

컴포넌트 개발과정에서 UML 표기법과 컴포넌트 모델링

김 재 생[†]

요 약

UML은 응용시스템 또는 컴포넌트 개발과정에서 분석 및 설계과정을 모델링하여주는 표준언어이며, CORBA, COM+, EJB 등은 컴포넌트의 재사용을 실현화시켜주는 기술이다. 컴포넌트 개발과정에서 재사용가능한 컴포넌트와 새로 개발할 컴포넌트를 구별한 후 개발하면, 응용시스템이나 컴포넌트의 개발비용을 줄일 수 있고, 질이 좋은 컴포넌트를 생성할 수 있다. 본 논문에서는 컴포넌트의 개발과정에서 생성되는 산출물들의 모델링시에 필요로 하는 컴포넌트 속성에 따르는 확장된 UML 표기법을 정의한다. 또한, 사례 연구로 특정 영역인 웹상의 경매 시스템 개발에 필요한 컴포넌트들의 모델링 및 UML 적용기법을 소개한다.

UML Notation and Component Modeling on the Components Development Process

Jae Saeng Kim[†]

ABSTRACT

UML is a standard language that models an analysis and design process in the development of an application system or components, CORBA, COM+, EJB and others are the techniques that are realized reuse of components. After it is distinguished between reusable components and new components on the components development process and is developed, the developing cost of components or application system can be reduced and the components of superior in quality can be created. This paper defines the extended UML notation that needed in modeling of the products created from the development process of components. Also, this paper introduces a modeling of components and UML application method that needed in development of an auction system based on Web.

키워드 : 컴포넌트 개발(Component development), UML 표기법(UML notation), 컴포넌트 재사용(component reuse), 컴포넌트 모델링(component modeling)

1. 서 론

컴포넌트란 객체지향 기법을 사용하여 만든 모듈로서 소프트웨어도 컴포넌트들을 조립해서 하나의 응용 소프트웨어로 만든다는 개념이다. 컴포넌트 기술은 컴포넌트의 개발, 생성, 조립, 시험, 품질평가, 유통으로 분류하여 볼 수 있다. 새로운 컴포넌트는 요구사항서, 컴포넌트 명세, 컴포넌트 공급, 조립, 테스트, 배포와 같은 개발과정을 거쳐서 개발되어진다. 좋은 컴포넌트는 명확한 인터페이스로 정의되어 있으며, 오래된 컴포넌트를 새롭게 호환성있는 것으로 쉽게 교체할 수 있다. 컴포넌트의 각 개발단계에서 발생되는 산출물 중 모델이나 명세서 등은 컴포넌트 조립이나 구

현시에 많이 반영되므로 더욱 중요하다[1].

이러한 컴포넌트 개발과정에서 컴포넌트를 모델링하거나 명세화할때에 UML, CORBA, COM+, EJB 등과 같은 도구를 사용하게 된다[2]. UML은 언어확장 메카니즘 스테레오 타입을 사용하여 컴포넌트 기반 개발을 지원해줄 수 있다. UML로 기술된 상세하고 명확한 명세는 컴포넌트 개발과정의 각 단계마다 기본 정보로서 재사용되며, 이때 개발된 컴포넌트는 테스트를 거쳐 사용자에게 공급되어진다[5, 7].

컴포넌트의 각 개발과정에 컴포넌트의 재사용 속성을 적용하여 재사용가능한 컴포넌트와 새로 개발할 컴포넌트를 구별하여 개발하면, 개발시간도 줄이고 질이 좋은 컴포넌트를 생성할 수 있다. 그러므로 개발자가 산출물을 이해하기 쉽도록 컴포넌트의 속성을 UML 스테레오 타입으로 정의하여 각 산출물에 표현해줄 수 있다면, 개발자는 응용시스템

* 본 연구는 2001년 김포대학 교내 연구지원비에 의해 연구되었음.

† 정 회 원 : 김포대학 컴퓨터 계열 교수

논문접수 : 2001년 10월 5일, 심사완료 : 2001년 12월 26일

이나 컴포넌트를 쉽게 개발할 수 있을 것이다.

본 논문에서는 컴포넌트의 각 개발과정에서 생성되는 산출물들의 모델링시에 필요로 하는 확장된 UML 표기법을 정의하였다. 또한, 사례 연구로 특정 영역인 웹상의 경매 시스템 개발에 필요한 컴포넌트들의 모델링 및 UML 적용 기법을 소개하였다.

2. 관련 연구

2.1 UML과 RUP

UML(Unified Modeling Language)은 1996년 6월 Grady Booch의 Booch 방법론, Ivar Jacobson의 OOSE(Object Oriented Software Engineering) 방법론, James Rumbaugh의 OMT(Object Modeling Technique)방법론을 통합하여 개발된 소프트웨어 모델링의 표준언어이다[3].

UML은 그래픽언어로서 소프트웨어 개발과정에서 나오는 산출물들을 가시화, 명세화, 구축, 문서화하는 역할을 수행할 뿐만 아니라 컴포넌트 기반 개발을 지원할 수 있도록 언어 확장 메카니즘인 스테레오 타입(Stereo types, << >> 기호)을 지원한다. 컴포넌트에 관한 표기법은 실행파일 <<executable>>, 라이브러리 <<library>>, 테이블 <<table>>, 파일<<file>>, 문서 <<documentation>> 등 5개의 표준 타입과, 패키지의 명세정보를 가져오는 관계인 <<import >>, 제약 조건 <<precondition>>, <<postcondition>>, 유즈케이스를 포함하는 <<include>>, 내용 추가나 확장 <<extend >> 등이 정의되어 있다.

RUP(Rational Unified Process)는 반복적인 소프트웨어 개발, 요구 명세서 관리, 컴포넌트 기반 구조, 모델의 시각화, 변화 제어, 품질 검증 등과 같은 6가지 기능을 제공한다. 또한, 도입, 정련, 구축, 전이의 4가지 단계를 세분화된 반복과정을 통하여 프로젝트를 수행하여 점증적으로 최종 시스템으로 진화해가는 일련의 반복적 개발 방법이다[3, 9]. 이 프로세스는 개발 시스템의 규모가 크고 개발시간이 길고 사용자의 요구사항이 실시간에 따라 반영하여야 하는 경우에 주로 사용되고 있다.

2.2 기존의 컴포넌트 모델링 방법들

Catalysis 방법론은 요구사항 분석 -> 시스템 명세서 -> 구조 설계 -> 컴포넌트 내부 설계 등의 각 개발과정에 UML 표기법을 사용하여 컴포넌트를 개발하였다[8]. 각각의 개발과정이 컴포넌트라는 개념하에서 순차적으로 컴포넌트를 개발하는 방법이다. UML을 사용하여 각 개발단계에 따라 체계적으로 개발하기 때문에 작은 규모의 프로젝트 개발에는 알맞지만 커다란 프로젝트일 경우에는 적용하기가 어렵다는 단점을 가진다.

Rational사의 개발도구는 대표적인 객체지향식 개발 방법론을 지향한다. 소프트웨어의 개발과정은 RUP(Rational Unified Process)와 UML을 기반으로 Rose 도구를 사용하여 컴포넌트와 인터페이스를 모델링할 수 있다[9]. 생명주기 각 개발단계에 RUP의 반복과정을 적용하여 컴포넌트를 개발할 수 있으므로 커다란 규모의 프로젝트에도 사용할 수 있다는 장점을 가지고 있으나 명세와 구현을 연결해주는 인터페이스 개념과 재사용을 완전히 지원해주지 못한다는 단점을 가지고 있다.

John Cheesman과 John Daniels는 컴포넌트개발과정에서 생성되는 각 산출물에 UML 스테레오 타입을 확장하여 정의하였다[6]. 이들은 UML을 확장하여 컴포넌트 기반 시스템의 아키텍처를 설계하고 명세화하는 방법을 제시하였다. 컴포넌트 명세, 컴포넌트 인터페이스, 구현 컴포넌트, 컴포넌트 오브젝트 등의 컴포넌트 개념들과 UML 확장 스테레오와 RUP 워크플로우를 실제 컴포넌트 개발에 적용하였다.

현재 대부분의 UML 도구들은 아직 컴포넌트 기반 개발을 제대로 지원하지 못하고 있으며, 특히 컴포넌트의 기능면과 명세면에서 UML 표기법을 완전히 지원해주지 못하고 있다. 또한, 컴포넌트의 각 개발과정에 재사용 가능한 컴포넌트와 산출물들의 모델링을 지원해주는 재사용에 관한 UML 표기법 지원도 필요하다.

본 논문에서는 컴포넌트의 각 개발과정에서 재사용이 이루어질 수 있도록 컴포넌트의 재사용 속성과 확장된 UML 표기를 정의하였다.

3. 확장된 UML 표기법의 제안

컴포넌트 개발단계에서 생성되는 산출물들은 여러 가지 종류가 있다. 우리가 모델링하고자 하는 클래스나 컴포넌트거나 어느 개발단계에 속하는 산출물인지 알기 위해서는 UML 표기를 확장해줄 필요가 있다.

비즈니스 개념 모델(Business Concept Model)은 업무영역의 지식을 표현하는 순수한 비즈니스만 표현하여 Requirements 패키지에서 저장 및 관리하므로 스트레오타입을 사용할 필요는 없다. 그러나 이 모델안의 모든 클래스에 대하여 <<Business Concept>> 스테레오 타입을 사용하여 다른 단계의 클래스와 서로 의미가 다르다는 것을 구분해준다.

비즈니스 타입 모델(Business Type Model)은 비즈니스 개념 모델을 토대로 대상 시스템의 비즈니스에 대한 정보(규칙과 다양한 시각의 도식화)를 분명하게 모델링한다. 이 모델에서 생성되는 클래스를 다른 단계의 클래스와 구별하기 위해서 <<Business Type>> 타입으로 정의한다.

개발하고자 하는 클래스나 컴포넌트를 모델링 할 때 사용하는 속성은 일반속성, 매개변수 속성, 재사용 컴포넌트 속성 등 3가지로 구분하기로 한다. 한 클래스에서 사용하는 일반속성인 데이터명과 타입을 <<Attribute>>로 정의하고, 매개변수를 갖는 속성은 일반속성과 특성이 다르므로 <<Parameter Attribute>>와 같이 정의한다. 또한, 개발하려고 하는 컴포넌트가 재사용가능도가 아니면 새로 개발해야 하는지를 미리 알기 위해서는 컴포넌트 개발 초기단계에서부터 재사용자와 제공자간에 서로 이해할 수 있는 컴포넌트 재사용 가능 식별이 필요하므로 <<Reuse Attribute>>로 정의한다. <<Reuse Attribute>>는 재사용 컴포넌트의 특성과 컴포넌트 정의들을 기준으로 해서 컴포넌트 분류항목을 다음과 같이 정의한다.

3.1 재사용 여부(Reuse Tag)

컴포넌트를 재사용할 것인지(1) 새로 개발할 것인지(0) 여부를 표시한다.

3.2 재사용 형태(Reuse Form)

재사용요소의 형태는 객체(Object), 클래스 라이브러리(Class Library), 디자인 패턴(Design Pattern), 프레임워크(FrameWork), 컴포넌트(Component) 등 5가지로 구분한다.

객체는 다른 프로그램 혹은 실행 코드의 일부로 실행되는 독립성과 자율성이 높은 단위이고, 클래스 라이브러리는 객체지향 프로그램에서 API와 같은 소스코드의 재사용이다. 디자인 패턴은 객체지향설계에서 자주 반복되는 구조인 추상화된 구조를 말한다. 프레임워크는 관련된 문제점들에 대한 추상설계를 가진 클래스들의 집합 또는 객체들의 집합을 말한다. 컴포넌트는 배포형태에 따라 설계 컴포넌트, 실행 컴포넌트, 원시코드 등이 있다. 실행 컴포넌트일 경우에는 구현언어를 CORBA, DLL, OLE/COM/ DCOM, ActiveX, Java Beans, C++, Java 등으로 구분해준다.

3.3 재사용 영역(Reuse Scope)

재사용 요소들의 분류는 컴포넌트의 기능에 따라 서로 연관이 있는 컴포넌트를 모아서 구분하는 것이 재사용을 쉽게 할 수 있다. 그러므로 컴포넌트의 재사용 영역은 기능에 따라 도메인, 기본 응용, 멀티미디어, 공통 상거래, DB 관련, 플랫폼 영역 등으로 구분한다[10].

3.4 재사용 레벨(Reuse Level)

재사용자에게 컴포넌트가 소프트웨어 개발 생명주기 중 어느 단계에서 재사용되는가를 알려주는 것이 필요하다. 예를 들면, 컴포넌트 개발과정중 컴포넌트 명세화 단계에서

컴포넌트가 발견되어 구현시에 코드를 재사용할 계획이라면 재사용할 단계가 구현단계임을 표시한다.

3.5 운영체제(Operating System)

최근 컴포넌트는 하나이상의 운영환경, 플랫폼을 지원하므로 Unix, Linux, Ms-dos, Window 98, Window 2000등의 복수의 운영체제를 표현할 수 있도록 한다.

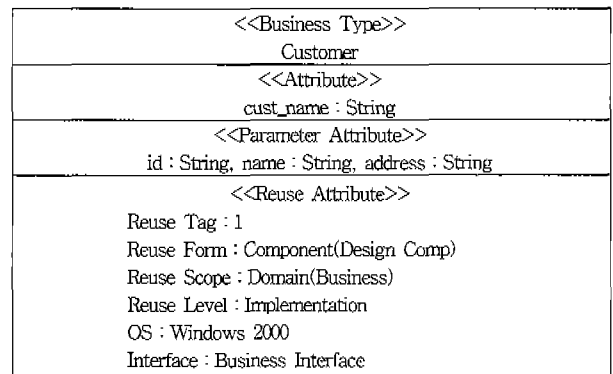
3.6 인터페이스(Interface)

컴포넌트는 인터페이스를 통해 다른 컴포넌트와 서비스를 주고받는다. 컴포넌트의 인터페이스는 비즈니스 인터페이스와 시스템 인터페이스로 구분한다. 비즈니스 인터페이스는 컴포넌트나 응용 객체들과의 상호동작을 명세하고, 시스템 인터페이스는 컴포넌트가 실행되는 플랫폼과의 상호동작을 명세하는 것이다.

<표 1>, <표 2>, (그림 1)는 추가된 UML 표기법, 재사용 속성에 사용되는 항목들과 비즈니스 타입 모델의 클래스 Customer모델링을 나타낸다.

<표 1> UML 표기법의 추가

의 미	표시형태	스테레오 타입
비즈니스 개념	클래스	<<Business concept>>
비즈니스 타입	클래스	<<Business Type>>
일반속성	오퍼레이션	<<Attribute>>
매개변수속성	오퍼레이션	<<Parameter Attribute>>
재사용 컴포넌트 속성	오퍼레이션	<<Reuse Attribute>>



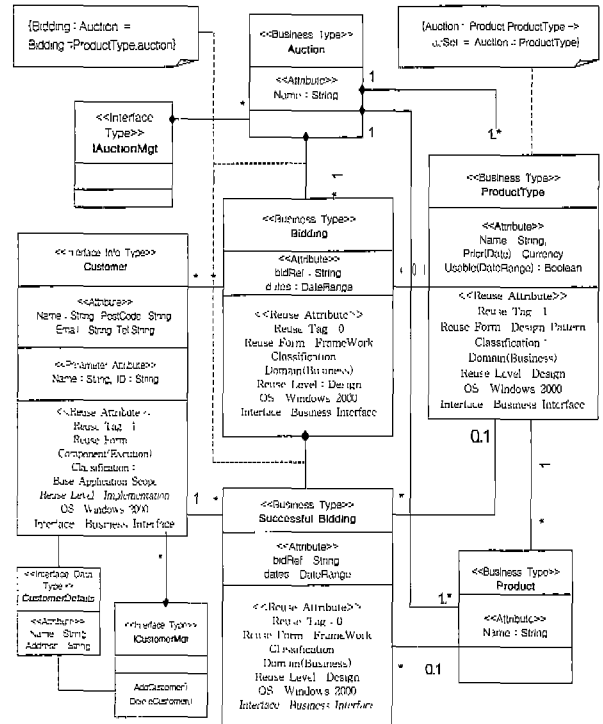
(그림 1) Customer <<Business type>> 클래스의 모델링

4. 컴포넌트 모델링(사례연구)

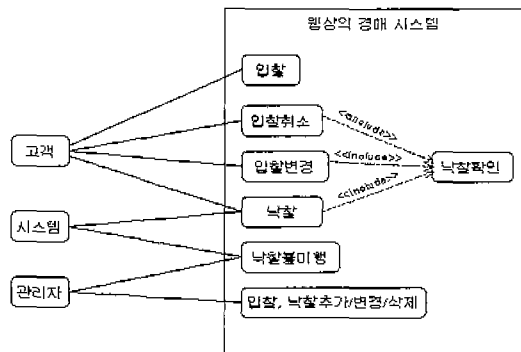
이장에서는 웹상의 경매 시스템을 사례연구로 하여 확장된 UML 표기법을 적용하여 컴포넌트 의 개발과정을 모델링해본다. 경매 시스템에서는 입찰, 입찰취소, 입찰변경, 낙찰, 낙찰불이행, 입찰 및 낙찰 추가/변경/삭제 등의 6가지의 이벤트를 가정해볼 수 있다. 이 시스템의 유즈케이스 다이어그램은 (그림 2)과 같다.

〈표 2〉 <<Reuse Attribute>>의 항목들

분류항목	세부 사항 표시	
Reuse Tag	제사용가능(1), 개발(0)	
Reuse Form	Object	DLL, Java Class code ...
	Class Library	MFC, Java AWT 등의 Library ...
	Design Pattern	추상화된 구조
	FrameWork	클래스들의 집합 또는 객체들의 집합
	Component	Design Component, Executable Component, Source Code ...
Reuse scope	Domain Scope	특정응용분야의 기능 컴포넌트(고객관리 등)
	Base Application Scope	공통적인 기능을 제공하는 컴포넌트(GUI등)
	Multimedia Scope	멀티미디어 컴포넌트
	Common E_Business Scope	E Business 컴포넌트
	DB Scope	Table 등
	Platform Scope	OS나 Network환경을 제공하는 컴포넌트
Reuse Level	Requirement, Anaysis, Design, Implementation Level	
Operation System	Unix, Linux, Ms-dos, Window98, Window2000등	
Interface	Business Interface, System Interface	



(그림 3) 경매시스템의 비즈니스타입 모델링에 적용된 UML 표기법



(그림 2) 웹상의 경매 시스템의 유즈케이스 모델

컴포넌트 명세의 산출물인 비즈니스 타입 모델에서 컴포넌트의 속성을 확장된 UML 표기법으로 (그림 3)와 같이 표기했다.

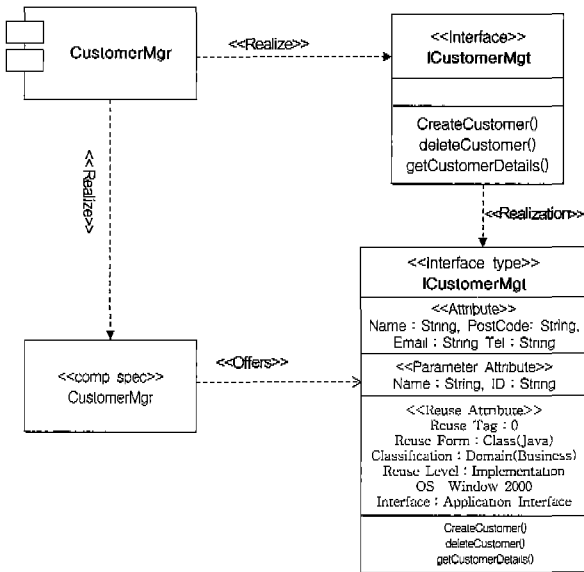
제약조건의 기술은 “{}”표시를 이용하여 OCL (Object Constraint Language)[4]로 기술하였다. 고객관리를 위한 인터페이스는 ICustomerMgt라는 명세 패키지로 정의하였고, 이 패키지는 ICustomerMgt 인터페이스 타입과 연관된 Customer 인터페이스 정보 타입과 필요한 고객의 세부 정보사항 CustDetails 데이터 타입을 포함한다. 입찰 Bidding 비즈니스 타입과 낙찰 Successful Bidding 타입은 IAuctionMgt 인터페이스에 할당되어 경매 정보를 관리하게 한다.

ICustomerMgt는 Customer 인터페이스 정보 타입에서 고객마다 인스턴스들이 필요하게 되므로 고객의 추가 및 삭제기능을 제공하는 인터페이스이며, 많은 고객을 지원하는 인터페이스 내부의 데이터구조(‘CustomerDetails’ data type)를 가진다.

Bidding 비즈니스 타입은 IAuctionMgt 인터페이스에 의해 관리되고 있는 정보들과 관련이 있으며, 고객에 대한 참조정보를 가지고 있다. Bidding 비즈니스 타입과 Customer 인터페이스 정보 타입간의 관계는 화살표(←, ◆)로서 정보의 흐름관계를 표시하였다.

경매시스템 관련 컴포넌트를 구현하고자 할 때, 컴포넌트는 ICustomerMgt와 IAuction Mgt 인터페이스를 사용해서 구현한다는 것을 알 수 있다.

컴포넌트는 컴포넌트 명세를 실체화하며, 인터페이스는 인터페이스 타입을 실체화하는 과정에 컴포넌트 속성들을 추가한 관계는 (그림 5)와 같다. 컴포넌트와 인터페이스의 관계는 UML 컴포넌트인 CustomerMgr이 <<comp spec>> 클래스인 CustomerMgr와 <<Interfac>> 클래스인 ICustomerMgr를 실체화하므로 <<realize>>관계로서 표시하였다. <<comp spec>> 클래스인 CustomerMgr은 <<Interfac type>> 클래스인 ICustomer Mgt에 정보를 제공해야 하므로 <<offers>>로 나타내었다. <<offers>>관계는 구현단계에서는 인터페이스들의 실체화단계로 모델링된다.



(그림 4) 컴포넌트 명세의 실체화 모델링

5. 평가

사례연구와 기존의 UML 개발 방법론의 장단점을 <표 3>과 같이 비교하였다.

<표 3> 본 연구와 기존의 UML 개발 방법론의 비교

	Catalysis 방법론	Rational사 방법론	Cheesman & Daniels 방법론	본 연구 방법론
객체지향개발 방법지원	○	○	○	○
재사용 UML 표기	×	×	×	○
컴포넌트의 재사용단계	구현단계	구현단계	구현단계	분석/설계/구현단계
데이터모델링 표기	△	△	△	△
명세와 구현 단계 연결	×	△	△	○
시스템의 이해도	○	○	○	○
유지보수 용이성	○	○	○	○
산출물크기	보통	크다	크다	크다
명세표기법	UML	UML+RUP	UML+RUP	UML+RUP

객체지향 개발 방법은 기존의 방법론과 본 연구 방법론이 다 지원하고 있으며, 컴포넌트를 재사용하는 개발 단계는 주로 구현단계에서 이루어졌기 때문에 그만큼 컴포넌트를 개발하는 속도가 늦어졌다는 것을 알 수 있다. 데이터 모델링 표기는 불분명하게 표기 및 사용되어 있었다. UML 사용자 지침서에도 데이터 모델링이나 데이터 흐름에 관한

표기가 없기 때문이라고 여겨진다.

명세와 구현 단계의 연결은 명세화 단계에서 직접 구현 단계로 컴포넌트 개발이 이루어지는 것을 의미한다. 시스템의 이해도와 유지보수 용이성은 표준화된 UML 표기법을 사용하므로 다른 사용자가 쉽게 컴포넌트를 이해할 수 있고 유지보수도 쉽게 관리 가능하다.

컴포넌트의 각 개발단계에 재사용 관련 UML 표기법을 사용한다면 그만큼 산출물의 부피가 커진다는 단점을 가진다. 그러나 상용화되고 있는 재사용 컴포넌트일 경우에만 재사용 UML 표기법을 지원한다면 산출물 크기는 문제되지 않을 것이다.

6. 결론

본 논문에서는 컴포넌트 개발과정에서 생성되는 산출물에서 컴포넌트를 모델링 할 때 필요한 UML 표기법을 정의하고 표현하였다. 컴포넌트의 특성에 따라 일반속성, 파라메타 속성, 재사용 속성 등을 UML 표기법으로 나타내어 응용시스템 개발뿐만 아니라 컴포넌트 개발과정에서도 컴포넌트를 재사용 할 수 있도록 정의하였다. 사례연구는 웹상의 경매 시스템에 컴포넌트의 재사용성을 나타내는 UML 표기법의 확장과 모델링 예를 제시하였다.

향후 연구로는 UML 확장 표기법과 재사용 UML 표기법을 실제 컴포넌트 개발과정 및 응용 시스템 개발에 적용해보고 재사용 효과를 테스트해보고 평가해보는 것이 필요하다.

참 고 문 헌

- [1] Philippe kruchten, "Modeling Component Systems with the Unified Modeling Language," First Int'l Workshop on CBSE, in conjunction with ICSE'98, 1998.
- [2] 최성운, 홍선주, "CORBA 컴포넌트 모델의 분석 및 전망", 정보처리논문지, 제7권 제4호, pp.46-51, 2000.
- [3] Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modeling Language User Guide," Addison Wesley, 1999.
- [4] Object Management Group, "Object Constraint Language Specification," Version 1.1, <http://www-4.ibm.com/Software/ad/standards/ocl.html>, 1997.
- [5] Robert Orfali, Dan Harkey, Jeri Edward, "Instant CORBA," WILEY, 1997.
- [6] John Cheesman & John Daniels, "UML Components," Cool

Software Korea, 2001.

- [7] Ed Roman, "Mastering Enterprise JavaBeans and the J2EE," John Wiley&Sons, 1999.
- [8] D'Souza, "Objects, Components, and Frameworks with UML." Addison Wesley, pp.1-106, 1999.
- [9] Grady Booch, "Visual Modeling with Rational Rose 2000 and UML," Addison Wesley, 2000.
- [10] 한국전자통신 연구원, "2000년 실행과제 계획서(공용 컴포넌트 개발)", 한국전자통신 연구원, 2000.



김재생

e-mail : jskim@kimpo.ac.kr

1998년 경희대학교 전자계산공학과(학사)

1990년 경희대학원 전자계산공학과(석사)

1997년 경희대학원 전자계산공학과(박사)

1993년~1997년 경희대학교 전산공학과

강사

1998년~현재 김포대학 컴퓨터계열 조교수

관심분야 : 재상용, 컴포넌트 개발, 품질 메트릭, 전자상거래 구축 등