

# 비즈니스 도메인을 기반으로 한 분산 컴포넌트 개발에 관한 연구

김행곤\*, 최하정\*, 한은주\*  
\* 대구효성가톨릭대학교 컴퓨터공학과

## A Study on the Development of Distributed Component based on Business Domain

Haeng-Kon Kim\*, Ha-Jung Choi\*, Eun-Ju Han\*  
\* Software Engineering Lab., Dept. of Computer Engineering,  
Catholic University of Taegu Hyosung  
<http://selab.cataegu.ac.kr>

### 요약

컴포넌트 지향 소프트웨어 개발은 독립적이고 확장성 있는 재사용 부품을 만들어 표준화되고 신뢰성 있는 소프트웨어 모듈 조립에 의한 것이며, 개발비용과 시간을 줄이고 유지보수를 용이하게 한다. 이러한 컴포넌트는 여러 가지 도메인 상에서 제공될 수 있도록 기존의 컴포넌트 생성을 위한 방법론이 제시되어져 왔다. 본 논문에서는 비즈니스 도메인을 기반으로 요구사항에서부터 분석 및 설계, 구현, 그리고 실행에 이르는 분산형 비즈니스 컴포넌트를 생성하는 개발 프로세스를 제시하여, 확장된 형태의 4가지 비즈니스 컴포넌트(attribute, operation, facility, package)를 얻어낼 수 있도록 하며, 분산형 컴포넌트는 웹을 기반으로 세분화한 시스템 관리를 단순하게 하고, 비즈니스 컴포넌트의 재사용성을 높이고 유지보수를 줄이며, 컴포넌트간의 통합을 쉽게 한다. 또한, 사용자가 원하는 실제계의 비즈니스 개념을 올바르게 적용할 수 있으며, 단일 개념의 단순한 형태로 사용자가 원하는 내용을 쉽게 변화시킬 수 있도록 하여 미숙한 재사용자들이 빠르고 효과적으로 활용할 수 있도록 한다.

## 1. 서론

컴포넌트지향 소프트웨어 개발이 지배적인 패러다임으로 인정되고 있는 것은 보다 독립적이고 확장성이 있는 재사용 부품을 만들며 이들 간의 조직화된 관련성을 이룩하여 하나의 소프트웨어를 표준화되고 신뢰성이 인정되는 소프트웨어 모듈들을 조립할 수 있기 때문이다. 이로 인해 개발 시간과 비용을 줄일 뿐 아니라 유지보수에 효과적이고 능동적으로 대처할 수 있도록 함으로써 다양한 연구에서 이론적으로 제시된 재사용의 개념을 실현할 수 있게 된다. 또한, 패키지 되어진 프레임워크를 선택적으로 구입, 적용할 수 있는 컴포넌트 소프트웨어 형태는 소프트웨어 재사용이 용이하다[1].

현재 컴포넌트 기술은 시스템 구축 시에 구성요소의 모듈화를 촉진하고 재사용성을 높여 최종적으로 인터페이스를 얻도록 하는데 목적을 두고 있으며, 어느 도메인에 구애 없이 구축될 수 있도록 컴포넌트 생성을 위한 기존의 여러 가지 방법론이 제시되어져 왔다[2].

본 논문에서는 현재 대중화된 웹을 기반으로 비즈니스 도메인 측면에서의 분산적 컴포넌트를 생성하도록 개발 프로세스를 제공하고 기존의 비즈니스 컴포넌트 범주를 확장시켜 4가지의 다양한 비즈니스 컴포넌트를 얻도록 하여 세분화된 비즈니스 컴포넌트 재사용을 가능토록 한다. 이러한 컴포넌트는 웹을 기반으로한 응용 시스템을 통해

사용자와 인터페이스 가능토록 하며, 하부구조를 통해 다양한 비즈니스 컴포넌트를 접할 수 있도록 하는데 그 목적을 두고 있다. 또한, 분산적 컴포넌트를 기반으로한 비즈니스 컴포넌트는 요구사항 분석에서부터 시스템 개발과 변화에 이르기까지 거의 독립적이고 반복적으로 행해지며, 결국 BOF(Business Object Facility)구조로 형성된다.

## 2. 관련 연구

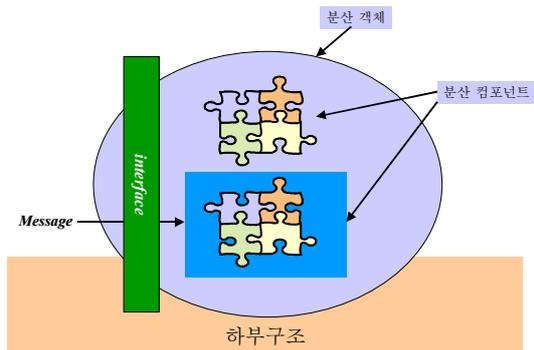
### 2.1 분산 컴포넌트 기술

비즈니스 컴포넌트 계층은 분산 컴포넌트로 구현되어질 수 있으며, 독립적으로 개발되어 수행 가능한 모듈이 된다. 이러한 분산 컴포넌트는 직접 비즈니스 컴포넌트 계층의 독립적인 개발을 제공하고 가능케 하며, 개발과 수행 시간에서는 결국 비즈니스 컴포넌트 그 자체가 된다. 스트레드 관리와 통신 프로그래밍, 그리고 병행관리등과 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.

이러한 분산 컴포넌트 접근을 통해 재구축 또는 변경 없이 또다른 하나의 컴포넌트를 서브 클래스화함으로써 상세화 할 수 있으며, 다수의 프로그래밍 가능한 언어들

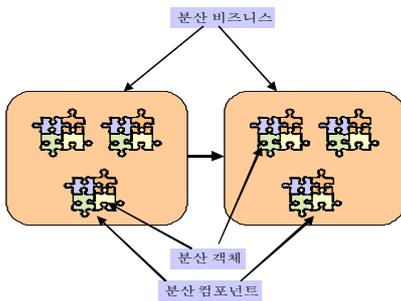
중 어느 것으로 작성하여도 무방한 중간적인 응용 코드로 표현된다. 또한, 기술적인 바인딩을 통해 다른 플랫폼으로 이식이 이루어지며, 메시지내의 메타 데이터를 사용하여 클라이언트의 변화 가능한 요구사항과는 독립적으로 인터페이스가 개발되어지므로 최소화된 의존성을 가진다.

(그림 1)에서는 분산 컴포넌트로 조합된 집합은 결국 비즈니스 컴포넌트인 분산 객체의 구현을 나타낸다. 분산 객체들간의 메시지는 메시지 파라미터들에 대한 메타 데이터로 운반되어지며, 각 분산 객체들은 메시지를 번역하고 설계 때 정의된 인터페이스로 접근되어진다[3].



(그림 1) 분산 컴포넌트의 구조

(그림 2)에서는 분산 컴포넌트와 비즈니스 컴포넌트간의 관계를 나타낸다. 분산 컴포넌트는 분산 객체로 이루어지며, 여러 단계에서의 비즈니스 개념들을 자동적으로 가지고 단일 단계에서의 비즈니스 컴포넌트 상에서 분산 객체를 구현하는 기술적인 컴포넌트라 할 수 있다. 반면에, 비즈니스 컴포넌트는 다수의 분산 클래스로 구성되며, 각각 역할을 가지고 있고, 큰 입자로 된 비즈니스 개념의 패키징화된 형태로 개발된 것이다[4].

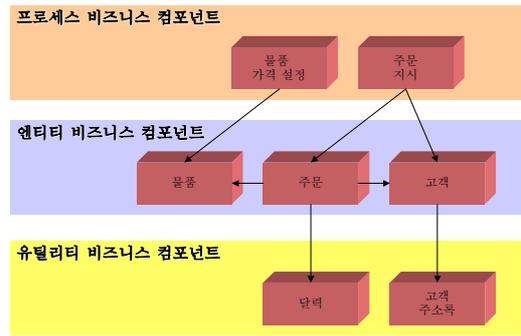


(그림 2) 비즈니스 컴포넌트와 분산 컴포넌트간의 관계

## 2.2 비즈니스 모델에서의 엔티티/프로세스/유틸리티 컴포넌트

### 1) 프로세스 비즈니스 컴포넌트(process business component)

문제 도메인에서 식별된 비즈니스 프로세스들을 나타낸 것으로, 처리에 관한 특정 논리를 가지고 있는 컴포넌트라 할 수 있으며, 도메인에서의 비즈니스 행위를 제시한다.



(그림 3) 비즈니스 모델에서의 비즈니스 컴포넌트 예

예를 들어, (그림 3)에서처럼 구매자가 특정 물품에 대한 주문을 하는 행동과 이에 대한 공급자의 가격 절충에 따른 결정의 행위부분을 컴포넌트화 할 수 있다.

### 2) 엔티티 비즈니스 컴포넌트(entity business component)

문제 도메인에서 식별되어진 엔티티, 즉 속성들에 대한 부분으로, 변화하지 않는 정적인 성격을 가지고 있는 컴포넌트이며, 비즈니스 프로세스 수행에서의 비즈니스 개념들을 제시한다. 또한, 데이터, 객체 저장소, 그리고 비즈니스 사용에 관한 서비스를 내포하고 있다. 예를 들어, 가격 결정에 따른 물품과 주문지시에 대한 주문, 고객, 판매자 등이 엔티티 비즈니스 컴포넌트에 속한다.

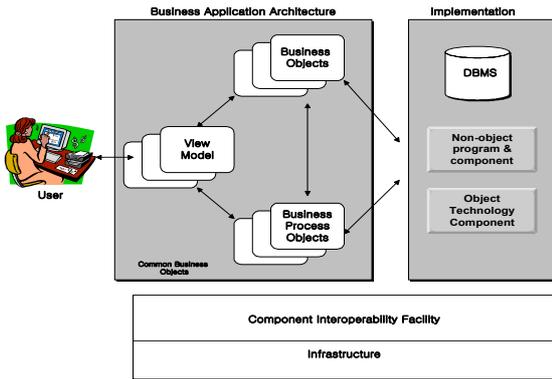
### 3) 유틸리티 비즈니스 컴포넌트(utility business component)

모든 상위 단계 종류의 비즈니스 컴포넌트를 모은 것으로, 프로세스 비즈니스 컴포넌트와 엔티티 비즈니스 컴포넌트를 통합화한 형태라고 할 수 있으며, 비즈니스 컴포넌트가 완전히 만들어진 후 사용되어진다[5].

## 2.3 비즈니스 컴포넌트 응용 구조

비즈니스 응용 구조는 중관계층에서 비즈니스 객체를 지닌 3단계 표준 구조로 이루어져 있으며, 프로세스형의 비즈니스 객체와 엔티티 형의 비즈니스 객체들로 구분 지어 표현되었다(그림 4). 비즈니스 응용구조는 OMG (Object Management Group)의 Common Facility인 RFP(Remote Facility Protocol)-4의 영향으로 일반 비즈니스 객체와 CIF(Component Interoperability Facility)인 즉 BOF(Business Object Facility)로 형성되었다.

RFP의 영향으로 plug & play 환경을 이룰 수 있는 OMG CORBA(Common Object Request Broker Architecture)를 구축하기 시작하였다. CORBA 하부구조는 분산 객체들간의 통신이 가능한 환경을 제공하며, CIF는 비즈니스 객체에 관한 일반적인 슈퍼 클래스를 제공한다. 도메인 기반 공통 비즈니스 객체는 표준화되어 있으며, 도메인 프레임워크는 공통 비즈니스 객체와 CIF 두 가지를 사용하여 개발되어진다[6].



(그림 4) 비즈니스 컴포넌트 응용 구조

### 3. 비즈니스 도메인을 기반 한 분산 컴포넌트 개발

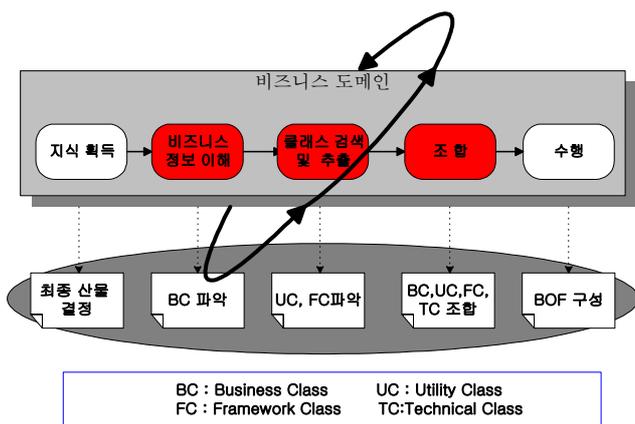
#### 3.1 분산 비즈니스 컴포넌트 개발 프로세스

##### 1) 요구사항(지식획득) 단계

사용자가 요구하는 비즈니스 개념을 얻기 위해 문제 도메인의 형태를 그려내고, 주요 비즈니스 개념을 식별할 수 있으며, 최종 비즈니스 산물 및 서비스를 결정한다.

##### 2) 분석(비즈니스 정보 이해) 단계

요구사항을 통해 비즈니스 개념을 추출하고, 비즈니스 측면의 논리를 적용한 클래스부터 파악하여 적용 가능하도록 한다. 분석단계에서 구현에 이르기까지는 계속 반복적으로 애트리뷰트, 프로세스, 그리고 활용 컴포넌트가 작성될 때까지 행해질 수 있다.



(그림 5) 분산 컴포넌트 개발 프로세스

##### 3) 설계(클래스 검색 및 추출) 단계

비즈니스 클래스 외의 프레임워크 클래스, 유틸리티 클래스를 얻어낸다. 프레임워크 클래스는 비즈니스 산물에서 가장 기본적으로 행해질 수 있는 동작이 내포된 클래스이며, 유틸리티 클래스는 클래스들간의 메시지 전달을 수행하는 클래스라고 할 수 있다. 이러한 클래스들은 결과적으

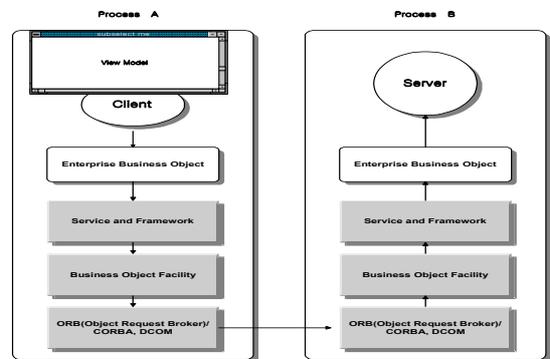
로 UML(Unified Modeling Language)을 통해 행위과정을 나타내고, 컴포넌트 메타모델을 얻도록 한다.

##### 4) 구현(조합) 단계

기술적인 클래스를 적용하여, 실제 클래스를 조합하는 부분으로서, client와 server간의 UID(User Interface Domain)인 View layer와 SRD(Shared Resource Domain)인 데이터베이스를 통한 공유 자원인 비즈니스 상태와 행위를 가지는 Enterprise layer가 있으며, 인터페이스를 위해 BOUI(Business Object User Interface)가 제공되는데, 이는 실제 세계에서 행해지는 수많은 동작들을 가진 컴포넌트 상태와 행위들이 사용자의 모델로 표현할 수 있도록 한다. 사용자의 편의를 위해 패키지 형태의 비즈니스 컴포넌트로 작성하도록 조합되어진다.

##### 5) 수행 단계

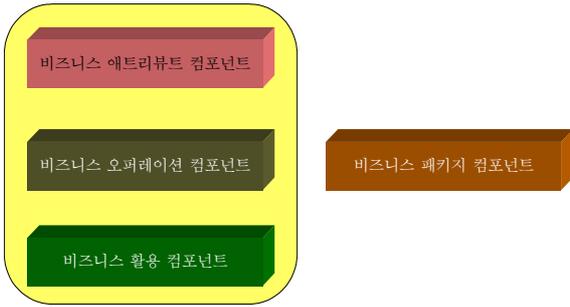
내포된 분산 클래스가 합쳐진 분산 객체인 비즈니스 컴포넌트를 생성하도록 하는데, 재사용 가능한 4가지의 분산 비즈니스 컴포넌트를 개발하여 수행하도록 하며 이는 결국 BOF(Business Object Facility)형태로 제공되어진다. 확장된 형태의 분산 비즈니스 컴포넌트는 다음의 3.2절에서 자세히 제시하도록 한다. (그림 6)에서는 이러한 분산 환경에서의 비즈니스 컴포넌트들간의 인터페이스를 통한 메시지 전달 수행을 나타내었다.



(그림 6) 비즈니스 컴포넌트 인터페이스를 통한 메시지 전달

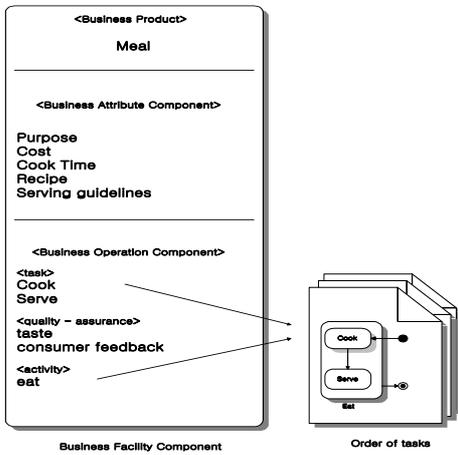
#### 3.2 확장된 분산 비즈니스 컴포넌트

비즈니스 도메인에서의 최종목표는 비즈니스 사용자에게 가치 있고 다양한 방법론과 비즈니스 프로세스에 의해 생성되거나 얻어질 수 있는 정보라 할 수 있다. 기존의 비즈니스 컴포넌트의 범주로 엔티티, 프로세스, 그리고 유틸리티 세 가지 종류의 컴포넌트가 사용되었으나, 사용자의 요구사항과 비즈니스 도메인의 확장으로 인해 많은 분산 비즈니스 컴포넌트를 제시할 수 있다. 이들은 모듈화 되고 캡슐화된 비즈니스 산물이자 서비스이며 웹을 통해 전달되어지며 비즈니스 컴포넌트 저장소를 통해 관리되어진다.

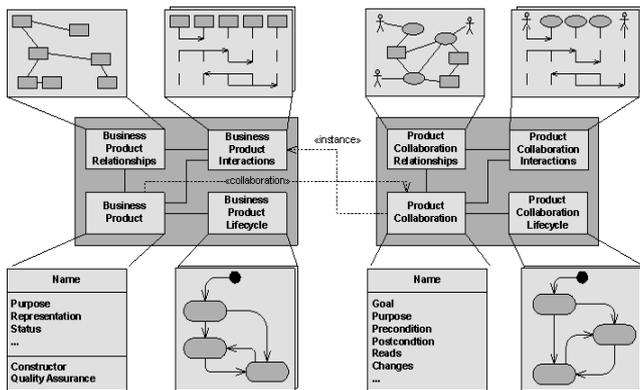


(그림 7) 확장된 분산 비즈니스 컴포넌트

본 논문에서는 3.1절의 프로세스를 통해 얻어질 수 있는 다음의 4가지 비즈니스 컴포넌트 카탈로그를 제시(그림 7)하였고 이를 UML로 표현한 것이 (그림 8-a, b)이다.



(그림 8-a) 분산 비즈니스 컴포넌트 카탈로그



(그림 8-b) 분산 비즈니스 컴포넌트 UML 명세

1) 비즈니스 애트리뷰트 컴포넌트(Business Attribute Component)  
비즈니스 산물의 애트리뷰트들과 속성들을 나타낸 부분으로 정적인 성격의 데이터를 집합이라고 할 수 있으며 컴포넌트들을 가시적으로 나타낸다.

2) 비즈니스 오퍼레이션 컴포넌트(Business Operation Component)  
비즈니스 산물을 생성하거나 수행하는 가장 작은 작업 단위의 타스크(task)와 타스크들의 집합체인 행위(activity)들로 이루어진 컴포넌트로, 실제 컴포넌트의 수행을 담당

하는 동적인 부분을 담당하는 컴포넌트이다.

3) 비즈니스 활용 컴포넌트(Business Facility Component)

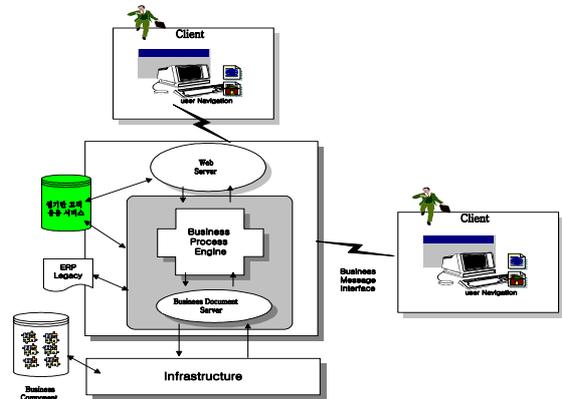
비즈니스 애트리뷰트 컴포넌트와 오퍼레이션 컴포넌트들을 조합하여 상위의 컴포넌트로 수행되는 활용 컴포넌트는 비즈니스 컴포넌트 재사용에 대한 단순한 기본 단위로 일컫는다.

4) 비즈니스 패키지 컴포넌트(Business Package Component)

비즈니스 패키지 컴포넌트는 활용 컴포넌트들간의 collaboration으로 최상위 컴포넌트라 할 수 있다. 크게 두 가지 성격을 가지는데, 비즈니스 산물에 관련된 관계, 상호동작, 생명주기, 그리고 collaboration에 관련된 관계, 상호동작, 생명주기에 대한 패키지 형태가 제공된다. 이러한 비즈니스 패키지 컴포넌트는 재사용하기 용이하며 쉽게 적용 가능하다.

### 3.3 웹에서의 분산 비즈니스 컴포넌트 시스템

다음(그림 9)과 같이 분산 비즈니스 컴포넌트를 세분화된 여러 단계로 나뉘어 시스템을 관리함으로써, 단순화된 작업을 행할 수 있도록 한다.



(그림 9) 비즈니스 컴포넌트 응용 시스템 구조

1) 사용자(Client)

스크린 상에서 존재하는 비즈니스 컴포넌트를 담당하는 부분으로 PC(Personal Computer) 혹은 NC(Networking Computer) 환경에서 접근 가능한 계층이며, View 모델을 통해 실제 엔터프라이즈 모델에 접근하여, 유용한 서비스를 사용하고 서비스 선택과 주문을 행할 수 있다.

2) 비즈니스 프로세스 엔진(Business Process Engine)

사용자 계층을 지원하고, 로컬 비즈니스 로직을 실행하는 계층으로, 사용자에 대한 단일 사용자 버퍼공간인 workplace를 제공하여 분산 혹은 중앙 집중식 서버 상에서 작업 가능하도록 한다. 또한, 데이터의 입력과 저장의 여부에 대한 응답을 제공한다. 여기서는 비즈니스 컴포넌트 서버의 부분으로서, 서버의 역할은 트랜지션 지향적인 컴

포넌트 액세스 프로토콜인 check-in, check-out, unlock, get 동작을 통해 비즈니스 컴포넌트 액세스를 제공하고, 비즈니스 컴포넌트 자신의 속성과 상속받은 속성을 구별하고, 모든 속성으로의 읽기 액세스는 모두 허용하고 쓰기 액세스는 특정 속성에 제한적으로 권한을 부여한다.

3) 비즈니스 도큐먼트 서버(Business Document Server)

비즈니스 컴포넌트간의 상호동작과 검증, 비즈니스 규칙을 적용하는 계층으로 데이터 무결성을 관리한다. 웹 응용의 비즈니스 컴포넌트 모델로부터 파일기반 자원들을 생성하고, 웹 서버를 통해 사용자에게 비즈니스 컴포넌트를 제시할 수 있도록 한다. 비즈니스 컴포넌트 계층을 통해 점진적으로 수정되어지는데, 비즈니스 컴포넌트 저장소의 개정 제어를 사용하고, 이미 수정된 비즈니스 컴포넌트를 포함하는 리소스를 생성한다.

4) 비즈니스 컴포넌트 저장소

공유되는 자원에 대한 실제 물리적인 처리부분으로서, 관계형 데이터베이스 시스템을 사용하여 광범위하게 사용 가능하도록 하며, 비즈니스 컴포넌트 모델을 기반으로 형성하는 프로토타입-인스턴스 모델의 단순함을 데이터베이스 테이블로 간단하게 표현할 수 있다. 데이터베이스 추상화계층을 구현하여 DBA-SQL(DataBase Access-SQL)를 통해 비즈니스 컴포넌트 액세스에 대한 데이터베이스 뷰를 가시화 할 수 있다.

5) 하부구조

컴포넌트의 효과적인 재사용 지원을 위해 CORBA, COM (Common Object Model)/DCOM(Distributed Common Object Model), EJB(Enterprise JavaBeans), JavaBean등과 같은 분산 컴포넌트를 이용한 미들웨어를 구축하여 웹상에서의 비즈니스 컴포넌트를 위한 분산 컴퓨팅을 통해 다양한 비즈니스 컴포넌트를 재사용토록 한다.

4. 사례연구

4.1 비즈니스 도메인 분석 및 단계별 적용

1) 요구분석

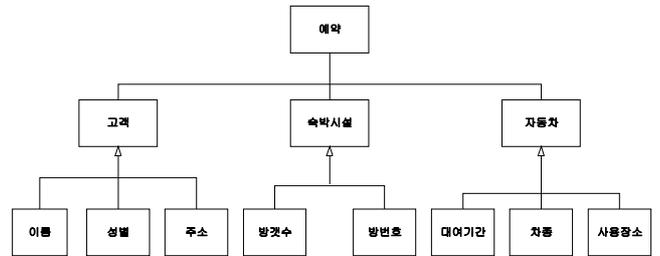
숙박 예약에서는 고객과 숙박시설, 예약날짜, 참조번호가 주어지며, 고객은 성과 이름, 결혼여부, 개인 정보가 나타난다. 리조트에는 이름, 지역, 숙박시설 개수, 참조번호가 있으며, 숙박시설에는 방번호, 수용인원, 기간, 참조번호가 주어진다.

예약 시스템은 숙박시설에 대한 예약 정보와 고객 관리 및 리조트, 그리고 숙박시설에 대한 아이템을 가지도록 한다. 고객은 자신이 원하는 날짜에 원하는 숙박시설 혹은 리조트를 예약할 수 있도록 하며, 예약시 여러 가지 정보를 입력할 수 있도록 해야 한다. 리조트와 숙박시설, 자동차 렌트등도 패키지화되어 있다. 결과적으로, 고객의 정보가 입력되어졌다면 이를 컴포넌트화 할 수 있고, 다른 시스템에도 고객의 정보가 재사용될 수 있으며, 그 행위들과도 패키징된 통합 형태의 정보를 사용하도록 하는데 목적을 둔다.

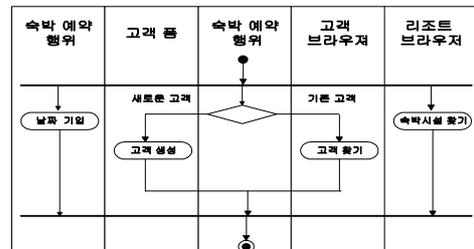
(그림 10) 비즈니스 도메인 분석

2) 분석 및 설계

UML을 통해 전체적인 예약 시스템의 정보들을 분석하고 상호 관련성을 파악한 뒤 각각 적용 가능한 클래스들을 참조하여 UML을 통한 다이어그램으로 나타낸 것이 (그림 11), (그림 12)와 같이 클래스 다이어그램과 상태 다이어그램으로 나타내었다.

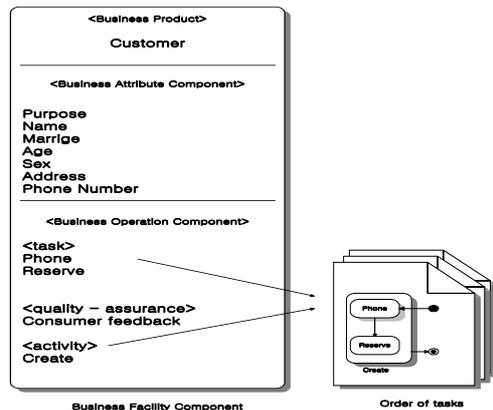


(그림 11) 숙박시설 예약 행위에 대한 클래스 다이어그램



(그림 12) 숙박시설 예약 행위에 대한 상태 다이어그램

3) 구현

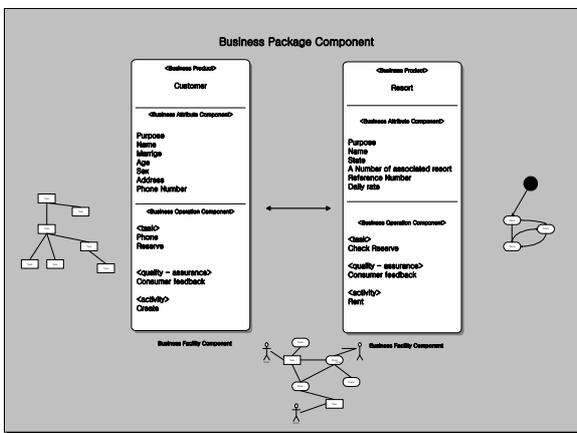


(그림 13) 숙박 예약에 대한 고객 컴포넌트 구현

(그림 13)에서는 고객에 대한 비즈니스 컴포넌트의 명세를 나타낸 것으로, 비즈니스 애트리뷰트 컴포넌트와 비즈니스 오퍼레이션 컴포넌트가 조합되어 비즈니스 활용 컴포넌트로 이루어져 사용자가 편리하게 고객이라는 컴포넌트를 재사용할 수 있도록 한다.

#### 4) 수행

마지막 단계인 실행단계에서는 고객과 숙박시설인 리조트간의 collaboration 관계를 조합하여 최상위단계의 컴포넌트인 패키지 컴포넌트로 작성한 명세를 (그림 14)와 같이 나타낼 수 있다. 4가지 유형의 컴포넌트들은 각기 독립적으로 수행가능하나 사용자의 요구에 따라 통합된 패키지 형태로도 사용 가능하다.



(그림 14) 숙박 예약에 대한 패키지 컴포넌트 생성

## 5. 결론 및 향후연구

인터넷 기술과 개인용 컴퓨터의 광범위한 확산은 소프트웨어의 새로운 응용 영역뿐만 아니라 소프트웨어 기능의 복잡성, 생산성 그리고 품질 등 새로운 쟁점들이 등장하였다. 최근 전자상거래가 인터넷/인트라넷을 통해 상용화되면서 기업의 상업적 목표에 대응하여 비즈니스 모델들이 대두되고 있으며 이는 전자상거래의 구축에 있어서 개발 시간과 비용을 줄이고 소프트웨어 재사용을 실현하는 것을 바탕으로 한다.

본 논문에서는 현재 대중화된 웹을 기반으로 비즈니스 도메인 측면에서의 분산적 컴포넌트를 생성하도록 개발 프로세스를 제공하고 기존의 비즈니스 컴포넌트 범주를 확장시켜 4가지의 다양한 비즈니스 컴포넌트를 얻도록 하여 세분화된 비즈니스 컴포넌트 재사용을 가능토록 한다. 이러한 컴포넌트는 웹을 기반으로 한 응용 시스템을 통해 사용자와 인터페이스 가능토록 하며, 하부구조를 통해 다양한 비즈니스 컴포넌트를 접할 수 있도록 한다.

이는 유지보수를 줄이며, 컴포넌트간의 통합을 용이하게 하고, 사용자가 원하는 실세계의 비즈니스 개념을 올바르게 적용할 수 있으며, 단일 개념의 단순한 형태로 사

용자가 원하는 내용을 쉽게 변화시킬 수 있도록 하여 미숙한 재사용자들이 빠르고 효과적으로 활용할 수 있도록 사용자 지향의 편리성을 제공할 수 있도록 한다.

향후 연구 방향으로는 이러한 프로세스를 통한 시스템 구축과 비즈니스 컴포넌트 저장소의 작성 및 관리에 있다.

### 【참고 문헌】

- [1] Hans-Werner Gellersen, "WebComposition : An Object-Oriented Support System for the Web Engineering Lifecycle", Computer Networks and ISDN systems, Vol.29, pp1429 ~ 1437, 1997.
- [2] Desmond D'Souza and Alan Wills, *Objects, Component, and Frameworks with UML*, Addison-Wesley, 1998
- [3] Peter Herzum and Oliver Sims, "The Business Component Approach", Available by web server from <http://jeffsutherland.org/oopsla98>, 1998.
- [4] Jeff sutherland, "Business Object Component Architecture : A Target Application Area for Complex Adaptive Systems Research", Available by web server from <http://jeffsutherland.org/oopsla98>, 1998.
- [5] Peter Eeles , Oliver Sims, *Building Business Objects*, Wiley, 1998.
- [6] Chris Marshall, "Business Object Management Architecture", Available by web server from <http://jeffsutherland.org/oopsla97>, 1997.
- [7] Jeff Sutherland, "The Emergence of a Business Object Component Architecture", Available by web server from <http://jeffsutherland.org/oopsla99>, 1999.
- [8] Trygve Reenskaug, "Multi dimensional layering of Business object component systems", Workshop 28 : Business object, OOPSLA'99, Position Paper 28, Aug., 1999.
- [9] Pavel Hruby, " Object-Oriented Architecture of Business Process Catalogue", Available by web server from <http://jeffsutherland.org/oopsla99>, 1999.