

# 메타데이터 기반 응용프로그램 생성기 설계

김치수\* · 오은진\*\*

## 요약

소프트웨어 개발은 보통 요구사항, 분석, 설계, 구현, 그리고 테스트의 5단계의 과정으로 이루어진다. 개발과정 중 시스템 설계의 빈번한 변화와 그에 따른 프로그래밍의 어려움으로 인한 시스템 설계와 구현 사이에 항상 불일치가 발생하게 된다. 본 논문에서는 시스템 설계와 구현 사이의 불일치를 줄이고, 소프트웨어의 개발이 신속하고 유연하게 되도록 비즈니스 로직을 인식하여 응용프로그램을 생성시켜주는 도구를 설계하였다. 응용 프로그램 생성기의 핵심 아이디어는 같은 영역에 있는 비즈니스 애플리케이션을 첫째, 공통의 비즈니스 로직과 화면표시 로직으로 나누어 인식한다. 둘째, 시스템 디자인을 지속적인 메타데이터로 다룬다. 셋째, 지속적인 메타데이터를 사용해서 요구사항에 맞는 비즈니스 애플리케이션을 구축하거나 커스터마이징한다.

## A Design of the Application Program Generator based on Meta-Data

Chi Su Kim\* · Eun Jin Oh\*\*

## ABSTRACT

Software development process consists of five phases : requirement, analysis, design, implementation and testing. There is almost always a gap between the system design and implementation stages, caused by a combination of the difficulty of programming and frequent changes on the system design. The goal of this paper is to reduce the gap between system design and implementation, and we design a tool producing an application program by recognizing business logic for a more rapid and flexible developing of software. The core idea of Application Program Generator is : firstly, to recognize that business application in the same domain share business logic and presentation logic ; secondly, to treat system design as persistent meta-data ; thirdly, to use the persistent meta-data to build or customize applications as required.

**키워드 :** 메타데이터(Meta-Data), 응용프로그램(Application Program), 생성기(Generator), 설계(Design)

### 1. 서론

현재 비즈니스 애플리케이션의 환경은 복잡하고, 다양하며, 빠르게 변화하는 현상을 보이고 있다. 이러한 현상에 능동적으로 대처하기 위해 소프트웨어 부품의 재사용을 기반으로 하는 CBD[1-2]나 객체지향 설계에서 반복되는 구조를 사용하는 디자인 패턴[3]과 같은 객체기술을 적극적으로 도입해서 소프트웨어를 개발하는 추세로 가고 있다.

객체기술이 소프트웨어 개발 생산성과 품질향상에 효과적인 개발 기법이기는 하지만 실제로 단기간 내에 고객의 요구에 따른, 혹은 비즈니스 도메인의 변화에 따른 애플리케이션의 개발이나 시스템 업그레이드 작업은 여전히 매우 어렵고 때로 위태로운 작업이 되기도 한다.

따라서 본 논문에서는 프로그래밍의 어려움과 시스템 요구사항의 빈번한 변화로 인한 시스템 설계와 구현사이의 문제점을 해결하기 위하여 시스템 설계로서의 비즈니스 객체

모델을 메타모델로 다루어 지속적인 메타데이터로 저장하고 관리하는 방법을 기반으로 저장된 메타모델로부터 구현코드를 생성시켜주는 도구를 설계하였다. 그 결과 환경의 변화나 요구사항에 맞도록 애플리케이션을 다시 작성하거나, 같은 영역의 시스템을 개발할 때 메타데이터로 저장된 객체모델을 사용함으로써 소프트웨어 개발의 생산성 향상, 개발 시간의 단축, 그리고 좀 더 쉽고 유연하게 소프트웨어를 개발할 수 있는 개발 방법을 제시하고자 한다.

일반적인 소프트웨어 프로세스 모델의 장점은 개발 단계가 구별되고 각 단계의 산출물이 분명하다는 것이다. 그런데 이 소프트웨어 프로세스 모델은 다음과 같은 단점을 갖고 있다.

- ① 분석과 설계의 결과가 직접적인 코드로의 어떤 도움도 제공하지 못한다. 이것은 잘 설계된 시스템이라 하더라도 시스템의 구현에 있어 프로그래머가 순수 시스템 설계를 컴퓨터 언어로 변환해야 한다. 그래서 코드 구현을 지원하는 몇몇의 케이스 도구가 도움을 주고 있지만 한 줄씩 코드를 써야 한다.

\* 공신회원 : 공주대학교 정보통신공학부 교수(공학연구원)

\*\* 준 회원 : 공주대학교 대학원 컴퓨터공학과

논문접수 : 2004년 8월 19일, 심사완료 : 2004년 10월 1일

- ② 특정한 비즈니스 영역에서는 잘 설계된 비즈니스 모델과 컴포넌트들이 이미 구축되어 있어서, 공통의 설계 프로세스 중에는 반복할 필요가 없는 것이 있다.
- ③ 시스템의 변화를 요구하는 요구사항이 있을 때, 재설계와 재구축의 이중 작업을 해야만 한다. 이것은 소프트웨어의 개발에 비효율적이다.

위의 단점을 극복하기 위해 본 논문에서 연구되어온 방안은 비즈니스 애플리케이션을 사용자 정의에 맞추면서 소프트웨어 개발자의 과중한 업무를 줄이고 소프트웨어 개발을 더 쉽게 만드는데 있어, 객체설계를 일관된 객체구현으로 활용하기 위해 다음과 같은 방법을 제안한다.

- ① 같은 영역에 있는 비즈니스 애플리케이션을 공통의 지속적인 핵심 비즈니스 로직과 화면표시 로직으로 나누어 인식한다.
- ② 비즈니스 애플리케이션의 시스템 설계를 메타모델로 추출해서 비즈니스 로직은 클래스 메타모델로, 화면표시 로직은 GUI 메타모델로 다른 후 지속적인 메타데이터로 메타데이터베이스에 저장한다.
- ③ 시스템 설계의 메타데이터를 편집할 수 있게 함으로써 비즈니스 애플리케이션의 커스터마이제이션에 초점을 맞추고 애플리케이션 커스터마이제이션을 데이터 입력 윈도우의 GUI 설계와 시스템 모델링과 연결한다.

## 2. 관련 연구

### 2.1 메타모델링과 메타데이터

본 논문에서 다양한 시스템 설계 모델을 정의하고 저장하는 전반적인 개념으로 사용되는 객체 모델링 기술을 표현하는 메타모델링의 의미는 모델의 모델을 만들기 위한 것이다. 모델을 만드는 항목은 상위 단계 모델의 항목으로 나타나는 인스턴스뿐만 아니라 템플릿의 특징을 가지고 있다.

예를 들면, 일반 객체 모델의 메타모델을 만들면 “meta”라는 단어는 메타모델의 항목에 접두사로 붙는다. 그래서 메타모델은 주로 meta-class, meta-attribute, meta-method, meta-relationship으로 쓰인다.

메타데이터에 대한 가장 단순하면서도 유용한 정의는 “데이터에 대한 구조화된 데이터”이다. 메타데이터란 어떠한 객체나 자원(물리적인 것이든 전자적인 것이든 상관없음)에 대한 서술 정보로 메타데이터라는 말 자체는 상대적으로 새로운 것이지만 그 바탕이 되는 개념은 수집된 정보들을 조직화하기 시작했던 것만큼 오래 되었다고 할 수 있다[4-6].

본 논문에서 시스템 설계 정보를 영구적인 데이터로 저장하기 위해서 사용되는 메타데이터는 데이터나 작업에 대한 속성이나 내용을 기술한다.

예를 들면, “Java class” 객체가 있을 때 객체의 메타데이

터는 메타클래스(Class name), 메타속성(Attribute), 메타행위(Method)같은 메타데이터를 사용해서 표현할 수 있다. 데이터와 관련된 것들은 객체 데이터의 메타데이터로 나타난다.

### 2.2 Rational Rose와 Borland JBuilder

Rational Rose는 거의 대부분의 UML의 주요특징을 지원하는 도구로 사용자에게 객체지향 분석, 설계, 모델링, 구현의 기능을 제공한다. 또한 Rational Rose는 시스템 모델링 기반으로 기본 코드를 생성할 수 있으며 round-trip engineering[7]을 지원해 분석과 설계에서 구현의 활동을 지원하고, 다시 분석할 수 있게 해서 프로젝트 라이프 사이클의 처리를 모든 팀 멤버에게 지원한다.

본 논문에서는 케이스 도구를 개발하는데 있어 Borland JBuilder를 주로 사용하는데 JBuilder는 Borland Software Corporation에서 만든 케이스 도구로 앞서가는 가장 포괄적인 시각화 반복 개발 환경(IDE ; Interactive Development Environment)으로 자바 애플리케이션을 구축하기 위한 도구의 하나로 알려졌다[8].

다른 강력한 IDE처럼, JBuilder도 하나의 도구에서 코드 편집, 코드 디버깅, 프로그램 테스트의 서비스를 제공해서 사용자가 자유롭게 편집-컴파일-테스트 사이클을 이동할 수 있게 한다. 또한, JBuilder는 사용자가 순수 GUI 설계를 할 수 있도록 해서 GUI 설계자 사이에서 가장 인기 있는 도구 중의 하나로 알려졌다.

### 2.3 Rational Rose와 Borland JBuilder의 문제점과 메타데이터

Rational Rose와 Borland JBuilder는 소프트웨어 개발을 향상시키는데 큰 기여를 했지만 특별한 목적으로만 사용되기 때문에 여전히 설계와 구현 사이의 문제점을 해결할 수는 없다. Borland JBuilder의 경우 코드 편집과 시각화 정의에 의한 GUI 윈도우 생성은 강력하지만 시스템 모델링은 다를 수 없다. 이와 비교해 볼 때 Rational Rose는 시스템 모델링엔 좋지만 GUI 환경을 다룰 수 없는 문제점이 있다.

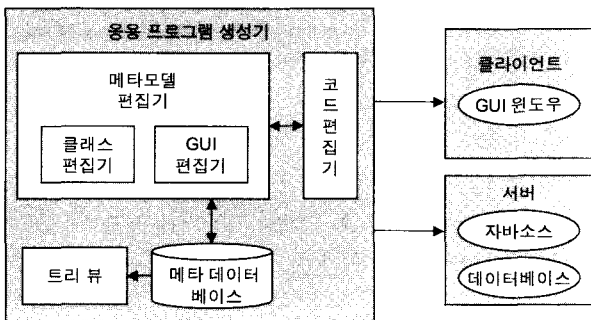
<표 1> Rational Rose와 Borland JBuilder의 비교

	지원 방법	적용 단계	특 정
Rational Rose	UML 중심의 객체 지향 모델링 도구	분석 설계	시스템 모델링 편집 시스템 모델링 생성
Borland JBuilder	통합된 개발 환경(IDE)지원	구현 테스트	소스코드 편집 GUI 윈도우 생성

본 논문에서는 이러한 문제점을 자원의 검색, 관리, 내용 등급, 보존, 저작권 관리, 전자상거래 등에 다양하게 활용될 수 있는 메타데이터[9]를 활용하여 시스템 모델을 구축하는 사례로 GUI 윈도우 생성과 시스템 모델링을 양쪽 다 고려하여 설계와 구현사이의 문제점을 해결하려는 시도로서 메타데이터베이스를 케이스 도구에 효율적으로 활용하는 특징을 가지고 있다.

### 3. 응용 프로그램 생성기 설계

본 논문에서 설계한 도구는 시스템 모델링, 시스템 GUI 설계, 코드생성, 데이터베이스 테이블 생성, GUI생성, 시스템 설계 편집 그리고 프로그램 코드 편집 등의 활동을 포함하는 순수한 자바기반 통합개발환경으로 “응용 프로그램 생성기”라 명명하였다. 이 도구의 목적은 비즈니스 애플리케이션에 대한 자바소스코드와 데이터베이스 테이블을 자동적으로 생성함으로써 본 논문에서 제시한 방법을 구현한 것이다.



(그림 1) 시스템 전체 구성도

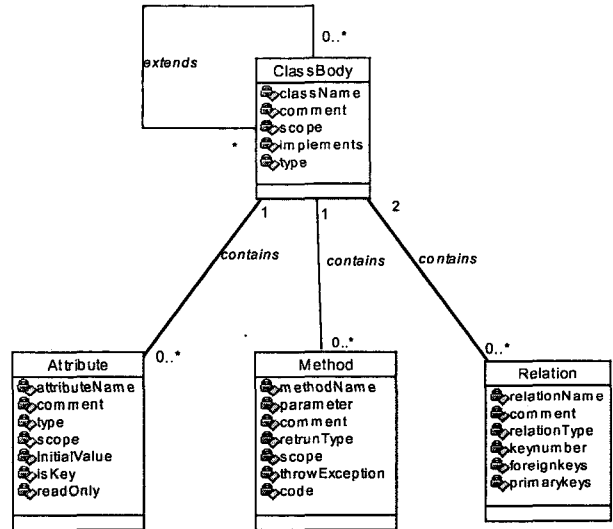
이 응용 프로그램 생성기는 다음과 같은 두 가지 핵심 부분으로 구성된다.

첫째는 시스템 설계에 대한 정보를 저장하는 데이터베이스로서 시스템 객체모델과 시스템 GUI 설계의 메타데이터를 저장하고 관리하는데 사용된다.

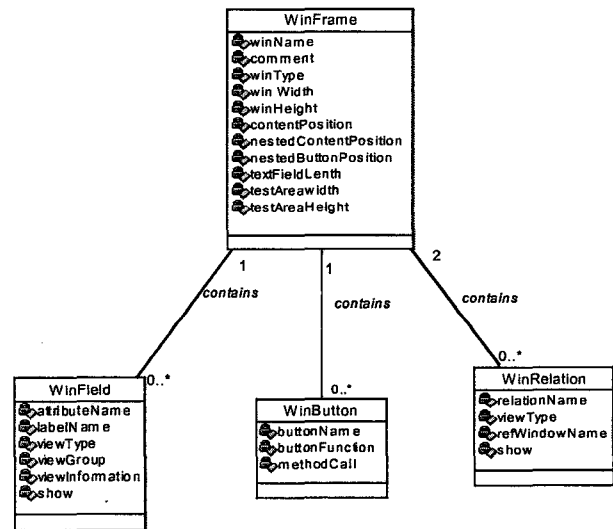
데이터베이스에 저장된 시스템 설계의 메타데이터는 두 번째 핵심 부분인 애플리케이션에 대한 소스코드와 데이터베이스 테이블 생성을 지원하는데 사용된다. 응용 프로그램 생성기로부터 생성되는 소스코드는 서버 측 모델에 대한 모든 자바소스코드인 비즈니스 서비스 클래스, 클라이언트 측 GUI 윈도우 클래스, 애플리케이션의 데이터를 저장하는데 사용되는 서버 측 데이터베이스 테이블, 다른 관련된 데이터베이스 관리 클래스들이다. 응용 프로그램 생성기의 GUI 에디터는 특별한 요구사항에 맞춰 시스템 설계를 하기 위한 메타데이터를 편집하는데 사용된다.

#### 3.1 메타모델 설계

응용 프로그램 생성기는 기능성을 고려하여 두 부분의 메타모델로 설계된다. 하나는 시스템에 대한 시스템 데이터 모델을 기술하는데 사용되는 것으로 클래스 메타모델로 불린다. 다른 하나는 시스템의 객체 모델을 기반으로 한 시스템에 대한 GUI를 기술하는데 사용되는 GUI 메타모델이다. (그림 2)는 클래스 메타모델을 보여주고, (그림 3)은 GUI 메타모델을 보여준다. <표 2>는 클래스 메타모델에서 각 클래스의 역할을, <표 3>은 GUI 메타모델에서의 각 클래스의 역할을 정리한 것이다.



(그림 2) 클래스 메타모델



(그림 3) GUI 메타모델

클래스 메타모델과 GUI 메타모델의 이름이 분리되어 있지만 응용 프로그램 생성기에서 각각은 서로 독립적이지 않고 결합되어 있다. 클래스 메타모델은 시스템 설계의 객체 모델링을 기술한다. GUI 메타모델은 객체 모델링을 기반으로 한 클래스 메타모델에 의해 표현된 GUI를 설계한다.

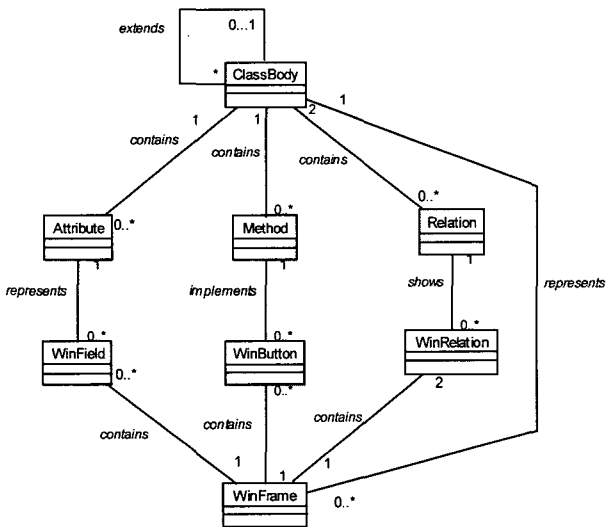
나누어진 메타모델은 (그림 4)와 같이 응용 프로그램 생성기의 완전한 메타모델로 함께 구성되어 사용된다.

<표 2> 클래스 메타모델에서 각 클래스역할

ClassBody	객체 모델에 있는 클래스에 대한 클래스의 이름이나 범위와 같은 일반 정보 정의
Attribute	속성의 이름, 타입, 초기값과 같은 클래스에 대한 속성 정의
Method	메소드의 이름, 요구되는 인수, 리턴 타입과 같은 클래스에 대한 사용자 정의 메소드에 대한 정보 정의
Relation	클래스는 두 클래스 사이의 One-One이나 One-many와 같은 객체 모델에 있는 클래스 사이의 관계 정보를 정의

〈표 3〉 GUI 메타모델에서 각 클래스의 역할

WinFrame	윈도우 크기나 레이아웃 같은 데이터 입력 윈도우 (Frame/Pane)의 전반적인 정보 정의
WinField	객체 속성이 데이터 입력 윈도우에서 어떤 항목으로 보일 것인가에 대한 정보 정의
WinButton	데이터 입력 윈도우에서 버튼이 어떻게 보일 것인가에 대한 정보를 정의, 어떤 액션을 취하는가에 대한 정보 포함
WinRelation	각각의 관련된 데이터 입력 윈도우와 어떻게 관련되는가에 대한 정보 정의



(그림 4) 응용 프로그램 생성기의 완성된 메타모델

3.2 데이터베이스 설계

데이터베이스 테이블에서 각각의 항목은 대응되는 클래스의 속성과 거의 똑 같다. 클래스의 속성과 테이블 항목의 차이점은 이름에서 다르다. 추가로 각각의 테이블의 기본 키와 외래 키가 정의될 필요가 있다. 다음 <표 4>는 메타데이터 테이블의 기본 키와 외래 키를 보여준다.

본 논문의 메타데이터베이스 설계의 장점은

- ① 메타데이터를 저장하는데 전체 8개의 테이블만 있다. 그 결과 데이터베이스 구조가 간단하고 분명하다.

- ② 각각의 테이블의 기본 키와 외래 키는 사용자가 요구하는 데이터를 빠르고 효율적으로 처리할 수 있게 해준다.
- ③ 데이터베이스 설계는 완벽하게 메타모델 설계와 대응하다. 따라서 시스템의 구현과 수정을 쉽게 한다.

3.3 응용 프로그램 생성기의 기능과 구조

응용 프로그램 생성기는 순수한 자바기반으로 설계된 것으로 시스템 모델링에서 자바코드 편집까지 수행한다. (그림 5)는 응용 프로그램 생성기의 주요기능을 나타내 주고 있는 것으로 응용 프로그램 생성기에 의해 제공되는 3개의 주요 특징이 있는데 트리 뷰, 모델 편집기, 소스코드 편집기이다.

트리 뷰는 사용자가 시스템에 대한 유용한 정보와 시스템에 이미 저장되어 있는 관련된 클래스를 빨리 찾을 수 있게 돕도록 설계되어졌다. 이 기능이 수행되어졌을 때 트리구조를 보여주게 되는데 기존에 있는 모든 기본 정보와 시스템에 정의된 클래스에 따른 속성, 메소드, 클래스사이의 관계를 보여준다.

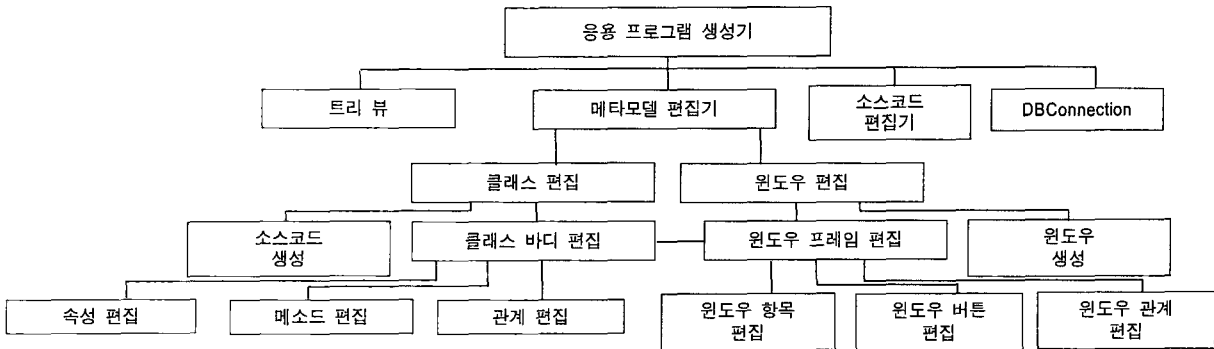
메타모델 편집기는 응용 프로그램 생성기의 주요 특징으로 시스템 설계의 메타데이터를 편집하고, 저장하며, 애플리케이션에 대한 코드를 생성하도록 설계되어져 있다. 메타모델 편집기는 두 부분으로 나누어지는데 클래스 메타모델 편집과 GUI 메타모델 편집이다. 클래스 메타모델 편집은 객체 모델링과 관련된 모든 프로세스를 다루는데 사용되고 GUI 메타모델 편집은 시스템 GUI 배치에 관련된 프로세스를 다루는데 사용된다.

소스코드 편집은 사용자가 시스템에 의해 생성된 자바 소스코드를 편집, 컴파일, 실행하는 기능을 제공하도록 설계되어진 특징이 있다.

클래스 바디 편집기는 시스템 객체 모델인 클래스 메타모델과 GUI 설계의 메타모델을 편집하는데 사용된다. DB-Connction은 응용 프로그램 생성기에서 JDBC를 사용해서 자바 클래스와 데이터베이스의 연결을 만들어주고, SQL 쿼리를 수행한다.

〈표 4〉 메타데이터 테이블의 기본 키와 외래 키

테이블 명	기본 키(Primary Key)	외래 키(Foreign Key)
ClassBody	className	
Attribute	className, attributeName	ClassBody table과 관련된 className
Method	className, methodName	ClassBody table과 관련된 className
Relation	className, relationName	ClassBody table과 관련된 className
Win Frame	className, winName	ClassBody table과 관련된 className
Win Field	className, winName, attributeName	WinFrame table과 관련된 assName과 winName Attribute table과 관련된 className과 attributeName
Win Button	className, winName, buttonName	WinFrame table과 관련된className과 winName
Win Relation	className, winName, relationName	WinFrame table과 관련된className과 winName Realtion table과 관련된 className과 relationName

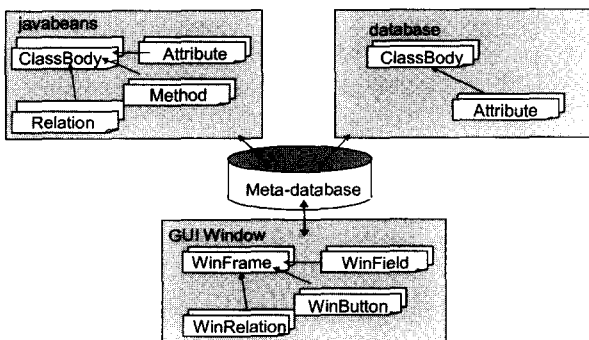


(그림 5) 응용 프로그램 생성기의 기능

3.4 코드 생성 설계

(그림 6)은 코드 생성 흐름도를 나타내 주는 것으로 소스 코드의 구현은 우선 설계 단계의 산출물인 UML의 클래스 다이어그램을 메타데이터 다룬게 되는데 클래스명은 Class-Body 객체로, 클래스의 속성은 Attribute 객체로, 클래스의 메소드는 Method 객체로 클래스의 관계는 Relation 객체로 추출되어 메타데이터 테이블에 저장한다.

저장된 타데이터는 소스 코드로 직접 매핑되는데 Class-Body 객체는 서버 측 자바 빈즈 클래스로 매핑된다. Attribute 객체는 자바 빈즈 클래스의 속성으로 getter/setter 메소드와 함께 매핑된다. Method 객체는 자바 빈즈 클래스의 메소드로 매핑된다. 자바 빈즈 클래스에서 ContainsOne 타입을 가지고 있는 Relation 객체는 Relation 클래스에서 관련된 클래스에 의해 정의된 속성으로 관련된 객체에 매핑된다. 자바 빈즈 클래스에서 ContainsMany 타입을 가지고 있는 Relation 객체는 Vector type 객체 속성으로 매핑된다.



(그림 6) 코드 생성 흐름도

ClassBody는 또한 애플리케이션을 생성하기 위한 서버 측 데이터베이스 테이블로 대응된다. 데이터베이스 테이블에서 Attribute 객체는 테이블의 항목으로 매핑된다.

WinFrame 객체는 클라이언트 측 자바 클래스로 매핑 되는데 JFrame, JDialog 클래스의 확장이다. 윈도우에서 WinField 객체는 TextField, JTextArea, JComboBox, JLabel을 가지고 있는 JCheckBox로 적절히 매핑된다.

윈도우에서 WinButton 객체는 JButton 으로 매핑된다. "TabbedPane" 타입의 뷰를 가지고 있는 WinRelation 객체

는 하나의 JFrame에 있는 하나의 탭 패인으로서 두개의 데이터 입력 패인으로 매핑된다. "List" 타입의 뷰를 가지고 있는 WinRelation 객체는 데이터 입력 윈도우에서 리스트의 항목을 처리하는데 사용되는 "nested button pane"과 함께 탭 패인 안에 객체들과 관련된 리스트와 매핑된다. 데이터 입력 윈도우에서 "Table"의 뷰 타입을 가지고 있는 WinRelation 객체는 관련된 객체의 테이블을 포함하는 탭 패인과 매핑된다.

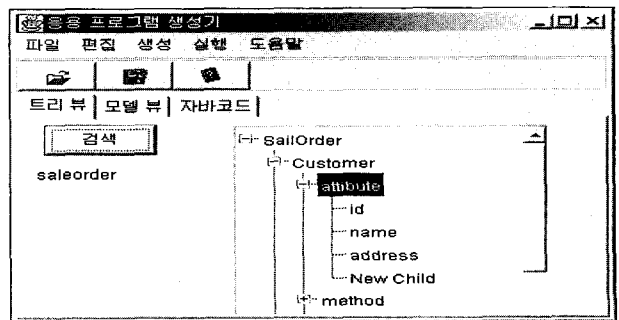
3.5 생성기 UI 설계

인터페이스 설계는 어떤 애플리케이션에서도 중요하다. 응용 프로그램 생성기에서 인터페이스 설계는 두 가지 주요 점을 고려해 수행되어졌다.

첫째, 도구의 인터페이스는 사용자에게 시스템 설계의 메타데이터를 직접 편집하는 방법을 제공할 것으로 기대된다.

둘째, 사용자에게 생성된 GUI가 어떻게 보이는가의 감각을 제공한다.

전형적인 GUI 윈도우는 시스템에서 메타데이터를 입력하거나 편집하는데 사용된다. 데이터 입력 윈도우는 응용 프로그램 생성기로부터 거의 비슷하게 만들어진다. 그러나 훨씬 더 보기 좋게 만들 수도 있다. 메타모델 설계를 보면, GUI 메타모델 설계는 주로 도구의 GUI 설계 아이디어를 기반으로 한다. (그림 7)은 응용 프로그램 생성기의 대표적인 메인 화면을 보여준다.



(그림 7) 응용 프로그램 생성기의 메인화면

3.6 비즈니스 애플리케이션 UML 모델링

응용 프로그램 생성기는 시스템 설계를 기반으로 만든 비즈니스 애플리케이션의 실행 가능한 자바소스코드와 데이터베이스 테이블을 생성한다.

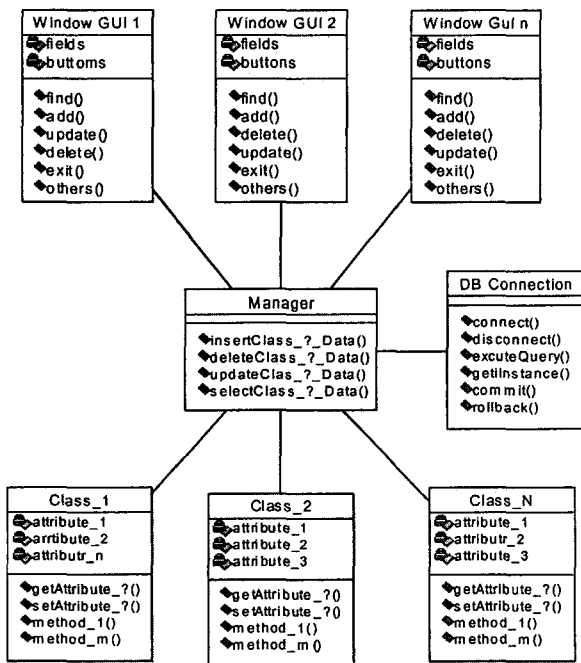
응용 프로그램 생성기에 의해 생성되는 클래스와 테이블은 애플리케이션의 중요한 부분을 구성한다.

(그림 8)은 생성된 애플리케이션의 UML 모델링을 보여준다. 서버 측 비즈니스 서비스 클래스는 전형적인 자바 빈즈로 속성 집합을 포함한다. 각각의 속성에 대한 getter/setter 방식과 사용자가 애플리케이션을 정의하는 방식을 포함하는 것이다.

클라이언트 측 GUI 윈도우 클래스는 JTabbedPane, JPanel, JScrollPane, JList, JTable, JTextField, JTextArea, JComboBox, JCheckBox, JButton으로 구성된 전형적인 자바 Swing JFrame/JDialog 클래스이다. 서버 측 데이터베이스 테이블은 레코드의 내용이 없는 구조적 테이블이다. 데이터베이스 관리 클래스와 그것의 보조 클래스는 JDBC의 기술을 사용한다.

4. 결론 및 향후 연구과제

시스템 설계의 빈번한 변화와 프로그래밍의 어려움으로 인한 설계와 구현사이의 문제점을 해결하기 위해 비즈니스 애플리케이션을 핵심 비즈니스 로직과 화면표시 로직으로 나누어 인식하였다. 또한 이를 같은 시스템 설계에서도 공유할 수 있도록 지속적인 메타데이터로 저장한 후, 저장된 메타데이터를 직접적인 구현에 사용하였다. 따라서 요구사항에 맞는 애플리케이션을 구축할 수 있도록 함으로써 애플리케이션의 개발을 좀더 간단하고, 빠르며, 유연하게 할 것으로 기대된다.

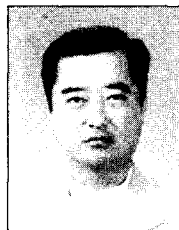


(그림 8) 비즈니스 애플리케이션 생성기 UML 모델링

향후 연구과제로 비즈니스 모델을 보다 더 총체적으로 파악하는 방법, 시스템에 대한 GUI를 생성하는 방법, 생성된 애플리케이션을 어떻게 보이게 할 것인가의 방법 등을 향상시키기 위한 구체적인 연구가 필요하며, 완전한 코드생성 방법의 연구가 계속되어야 한다.

참고 문헌

- [1] Bradford, K., "Software Components as Application Building Blocks," at URL : <http://www.quoinc.com/quoinc/ComponentsABB.html>, 1998.
- [2] 이회락, "게임 사이트 구축을 위한 컴포넌트 설계", 공주대학교 석사학위 논문, 2003.
- [3] ShallowayA., James R. T. Design Patterns Explained., Addison Wesley, 2002.
- [4] 한국 더블린 코어 메타데이터, "메타데이터(matadata)란 무엇인가?", at URL : <http://www.dublincore.or.kr/faq.htm>, 2001.
- [5] e-Government Interoperability Framework, <http://www.govtalk.gov.uk>.
- [6] Resource Description Framework, <http://www.w3.org/RDF>.
- [7] 플라스틱 소프트웨어, "소프트웨어 모델링도구선택", at URL : [http://agora.plasticsoftware.com/UMLKoea/View.aspx?brd=umlk\\_storage&pn=0&n=56000](http://agora.plasticsoftware.com/UMLKoea/View.aspx?brd=umlk_storage&pn=0&n=56000).
- [8] 볼랜드 자바팀, JbuilderStudyNet 공저, 최고의 자바개발 솔루션 JBuilder7, 가남사, 2002.
- [9] 국가지식정보통합검색시스템, <http://www.knowledge.go.kr>.



김치수

e-mail : cskim@kongju.ac.kr

1984년 중앙대학교 전자계산학과(학사)  
 1986년 중앙대학교 전자계산학과(석사)  
 1990년 중앙대학교 전자계산학과(박사)  
 1990년~1992년 공주교육대학교 전임강사  
 2001년~2002년 University of Auckland, NZ 방문교수

1992년~현재 공주대학교 정보통신공학부 교수  
 관심분야 : 객체지향 분석 설계, CBD 방법론



오은진

e-mail : oej73@kongju.ac.kr

2001년 한국방송통신대학교 컴퓨터학과 (이학사)  
 2004년 공주대학교 컴퓨터공학과 (공학석사)  
 2004년~현재 공주대학교 컴퓨터공학과 (박사과정)

관심분야 : 소프트웨어 공학, CBD 방법론