



(그림 11) 분산 컴포넌트 명세 개발 프로세스

Step 1 : 요구 명세 단계(요구사항 분석 단계) : 사용자형태를 잘 파악하여 그려내며, 주요 비즈니스 개념을 식별하여 최종 비즈니스 산물 및 서비스를 결정하기 용이하도록 요구 명세를 획득한다.

Step 2 : 분석 및 설계 명세 단계(분산 컴포넌트 이해 및 모델링 단계) : 요구 명세를 통해 사용자가 원하는 시스템을 조합할 수 있도록 기본 요소인 사용자, 워크 스페이스, 엔터프라이즈, 그리고 리소스

측면에서의 고려사항들을 분석하여 내부, 외부, 그리고 의존성 명세로 정확하고 상세하게 작성하여, 비즈니스 컴포넌트로 조합 가능한 명세를 제공한다. 내부에서는 컴포넌트의 실제 물리적인 부분에 대해 명세하며, 외부에서는 컴포넌트들간의 관계를 표현하는 인터페이스를, 그리고 의존성 명세에서는 컴포넌트들 내의 상호 관계에 대해 명세 한다. 이러한 명세 정보들은 통합된 데이터베이스를 통해 저장되어 조합 시 사용되어 질 수 있으며 반복적으로 과정이 수행된다. 분석을 통해 얻어진 명세들을 통해 분산 컴포넌트들간의 관계들을 시각적으로 나타내어 이들간의 상호동작을 쉽게 파악하고 분산 컴포넌트들이 비즈니스 논리에 맞게 조합되는데 도움을 준다. 객체 지향 어플리케이션 설계를 위한 표준 언어로서 인정되고 있는 UML (Unified Modeling Language) 표기법[8]을 사용하여 동적이고 다중적인 경로를 통해 구성 요소들간에 독립적이면서도 밀접한 상호 연관성을 유지할 수 있는 모델을 제시한다.

Step 3 : 구현 단계(조합 단계) : 구분된 각 분산 컴포넌트를 실제 조합하는 부분으로서, 비즈니스 컴포넌트들이 최종의 시스템에 잘 적용될 수 있도록 각각의 기능에 맞게 조합되어져야 한다. 비즈니스 컴포넌트는 컴포넌트의 역할과 재사용성을 고려하여 하나이상의 분산 컴포넌트들로 구성되며 결국 독립된 하나의 비즈니스 컴포넌트를 생성할 수 있다.

3.2 분산 컴포넌트 명세

분석/설계된 명세는 분산 컴포넌트의 각 역할을 잘 구분할 수 있도록 세 가지 측면에서 명세 되어진다. 내부명세는 화이트박스 형태로 실제 시스템에 대한 명세들이며, 외부명세는 컴포넌트들간의 관계를 나타내는 인터페이스 측면의 명세이고, 마지막으로 의존성 명세는 하나의 비즈니스측면 컴포넌트들내의 분산 컴포넌트들간의 관계와 다른 비즈니스측면 컴포넌트들과의 관계를 나타내는 명세이며 <표 1>에서 <표 7>로 나타낼 수 있다.

3.2.1 내부명세(Internal Specification)

작업 영역공간이 워크 스페이스와 사용자 인터페이스, 그리고 엔터프라이즈 및 리소스 분산 컴포넌트에 대한 명세를 할 수 있다. 먼저 작업영역 분산 컴포넌트 내부 명세에서는 워크 스페이스 사용자가 단일 사용자, 혹은 다중 사용자인가를 파악하여 사용자의 범위를 정확히 명세하였다<표 1>. 또한, 사용자 인터페이스에서는 사용자 측면에서의 실제 시스템 화면 구성부분으로서 어떠한 방식의 컴포넌트 방식을 사용하는가를 선택하여 정하였다<표 2>. 엔터프라이즈 분산 컴포넌트 명세에서는 실제 비즈니스 도메인측면에서 행해질 수 있는 모든 동작들을 명세화하는 부분으로서, 서버에서 동작할 수 있는 컴포넌트들의 상태를 나타내었다<표 3>. 또한 리소스 명세부분에서는 서버와 연관하여 컴포넌트들의 데이터베이스 내에서의 저장형식에 대해 명세하였다<표 4>.

〈표 1〉 사용자 인터페이스 분산 컴포넌트 내부 명세

기 능	세부기능	
display	User Interface Component	VBX
		JavaBeans
	component-based user interface	
	Pluggable user interface	pull
		push

〈표 2〉 작업영역 분산 컴포넌트 내부

기 능	세부기능
workspace	single-user mode
	multiple-user model

〈표 3〉 엔터프라이즈 분산 컴포넌트 내부 명세

기 능	세부기능	
state	Database	get
		put
	component accounting	
	introspection	

〈표 4〉 리소스 분산 컴포넌트 내부 명세

기 능	세부기능
Database model	single integrated DB
	multiple autonomous islands of data
	single integrated physical DB
Granularity	RDB
	OODB
primary key	
foreign key	logical foreign key
	physical foreign key
nomalization	
relationship	one to many
	many to many
access	Direct SQL access
	layered access

3.2.2 외부 명세(External Specification)

다른 분산 컴포넌트들간의 상호 관계를 위한 인터페이스 표현으로 최종 시스템의 특정 행위에 대한 형식적인 의존성을 가진 행위라고 할 수 있으며, 여러 가지 인터페이스 종류에 대한 선택을 할 수 있도록 하였다<표 5>, <표 6>.

〈표 5〉 작업영역 분산 컴포넌트 외부 명세

기 능	세부기능
interface categories	business entity life cycle
	collection management
	functional interface
	component lifecycle
	socket operation
	configurability
	introspection
	dispatch
time-type	run-time interface
	build-time interface
	design-time interface
naming conventions	
interface design patterns	one operation for each kind
	a group operation for single attribute
	a tagged data string as the operation parameter
	a mix of tagged data parameter and typed parameter
Business Data Type	dependent object
	network class

〈표 6〉 엔터프라이즈 분산 컴포넌트 외부 명세

기 능	세부기능
interface categories	business entity life cycle
	collection management
	functional interface
	component lifecycle
	socket operation
	configurability
	introspection
	dispatch
naming conventions	
time-type	run-time interface
	build-time interface
	design-time interface
interface design patterns	one operation for each kind
	a group operation for single attribute
	a tagged data string as the operation parameter
	a mix of tagged data parameter and typed parameter
Business Data Type	primary key inclusion
	consistency with database
	nesting minimization
	self-containment
	system wide consistency
	boundary values

3.2.3 의존성 명세(Dependency Specification)

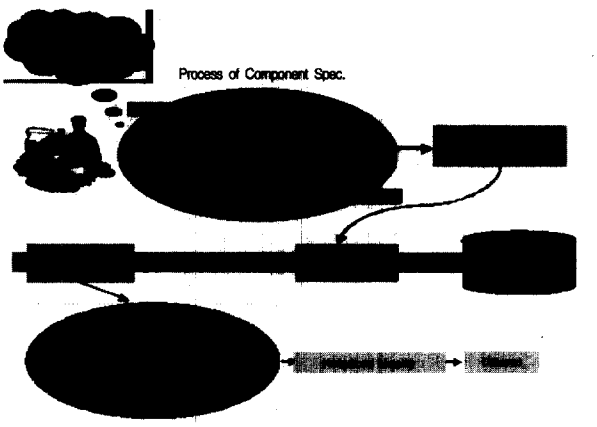
외부와 내부 명세에서 제시된 여러 가지 분산 컴포넌트들 간의 관계를 나타내는 것으로서 하나의 비즈니스컴포넌트내의 분산 컴포넌트들간의 관계와 다른 비즈니스 컴포넌트들 내의 분산 컴포넌트들과의 관계 등 그들간의 종속여부를 나타내었다<표 7>.

〈표 7〉 분산 컴포넌트 의존성 명세

기능
internal
external

3.3 시스템 아키텍처

사용자가 원하는 설계 패턴 중심의 컴포넌트를 등록 명세를 통해 등록하고 검색할 수 있도록 하여 과다한 정보제공을 위한 시간적, 공간적 비용을 줄이도록 한다. 또한 일시적이고 미숙한 재 사용자들이 빠르고 효과적으로 활용할 수 있도록 사용자 지향의 편리성을 제공할 수 있도록 한다. 이러한 웹 기반의 컴포넌트 개발 프로세스는 (그림 12)와 같이 제시하였다.



(그림12) 비즈니스 컴포넌트 개발 아키텍처

사용자 측면에서는 이러한 서버에 접속, 등록하여 자신이 원하는 컴포넌트를 분산 컴포넌트 명세 개발 프로세스를 통해 사용자가 원하는 컴포넌트로 명세하고, 이를 검색하여 정확하고 다양한 분산 설계 패턴 컴포넌트로 접근 가능하다. 이는 차후에 정보를 자신의 컴퓨터로 다운로드 할 수 있도록 검색 시스템으로 구축되어진다.

개발자측면에서는 서버에 접속 후 자신을 개발자로 등록하고 분산 컴포넌트 명세 개발 프로세스를 통해 만들어진 컴포넌트와 기존에 있는 여러 가지 컴포넌트를 EJB형식과 유사하게 맞추어 컴포넌트 저장소에 저장하는 통합 환경을 구축하도록 한다.

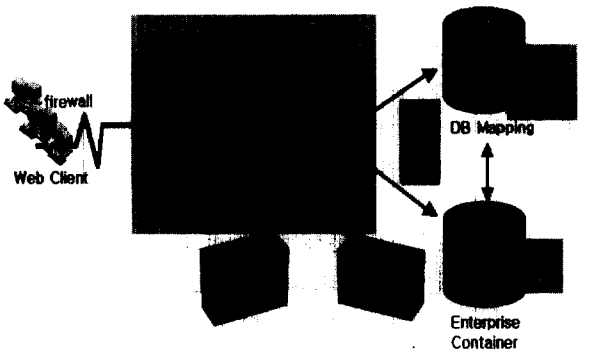
이들은 표준을 지향하는 개발을 통해 다른 시스템으로 포맷팅이나 확장이 쉽다. 이는 기존의 웹 애플리케이션 서버 형식인 CGI 방식에서의 문제점으로 잦은 프로세스의 부하와 분산과 서버의 자원 절약에 대한 미비를 보완하며 다양한 컴포넌트를 사용자가 이용할 수 있도록 한다.

기존의 웹서버와는 다른 데이터 처리를 위한 프로그램 층과 웹 서버층을 각기 독립적으로 관리 할 수 있는 서버를 두어 이를 통합 관리하고, 현존하는 객체 컴포넌트 기술 중에서 가장 현실적이고 보편화된 방안인 자바를 통해 보다 편리하고 표준화되도록 한다. 또한 트랜잭션 관리나 보안, 데이터베이스 커넥션 그리고 쓰레딩 등의 다양한 문제로부터 개발자를 분리시키기 위해서 EJB 형태의 명세 표준안을 이용한다.

웹 브라우저나 컴퓨터의 에러에 의한 트랜잭션 등의 손실을 방지하기 위해 JTA(Java Transaction API)/JTS(Java Transaction Service)를 통해 상태를 관리하며, 잦은 서버 접속에 의한 데이터 손실 및 접속 종료, 그리고 서버 플랫폼에 세션 정보를 유지시키고, 애플릿을 이용하여 클라이언트에 저장한다.

초기의 다운로드시 많은 시간 소요와 단순한 조회의 이용 시 서버에 크게 부하가 걸리는 점을 보완하기 위해 서블릿을 지원하고 프로세스의 잦은 생성과 소멸을 줄이기 위해 멀티

쓰레드 방식을 사용하며 서버 내에 CORBA 기술을 지원하도록 하여 부하 분산기능을 수행토록 한다. 본 논문의 구현 환경은 (그림 13)과 같다.



(그림 13) 컴포넌트 기반 웹 응용 관리 시스템

4. 사례 연구

4.1 요구사항 분석

인터넷에서 제공하는 많은 사이버 쇼핑몰에서 사용자가 전자상거래를 할 때 요구되어지는 실 예를 토대로 분석하였다. 이는 분석과 설계 명세에서 분산 컴포넌트로 분해되기 위한 초기단계로서, 사용자가 원하는 최종의 시스템을 구현하기 위해서는 상세한 분석이 필요하게 된다.

4.2 분석 및 설계 명세

분산 컴포넌트 개발 프로세스를 통해 다음과 같은 명세를 추출한다. 앞에서 제시한 명세 안을 토대로 웹 상에서 물품을 구입할 때의 상거래를 위한 사용자가 필요한 요구사항을 기반으로 분석/설계하였다<표 8>~<표 11>.

〈표 8〉 요구사항명세

고객 요구사항
사용자가 원하는 물건을 구매할 때 사용자의 웹 브라우저를 통해 사업자의 사이트에 접속하며, 물건을 구매하기 위해 신규등록을 하여 자신의 개인 정보를 서버에 저장시켜 놓는다. 또한 원하는 물건에 대한 정보를 얻고자 할 때 서버로부터 물건에 대한 정보를 얻을 수 있으며, 서버는 구매 정보를 전용 데이터베이스를 통해 탐색하여 사용자에게 정보를 전달해 준다. 구매 요구가 들어오면 서버는 계약된 납품업자에게 연락을 하여 사용자가 원하는 물품을 제작토록 연락하며, 납품업자는 물건에 대한 명세를 받아서, 제작하도록 한다.

〈표 9〉 사용자 인터페이스 명세

기능	세부기능		명 세
display	User Interface Component	JavaBeans	웹 상에서 사용자가 정보 획득, 검색
	browser		netscape

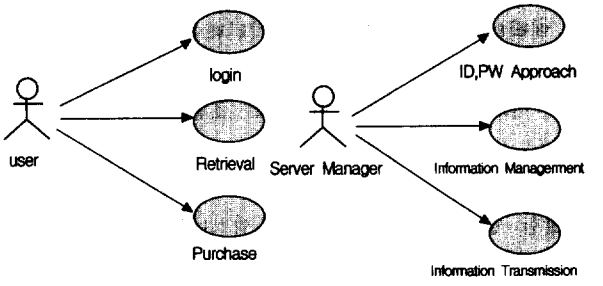
〈표 10〉 엔터프라이즈 명세

기능	세부기능		명 세
state	Database	get	데이터베이스로부터 정보 획득,저장
		put	
	component accounting		컴포넌트의 관리

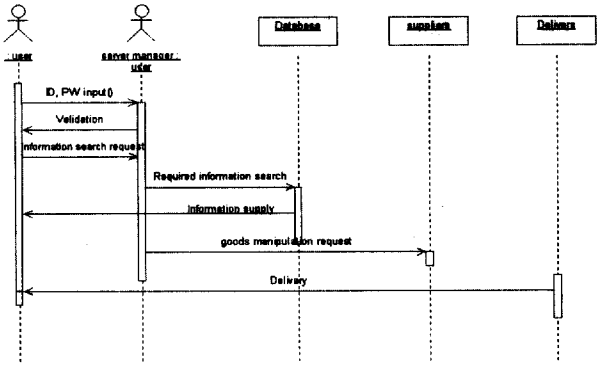
〈표 11〉 리소스 명세

기 능	세부기능	명 세
Database model	single integrated DB	모든 명세를 한 곳에 저장
Granularity	RDB	컴포넌트들간의 관계 표현
primary key		고객, 납품업자 번호, 물품 번호
foreign key	logical foreign key	서버 정보
	physical foreign key	서버번호, 데이터베이스 번호
normalization		2NF
access	Direct SQL access	정보요청, 신규등록, 정보검색
relationship	one to many	서버와 고객
	many to many	고객과 물품

또한, e-비즈니스 컴포넌트의 프로세스 측면에서 <표 8>~<표 11>에서 얻은 명세들은 다음과 같은 Use Case Diagram(그림 14)과 Sequence Diagram(그림 15)로 명세화된 내용들을 도식화 할 수 있도록 표현하였다. 사용자와 서버 관리자 측면에서의 동작을 고려하여 작성하였다.



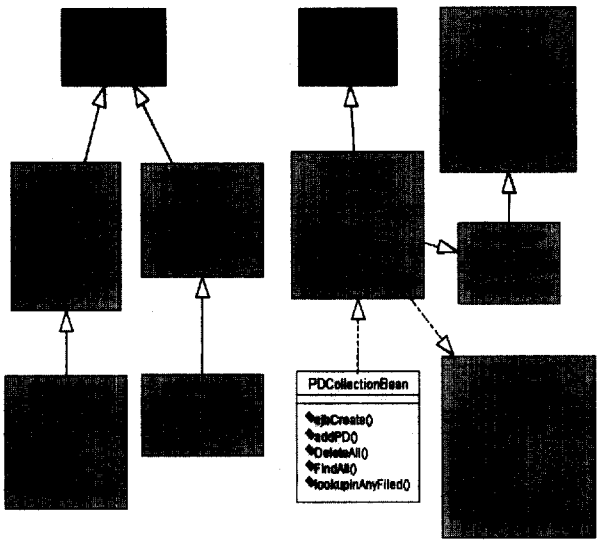
(그림 14) Use case diagram



(그림 15) Sequence Diagram

4.2.1 세션 빈 작성

EJB 컴포넌트는 클라이언트가 호출하여 사용하는 메소드를 포함하고 있다. 또한, 세션 빈과 엔티티 빈으로 나누어지며 여러 개의 클래스들과 그 객체로 구현되어 동작한다. 세션빈은 비즈니스 프로세스를 나타내는 컴포넌트로서 기업내의 업무를 상징한다. (그림 16)에서는 앞에서 분석한 웹 상에서의 쇼핑몰에서의 물품구매에 관련된 비즈니스 프로세스들을 세션 빈으로 표현하였다.



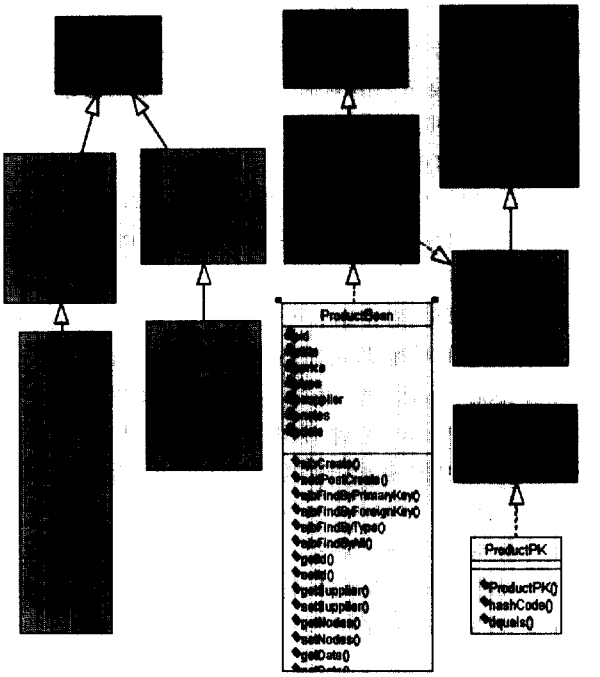
(그림 16) 세션 빈 다이어그램

4.2.2 엔티티 빈 작성

비즈니스 프로세스가 사용하는 데이터들의 집합을 상징하는 엔티티 빈은 일반적으로 데이터베이스의 테이블을 상징한다. (그림 17)에서는 쇼핑몰에서 구매자와 공급자와의 관계를 나타내는 속성값과 객체들로 표현되었다.

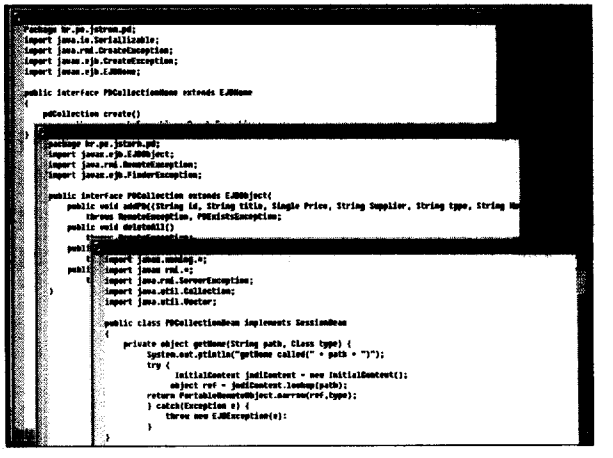
4.3 구현단계

개발한 엔터프라이즈 빈이 프로세스나 작업의 과정부분에서는 세션 빈을, 데이터베이스 부분에서는 엔티티 빈에서 작성한다. 서버의 세션빈을 사용하기 위해 홈 인터페이스의

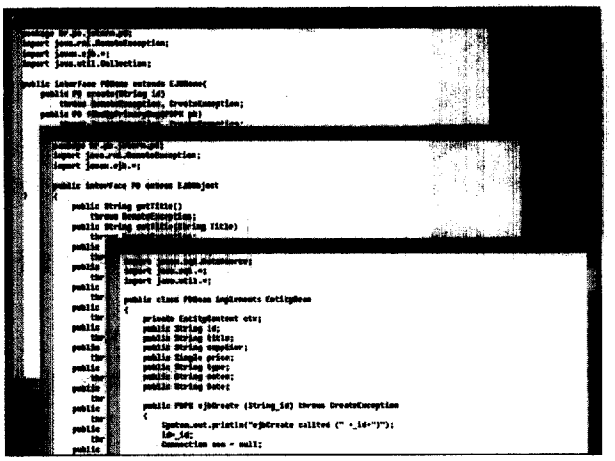


(그림 17) 엔티티 빈 다이어그램

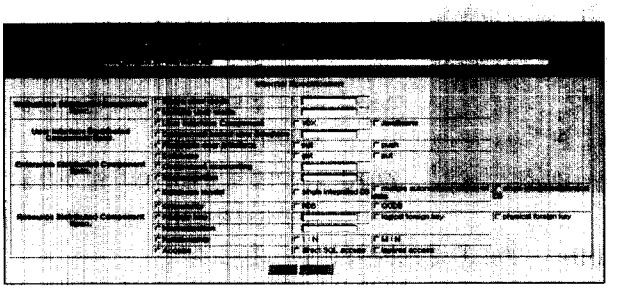
제시하며 각각 리모트, 홈 인터페이스와 빈 클래스 소스를 create()를 호출하여 EJB 객체를 생성하고, 홈 인터페이스를 직접 구현한 클래스는 자동으로 생성되고 적절한 시기에 그 클래스에서 홈 객체가 형성된다. 또한 홈 객체를 통해 EJB 객체를 생성하고 그 레퍼런스를 얻은 후 리모트 인터페이스에 선언된 메소드를 호출하도록 리모트 인터페이스를 작성하였다. 다음은 가장 중요한 부분으로서 비즈니스 메소드와 EJB 컨테이너 규약 메소드를 사용한 두가지 종류의 엔터프라이즈 빈 클래스를 작성하였다. 이들은 마지막으로 XML[9]로 작성된 웹에서의 분산 컴포넌트의 명세는 사용자가 원하는 부분을 작성하여 이를 명세화한 내용을 제시하였다.(그림 18)~(그림 22). 이러한 명세는 결국 EJB 형태로 저장되어진다.



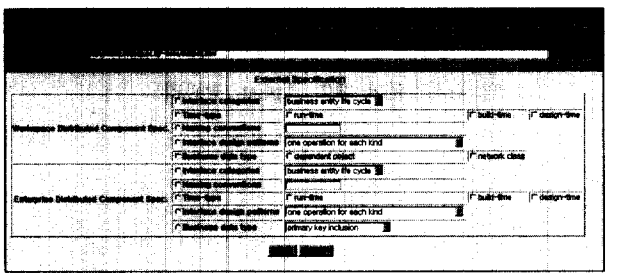
(그림 18) 세션 빈의 홈, 리모트 인터페이스와, 빈 클래스



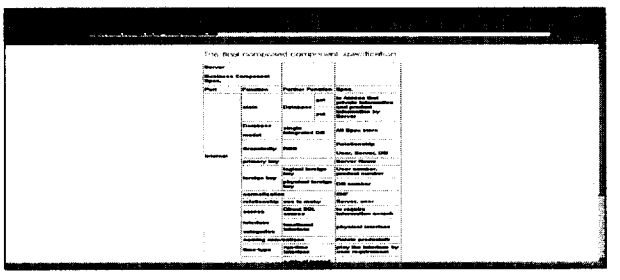
(그림 19) 엔티티 빈의 홈, 리모트 인터페이스와, 빈 클래스



(그림 20) 웹에서의 내부명세 작성



(그림 21) 웹에서의 외부명세 작성



(그림 22) 조합된 e-비즈니스 컴포넌트 명세

5. 결 론

현재 인터넷을 이용한 비즈니스 거래와 분산 업무 처리가 확대되면서 인터넷을 기반으로 한 컴퓨터 환경의 변화로 정확한 정보와 서비스를 지원하기 위해 컴포넌트 기반 기술이 많이 확대되고 있다. 이는 보다 독립적이고 확장성이 있는 재사용 부품을 만들고 이들간의 조직화된 관련성을 이룩하여 하나의 소프트웨어를 표준화하고 신뢰성이 인정되는 소프트웨어 모듈들로 조립하는 것이다. 이로 인해 개발시간과 비용을 줄일 뿐 아니라 유지보수에 효과적이고 능동적으로 대처할 수 있도록 함으로써 다양한 연구에서 이론적으로 제시된 개념을 실현할 수 있게 되었다.

본 논문에서는 네트워크 접근 가능한 인터페이스를 제공하는 4가지의 분산 컴포넌트의 요소로 나뉘어 이들이 비즈니스 컴포넌트들로 구성되기 위해 분산 컴포넌트 명세 개발 프로세서를 통해 각각의 역할과 기능들을 정확하게 분석/설계 명세로 정의하고, 여기서 얻어진 명세를 기반으로 비즈니스 컴포넌트를 구축하였다. EJB형식의 컴포넌트 명세를 작성하며 연속적인 사용자와의 인터페이스를 위해 서버내의 애플릿인 서블릿을 구축하여 트랜잭션 관리 및 재사용성을 증대시킬 수 있도록 하였다. 또한 요구가 발생할 때마다 사용자는 요구 명세를 통해 부품에 대해 질의를 작성하여 결과를 빠르고 쉽게 얻을 수 있는 인터페이스로 제시하였다.

결국, 인터넷 기반의 비즈니스 업무 처리 서비스를 폭넓게 하고 효율적으로 관리함으로써 과다한 정보 제공을 위한 시간적, 공간적 비용을 줄일 수 있도록 하였다.

참 고 문 헌

[1] John E. Mann, "Rules for E-Business," Avaliable by web server from http : //www.psgroup.com, April, 2000.

[2] Peter Eeles, Oliver Sims, Building Business Object, OMG Press, 1998.

[3] Jeff Sutherland, "The Emergence of a Business Object Component Architecture," Available by web server from http : // jeffsutherland.org/oopsla99/, 1999.

[4] Peter Herzum, Oliver Sims, Business Component Factory, OMG Press, 2000.

[5] 차정은, 컴포넌트 기반 개발 프로세스 지원을 위한 컴포넌트 저장소의 설계 및 구현, 대구가톨릭대학교대학원 전산통계학 전공 박사학위 청구논문, February, 2001.

[6] Paul Allen, *Realizing e-Business with Components*, Addison-Wesley, 2001.

[7] 박지훈외 5명, *엔터프라이즈 자바빈즈*, 대청, June, 2001.

[8] Chris Marshall, *Enterprise Modeling with UML*, Addison-Wesley, 2000.

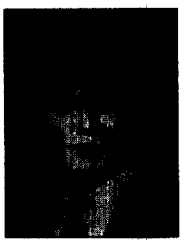
[9] M.-l. Lo, S.-K. Chen, S. Padmanabhan, and J.-Y. Chung, "XAS : A System for Accessing Companionized, Virtual XML Documents," 23rd International Conference on Software Engineering, pp.493-506, May, 2001.

[10] 김행곤외 3인, " 분산 컴포넌트 명세에 기반한 비즈니스 컴포넌트 구축에 관한 연구", 2000 한국정보과학회 추계학술논문 발표회, 포스터 발표, 2000.

[11] 김행곤외 2인, "웹 환경에서의 e-business 컴포넌트에 관한 연구", 2001년 3월 한국정보처리학회 소프트웨어 공학연구회지, 2001.

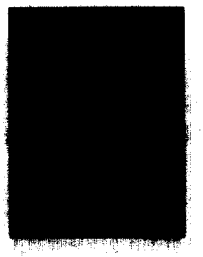
[12] Kim Hang Kon, Choi Ha Jung, Han Eun Ju, "A Study of Distributed Component Specification to build e-business

component on Servlets," SNPD, pp.177-185, August, 2001.
[13] Sun, *Designing Enterprise Application with the Java™ 2 Platform*, Enterprise Edition, May, 2000.
[14] Wilkes, Lawrence, *Understanding Component Based Development*, Addison-Wesley, June, 2000.
[15] Paul Harmon, "UML Model E-Business," Software Magazine Apr/May, 2001.
[16] Frank P. Coyle, "Legacy Integration-Changing Prospectives," IEEE Software, March/April, pp.37-41, 2000.
[17] 조완수, *UML 객체 지향 분석-설계*, 홍릉과학출판사, 2000.
[18] Craig Larman, *Applying UML and Patterns*, Prentice Hall, 1998.
[19] Kurt C. Wallnau, etc. *Building Systems from Commercial Components*, Addison Wesley, 2001.
[20] George T. Heineman, etc. *Component-Based Software Engineering*, Addison-Wesley, 2001.



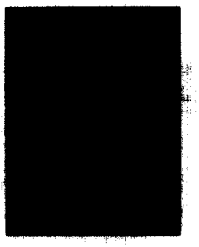
김 행 곤

e-mail : hangkon@cuth.cataegu.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
1990년~현재 대구가톨릭대학교 컴퓨터공학과 부교수
2001년~현재 Central Michigan University 교환교수
관심분야 : CBSE, 소프트웨어 재공학, CASE, 유지보수 자동화 툴, 사용자 인터페이스, 요구공학 및 도메인 공학



최 하 정

e-mail : tresa@netsgo.com
1993년 효성여자대학교 전자계산학과 (이학사)
1996년 대구효성가톨릭대학교 대학원 전산통계학과(이학석사)
2001년 대구가톨릭대학교 대학원 전산통계학과 박사수료
1998년~현재 대구가톨릭대, 대구산업정보대학 시간강사
관심분야 : CBSE, 소프트웨어 재공학, B2B, EJB, Intranet Application



한 은 주

e-mail : hanmentor@hanmentor.com
1994년 경일대학교 전자계산학과(공학사)
1996년 대구가톨릭대학교 대학원 전산통계학과(이학석사)
1999년 대구가톨릭대학교 대학원 전산통계학과(이학박사)
1999년~2001년 대구산업정보대학 겸임교수
2001년~현재 Spomation 기술자문
2001년~현재 NetMan 선임연구원
2001년~현재 경일대학교 초빙교수
관심분야 : 소프트웨어 분석 및 개발 방법론, CBSE, Web기반의 소프트웨어 개발, Agent, 망관리