

# UML을 이용한 동적 문제출제시스템의 설계

박운영, 김윤호  
안동대학교 컴퓨터공학과

## A Design of Dynamic Test Generating System Using UML

UnYoung Park, YunHo Kim  
Dept. of Computer Engineering, Andong National Univ.  
E-mail : highland@comeng.andong.ac.kr

### 요 약

인터넷 사용자의 급속한 증가와 통신망의 고속화는 교육 분야에 있어서도 많은 변화를 가져왔다. 최근 주목을 받는 분야 중에 하나는 인터넷 기반의 가상대학이다. 가상대학에서 해결해야 할 중요한 부분은 학생의 수학 능력을 평가하는 시스템이다. 본 논문에서는 UML를 이용하여 자동화된 문제출제시스템과, 출제된 문제를 학습자에게 출력하는 시스템을 설계하고 구현한다. 편리한 사용자 인터페이스를 가지고, 멀티미디어를 가미한 문제 출제 시스템으로의 확장성을 제공하는 동시에 현재 강의 시스템과 병행하여 최대한의 교육 효과를 내는데 그 목적이 있다.

### 1. 서론

인터넷 사용자의 급속한 증가와 통신망의 고속화는 교육 분야에 있어서도 많은 변화를 가져왔다. 최근 주목을 받는 분야 중에 하나는 인터넷 기반의 가상대학이다. 가상대학에서는 교수와 학생간의 교수와 학습, 평가와 응시가 원격으로 이루어지게 된다. 현재의 동향은 주로 원격 강의 시스템에 치우쳐있다. 가상대학에서 해결해야 할 또 다른 중요한 부분은 학생의 수학 능력을 평가하는 시스템이다. 현재 가상대학 중 시험시스템에서는 문제출제를 위해 웹 페이지를 작성하는 형태를 띠고 있고 웹 페이지의 작성은 문제 출제의 번거로움을 야기시키게 된다. 문제출제에 관련된 기존 연구들은 Script Language 나 HTML 로 문제출제시스템을 구성함으로써 사용자 인터페이스부분에서 취약한

측면을 가지고 있으며 출제되는 문제들의 경우 멀티미디어 요소를 가미할 수 없다[ 1, 2, 3]. 본 논문에서는 자바를 이용한 자동화된 문제 출제 시스템과, 출제된 문제에 대해 학습자가 응시할 수 있는 평가 시스템을 UML(Unified Modeling Language)를 사용하여 객체지향 방법론에 따라 설계하고 구현하였다.

### 2. 문제출제시스템의 객체지향분석 및 설계

본 논문에서는 UML을 이용하여 향상된 사용자 인터페이스를 가지는 문제출제시스템을 객체지향에 의거하여 분석 및 설계하였다. 문제출제시스템에 대한 Use Case를 모델링하고 Use Case 모델을 기반으로 하여 각 Use Case에 대한 Interaction Diagram을 작성한다. 클래스를 모델링하기 위해서는

Use Case 모델을 사용하여 후보 클래스들을 식별해내고 Interaction Diagram을 분석하여 세부적인 클래스들을 설계해 나간다. 클래스의 식별과 세부 설계가 완료되면 식별된 클래스들 중 영속성을 가지는 클래스들에 대해 데이터베이스로 매핑시키는 과정을 거친다.

## 2. 1 Use Case 모델링

Use Case 모델은 개발 생명 주기(Development Life-Cycle)에서 중요한 역할을 하게 된다. Use Case 모델은 분석모델(Analysis Model)에 의해 구조화되고, 디자인 모델(Design Model)에 의해 구체화되며 구현 모델(Implementation Model)을 생성하는데 중요한 바탕이 된다. 또한 테스트 모델(Test Model)에서도 사용될 수 있다. Use Case 모델링은 현재 대부분의 객체지향 개발 방법론에 있어 시작점이 되고 있다. Use Case는 시스템 전체나 시스템부분의 행위를 명세하고 변형이 포함될 수도 있는 순차적으로 발생하는 활동의 집합을 기술한다. Use Case를 구성하기 위해서 시스템과 상호 작용하는 Actor를 식별해낸다. 객체지향 분석의 다른 측면들처럼 Actor들은 초기 반복 시에는 쉽게 식별되지는 않는다. 초기 반복 시에는 요구된 시스템의 기능을 달성하고 시스템으로부터 의도한 이익을 유도하기 위해 상호 작용하는 Actor들을 식별해내고 이후 여러 번의 반복을 통해 앞서 식별된 Actor들의 작업을 할 수 있게 시스템을 지원하는 Actor들을 식별해 낸다.

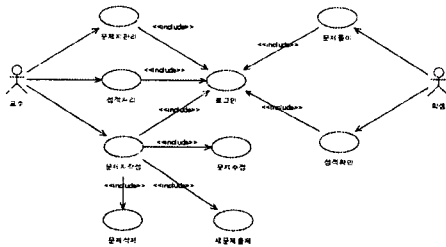


그림 1. 시스템 Use Case Diagram

그림1은 문제출제시스템의 교수준 Use Case

Diagram이다. 교수준 Use Case Diagram은 시스템의 외부 인터페이스를 고려한 전체 시스템의 상황을 이해하는데 목적이 있다.

시스템의 사용자로 교수와 학생이 있고 시스템이 교수에게 제공하는 기능으로써는 문제지관리, 성적처리, 문제지작성이 있고 학생에게 제공하는 기능은 문제풀이와 성적확인이다. Use Case Diagram에서 문제출제시스템의 사용자인 교수와 학생을 Actor로, 각 Actor에 제공하는 인터페이스들을 Use Case로 식별한다. 따라서 문제출제시스템의 Actor는 교수와 학생으로, Use Case는 문제지관리, 문제지작성, 성적처리, 문제풀이와 성적확인으로 식별된다.

Use Case 간의 포함 관계는 기본 Use Case에서 다른 Use Case의 행위를 명시적으로 수용하는 것을 의미한다. 포함된 Use Case는 그것을 포함한 기본 Use Case의 일부로만 인스턴스를 만들 수 있고 독자적으로는 만들지 못한다. 포함관계를 사용함으로써 똑같은 사건 흐름을 여러 번 설명하지 않아도 되고, 단지 공통 행위를 자신의 Use Case에 포함시키면 된다. 그림1에서 로그인 Use Case는 각 Actor의 Use Case마다 반복적으로 수행되는 사건의 흐름으로 독립된 Use Case로 분리시켜 각 Use Case에 포함되도록 하였다. 문제지작성 Use Case에는 로그인 Use Case이외에 새문제출제, 문제수정, 문제삭제 Use Case를 포함하고 있다.

Use Case를 명세화 하려면 이벤트의 흐름을 문장으로 분명히 설정하여 외부인이라도 쉽게 알 수 있게 해야 한다. 이벤트의 흐름을 작성할 때는 Use Case가 언제 어떻게 시작되고 끝나는지를 포함해야 한다. 그림1과 같이 Actor와 Use Case들이 식별되면 각 Use Case별로 이벤트의 흐름을 기술한다. Use Case별 이벤트의 흐름을 기술할 때에는 Use Case Name, Use Case에 연관되는 actor, 이벤트의 흐름과 오류에 대한 처리 등을 포함한다. 먼저 시스템에 접근할 때 가장 빈번하게 사용되는 로그인 Use Case에 대한 이벤트의 흐름은 표1과 같다.

Use Case Name	로그인
Actor	교수, 학생
Flow of Events	M-1 사용자(교수, 학생)가 로그인 메뉴를 선택하면 Use Case가 시작된다. M-2 시스템은 로그인 대화상자를 출력한다. M-3 사용자는 사용자 이름과 비밀번호를 입력한다. M-4 시스템은 사용자로부터 입력된 정보를 확인한다. M-5 시스템은 사용자에게 시스템 접근 권한을 부여한다. M-6 Use Case를 종료한다.
Alternative Flows	E-1 잘못된 사용자이름 입력하였을 때 E-2 잘못된 비밀번호 입력하였을 때

표 1. 로그인 Use Case 에 대한 이벤트 흐름

로그인 Use Case에 관련되는 actor는 교수와 학생이다. 이벤트의 흐름에서는 Use Case의 시작 조건과 종료조건을 명시한다. 로그인 Use Case는 사용자가 로그인 메뉴를 선택하였을 때 시작되고 시스템이 사용자에게 시스템에 대한 접근 권한을 부여하였을 때 종료된다.

Use Case Name	문제지작성
Actor	교수
Flow of Events	M-1 교수가 시스템을 실행하면 Use Case가 시작된다 M-2 교수가 로그인 메뉴를 선택하면 로그인 Use Case가 시작된다 M-3 시스템은 교수가 출제 가능한 문제지들을 출력한다. M-4 교수는 출제 가능한 문제지들 중 하나를 선택한다. M-5 시스템은 교수가 선택한 문제지의 문항들을 출력한다. M-6 (교수가 종료를 선택할 때까지 반복 수행) 교수가 종료 선택하지 않고 새문제출제를 선택하면 새문제출제 Use Case가 수정하고자 하는 문제를 선택하고 문제수정을 선택하면 문제수정 Use Case가. 삭제하고자 하는 문제를 선택하고 문제삭제를 선택하면 문제삭제 Use Case가 시작된다. M-7 교수가 종료 를 선택하면 Use Case가 종료된다.
Alternative Flows	E-1 출제 가능한 문제지가 없을 때 교수에게 메시지 출력

표 2. 문제지작성 Use Case에 대한 이벤트 흐름

표2에서는 교수와 연관되는 문제지 작성 Use Case에 대한 이벤트의 흐름을 기술하고 있다. 문제지작성 Use Case에서는 M-2단계에서 로그인 Use Case가 포함되어 있음을 나타내고 있고 M-6단계에서는 교수의 선택의 반복 과정에 대해 나타내고 있다. 교수가 종료를 선택하기 전까지 새문제출제 Use Case, 문제수정 Use Case, 문제삭제 Use Case를 교수의 선택에 의해 반복적으로 수행한다.

## 2.2 동적 모델링

객체지향 설계 과정에서는 객체지향 분석을 사용하여 생성한 분석 모형을 소프트웨어 구축의 청사진이 되는 설계모형으로 변환시킨다. 객체지향 설계의 첫번째 과정으로 객체지향 분석 시에 작성된 각 시나리오에 대한 Interaction Diagram을 작성한다

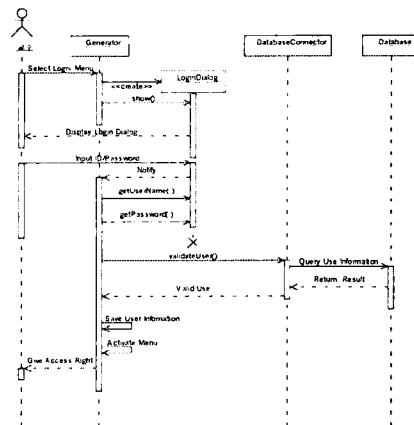


그림 2. 로그인 Use Case의 Sequence Diagram

그림 2에서는 로그인 Use Case에 대한 Sequence Diagram을 보여준다. 로그인 과정을 시작하는 교수와 Generator<sup>1</sup>, LoginDialog, DatabaseConnector Database 클래스의 객체 사이의 메시지 전달을 시간 순서에 따라 보여준다.

문제지작성 Use Case에 대한 Sequence Diagram은 아래 그림3과 같이 나타난다.

<sup>1</sup> Georgia체는 문제출제시스템에서 사용되는 클래스, 패키지명을 나타냄.

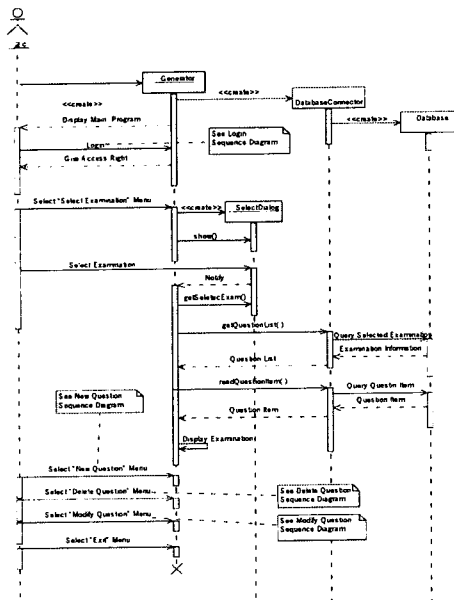


그림 3. 문제지작성 Use Case의 Sequence Diagram

문제지작성의 첫번째 단계는 사용자의 로그인에서 시작된다. 다음 단계에서는 DatabaseConnector는 데이터베이스에 사용자에게 대한 질의 수행 후 그 결과를 돌려 받게 된다. 이 결과에 따라 사용자가 선택할 수 있는 과목명을 사용자에게 출력해주게 된다. 사용자는 출력된 과목 중 하나를 선택하고 Generator는 해당되는 문제들을 데이터베이스로부터 읽어와 사용자에게 출력한다. 사용자는 새로운 문제를 출제하거나 기존 문제를 삭제하거나 수정할 수 있다.

Use Case에 대한 분석이 충분하게 수행되면 결과로 생성된 클래스를 아키텍처 설계에서 사용한다. 아키텍처 관점에서 중요한 클래스를 식별하고 설계 패키지나 서브시스템으로 그룹화하는 일도 수행한다. 그런 다음 서브시스템을 알맞은 계층으로 조직화한다. Use Case 명세를 분석한 결과로부터 설계자는 클래스의 특징과 관계를 식별하고 클래스를 더 자세히 정의한다. 이 과정에서 설계자는 아키텍처의 설계를 이용한다. 또한 서브시스템 인터페이스를 정의한다. 서브시스템의

인터페이스는 서브시스템에 포함된 클래스의 설계를 참조하여 만들어진다. 그런 다음 Use Case를 다시 완전하게 설계한다. Use Case는 Sequence Diagram이나 Collaboration Diagram에서 클래스 또는 서브시스템간의 상호 연산을 통해 완전하게 명시된다. 시스템이 대규모의 데이터를 포함하고 있을 때에는 데이터베이스 설계자는 persistent 클래스를 데이터베이스의 테이블에 할당하는 등의 작업을 아울러 수행한다.

### 2.3 클래스 모델링

클래스 모델링의 목적은 클래스를 분석해내고 분석해낸 클래스들의 속성과 클래스들 사이의 상호관계를 찾아내는 것이다. 일반적으로 클래스들을 식별해내기 위해 앞에서 구성된 Use Case 모델을 바탕으로 하여 CRC(Class Responsibility Card)와 같은 방법들을 사용하는데 Use Case의 이벤트 흐름으로부터 클래스들을 식별해낼 수 있다. 개발자는 각 Use Case의 이벤트의 흐름을 주의 깊게 분석하여 Use Case에서 중요한 역할을 하는 항목들을 식별해 낸다. 클래스를 식별해내는 것은 모든 Class Diagram과 Object Diagram을 작성하는 바탕이 된다. 클래스를 식별해 내는 여러 가지 방법들이 있지만 Use Case 명세를 이용해서 클래스를 식별하는 것이 가장 손쉬운 방법이다.

각 이벤트 흐름에 대한 interaction 다이어그램이 작성되면 객체지향 분석에서 추출한 클래스들에 대해 세부적인 디자인을 해야 한다. 객체지향분석 단계에서는 후보 클래스와 각 클래스의 속성만을 식별한다. 클래스와 클래스의 메서드는 객체지향설계 과정에서 클래스 다이어그램에 삽입된다. 시스템의 동작(Action)은 시나리오 별 Interaction diagram을 분석함으로써 결정된다. 클래스의 세부적인 설계와 관리의 용이성을 위해 각각의 클래스들을 generator, viewer, DatabaseHandler, examination 이라는 네 패키지로 나누었다.



클래스로 구성되어 있다. 데이터베이스 연결을 위해 Database와 Results라는 두 개의 클래스를 FACADE Pattern을 사용하여 구현하며 DatabaseConnector 클래스가 두 개의 클래스를 사용하도록 구성하였다.

### 2. 4 Database 설계

클래스 모델링의 결과로 식별된 클래스들 중 영속성을 가져야 하는 클래스들에 대한 데이터베이스로의 매핑을 수행한다. 일반적으로 데이터베이스는 Entity-Relationship diagram으로 표현한다. E-R diagram은 데이터베이스의 데이터 자체에만 중점을 두고 표현하는데 반해 UML의 Class Diagram에서는 trigger나 stored procedure와 같은 오퍼레이션도 표현 가능하다. 데이터베이스는 UML의 component view에서 컴포넌트로 사용될 때 <<Database>> 스테레오타입을 사용하여 표현한다. 또한 데이터베이스 내부의 schema를 표현하기 위해서는 <<Schema>> 스테레오타입을 가지는 패키지를 사용하여 나타낸다. 데이터베이스 schema에서의 관계형 테이블은 <<Table>> 스테레오타입을 가지는 클래스로 표현한다.

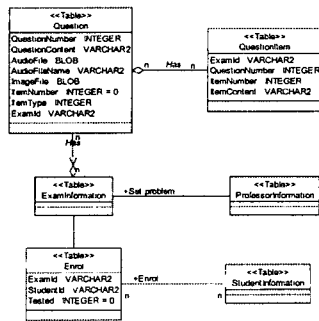


그림 9. 데이터베이스 테이블

그림9는 데이터베이스 테이블에서는 문제출제시스템에서 사용되는 테이블들을 나타내고 있다. 각각의 클래스들은 <<Table>>스테레오 타입으로 표기하여 데이터베이스 테이블임을 표현하고 있다. Question이라는 테이블과 QuestionItem이라는 테이블은 포함관계를 나타내고 있고 n:n 관계를 나타내고 있다. 테이블과의 연관관계도 클래스 다이어그램의 여러 가지

연관관계를 사용해 표현한다. 교수와 학생에 대한 정보를 저장하는 클래스인 ProfessorInformation과 StudentInformation 클래스를 동일한 이름을 가지는 테이블에 대응시킨다. 시험지의 정보와 시험지에 속하는 문제와 각 문제의 객관식 항목을 저장하는 ExamInformation, Question, QuestionItem 테이블을 설계한다. Question 테이블에는 문제 출제 시 그림을 추가시키기 위한 필드가 존재한다.

### 3. 결론

본 논문에서는 UML를 이용하여 자동 문제출제시스템을 설계하고 구현하였다. 본 문제출제시스템은 문제 출제 과정의 번거로움을 없애고 손쉽게 대량의 문제를 출제할 수 있는 방법을 제공한다. 문제 출제 시스템에서 자바를 이용함으로써 유연한 사용자 인터페이스를 구현하였으며 다양한 동영상, 사운드, 음성 지원 등과 같은 멀티미디어적 요소를 추가시키기 위한 확장성을 제공한다.

### [참고문헌]

- [1] 오민권, 윤여창, 웹을 이용한 온라인 학습 평가 시스템, 한국정보처리학회, 2000 추계학술발표논문집 제7권 제1호, 2000
- [2] 박기석, 이재영, 김동한, 자바를 이용한 웹 기반의 문제 출제 시스템, 한국정보과학회, '99 가을 학술발표논문집, 1999
- [3] 최돈은, 서현진, 박기석, 이재영, 동적인 문제 출제 시스템의 설계 및 구현, 한국정보과학회, 2000 추계학술발표논문집, 2000
- [4] Stephen R. Schach, "Classical And Object-Oriented Software Engineering With UML And Java", McGraw-Hill, 1999
- [5] Grady Booch, James Rumbaugh and Ivar Jacobson, The Unified Modeling Language User Guide, Addison Wesley, 2000
- [6] Grady Booch, Object-Oriented Analysis And Design With Applications, Benjamin/Cummings Publishing Company, 1994