

Design-First Approach 와 UML이용 (A Case Study on the Design-First Approach and using UML for a Video Rental Store)

국중각(Joong-Kak Kook)¹⁾

<요약>

객체지향기술을 이용하여 소프트웨어를 개발할 때 소위 “큰 그림” (big picture)으로 나타내는 “design-first” approach 방식으로 채택하는 하는 경향이 있어 관심을 끌고 있다. 본 연구도 이 방식을 채택하여 video rental store(VRS) system을 개발하였다. 이 system은 효과적으로 모델링(modeling) 할 수 있도록 UML을 이용하였으며, PHP script, MySQL Database, Java를 함께 사용하였으며, 실습 프로젝트(project)로 구현하였다.

단계별 프로세싱을 통해 학생들이 프로그래밍을 작성하는데 원칙을 따르도록 지도 하였으며, 코딩(coding)을 시작하기 전에, 계획, 분석, 그리고 설계를 이행하도록 지도했다. 학생들이 가능한 실제 세계와 같은 경험을 가지도록 노력했다. 그리고 전반적으로 그 개발과정은 긍정적이었다.

<abstract>

In this paper, teaching object-oriented design in a programming course is a big challenge for instructors. “Objects-first” is a popular strategy for teaching object-oriented programming in the educational settings, by introducing the concepts of objects, classes, and instances before procedural elements of a programming language. Still, this approach emphasizes coding rather than other critical aspects of software development, notably problem-solving and design. We propose a “design-first” approach, which subsumes an objects-first approach into lessons that also introduce object-oriented analysis and design, using elements of UML before implementing any code. We also present video rental store (SRS) that uses the design-first approach to help students of various learning styles in a software engineering course. It interfaces with an IDE we have chosen specifically to support the design-first approach, and SRS which has been shown to be effective in helping students learn object-oriented programming concepts.

논문접수 : 2006. 12. 28.

심사완료 : 2007. 1. 20.

1) 정회원: 삼육대학교 컴퓨터학부 부교수

1. 서론

객체지향 프로그래밍(object-oriented programming: OOP)이 10여년 이상 교육계에 도입되었지만, 아직도 논쟁이 존재하고 있다. 언제(when) 그리고 어떻게(how) 그것을 교육할 것인가에 대한 issue중의 하나이다. Bruce Kim(2004)는 이 course를 교육하는데 강사가 접근할 수 있는 3가지 방안을 제시하였다.[12] 1안) “object-later” approach: 객체(objects)를 후에 배우라. 학생들이 절차적인 언어 요소(constructs, loops, array,..)를 배운 후에 다음으로 객체를 배운다. 2안) “object-first” approach: 객체를 먼저 배우라. 예컨대, BlueJ, DrJava, 또는 MicroWorld 와 같은 교육학적인 tool을 이용하여, 먼저 객체를 배운다. 3안) 최초의 코스로 절차적인 언어 또는 함수 언어를 먼저 배우라. 그리고 2년에 가서 객체지향 언어를 공부하라. 이들 중 어느 안을 채택하든, 객체지향의 기초와 튼튼한 이해가 컴퓨터전공을 하는 학생들의 계속적인 전공 공부에 중요하며, 그리고 학생들은 처음 무엇을 배웠느냐에 따라, 가장 잘 배울 수 있다고 우리는 믿고 있다.

상기 2안인 “object-first” approach에 대하여 언급하고자 한다. 이 “object-first” approach의 주목할 점은 문제해결(problem solving) 기술에 대하여 상대적으로 약하게 강조하고 있다는 점이다. 따라서, 이 부분을 보완 해 줄 수 있는 기술을 산업계의 표준이 된 UML요소가 합류하게 되었다. 간단히 말하면 <“object-first” + UML>이 새롭게 나타난 경향은 “design-first” approach 방식으로 나타났다.[1][7].

이 “design-first”는 소프트웨어 개발에서 coding 보다는, 우선 먼저 전체를 보는 “big pictures”로 소프트웨어를 개발하는데 큰 그림을 강조하고 있다[12]. 이 design-first approach는 object-first approach의 내용을 상당 부분 수용하고 있다. 객체지향에서 classes, constructors, and methods를 어떻게 사용하느냐를 배우는 것이다. 즉, 고객의 문제

(customer’s problem)를 어떻게 이해하고 해결하느냐 이다. 그들은 역시 빠르게 고객의 문제를 분석하기 위하여 use case를 어떻게 발전시키느냐를 배우며, 그리고 또한 UML(Unified Modeling Language)에서 class diagram 어떻게 설계하느냐를 배우는 것이다.

본 연구는 이 design-first approach 방식을 채택하여, 비디오대여 시스템(VRS System)을 통해 이 주제에 접근하고자 한다. UML은 물론, 개발 툴(tools)로는 Visio2005, EditPlus, BlueJ, PHP, MySQL Databse, Java, 등등..으로 개발환경을 갖춰 작업을 수행 하였다. 학생들은 각 단계별로 작업을 가졌으며, 실습 프로젝트로 이루어졌다.

2. 소프트웨어공학 교육의 예로점

기초과정에서 다수의 학생들이 C++로 또는 Java로 객체지향 프로그래밍(Object-oriented programming: OOP)을 등록하여 공부하고 있다. 비록 객체지향 프로그래밍에 대하여 초기 단계에서 공부함이 필요하지만, 프로그래밍을 배우는 학생들에게 다음과 같은 어려움을 겪고 있다:

- 학생들은 프로그래밍을 배울 때 문제를 분석하고 설계를 통한 해결 방안이 강조되어야 하나, 이 보다는 그 언어의 syntax에 초점을 맞추고 있다.
- 학생들이 문제해결과 코딩사이에서 연결(관계)을 만들 수 없기 때문에, 자주 동기를 잃게 되고, 일부의 학생은 급기야 다른 전공을 생각하게 된다.
- 소프트웨어 설계와 프로그래밍 개념에 대하여 기초적인 이해가 없는 학생들은 더욱 복잡한 프로그래밍 코스를 접하게 될 때 매우 어려운 심적 갈등을 겪고 있다.

- 고급수준 프로그래밍 코스(courses)에 등록한 학생은 예컨대, 일부 과목(compiler construction, computer graphics, .. 등등)은 코스 지원 자료를 공부하는 대신, 코스 프로젝트를 수행하는데 코딩 시간을 지체하여 더 많은 초과 시간을 사용하고 있다.

이러한 문제는 학생들이 소프트웨어 공학 코스를 등록하여 수업을 받게 될 때에, 나타나고 있는 현상이다. 그리고 이러한 현상은 OOP에 대한 수년간의 경험과 더불어 감지되고 있다. 학생들은 프로그래밍 개념과 문제해결 능력의 부족으로 인해 프로그래밍을 하는데 불편하고, 자주 고충을 갖고 있다. 주로 프로그래밍을 공부할 때 syntax에 많은 시간을 보내게 되어, 정작 중요한 기본 내용에는 충실하지 못한 경우를 보게 된다.[3],[4]

또 하나의 주요한 점을 들자면, 돈만 많다고 화려한 가정이 되는 게 아닌 것처럼, 코딩(coding)만 잘한다고 질 좋은 소프트웨어를 개발되는 것은 아니다. 오히려, 코딩 이전의 분석과 설계단계를 어떻게 진행하느냐에 따라 프리젝트(project)의 성패가 갈리는 경우가 경험상 보아 왔다.

객체지향 코스는 교과과정에서 객체지향분석과 설계(object-oriented analysis and design: OOAD)에 대하여 설명 강조하더라도, 특별한 준비 없이 학생들에게 객체지향 프로그래밍(object-oriented programming: OOP) 개념을 일반적으로 소개하고 있다. OOAD(object-oriented analysis and design)와 방법론(methodologies)은 일반적으로 고급 프로그래밍(higher-level programming) 코스나 소프트웨어공학(software engineering)코스로 미루게 된다.[3] 심화수준 코스에서 OOAD의 개념을 소개할 때에, 대부분의 학생들은 어떤 문제에 대하여 초기단계의 분석과 설계 보다는, 구현단계에 와서야 분석과 설계를 발전시키는 경향이 있다. 우리는 이러한 상황을 관찰할 수 있었다. 일부의 학생들은 문제를 분석할 때 중요한 이슈(issues)들을 보통 간과(看過)하는 경

향이 있다. 이는 불완전 하거나 정확하지 않은 설계나 이행을 가져오게 된다. 사실, 많은 학생들은 분석테크닉을 적용하는 것은 “시간낭비”이다 라고 생각하고 있다. 결과적으로 학생들은 해결책을 위하여 개념적 모델이 되는 생각과 문제에 대한 이해, 그리고 설계를 받아들이는데 어려움을 겪고 있다.[6]

3. 개발환경 및 개발도구

본 연구는 상기에서 언급한 것과 같이, design-first approach 방식을 따르며, 이 연구는 소프트웨어 공학을 수강하는 학생들에게 학기 초부터 UML을 이용한 OO응용을 통하여 다음과 같은 기초를 준비하였다:

- 객체(object)를 이용하고, 가시적인 툴(visual-based tools)을 이용하여, 소프트웨어 개발을 보이도록 시도했다.
- 학생들의 나름대로의 발전을 통해, 문제해결을 위해서 객체와 툴(tools)을 응용함으로써 실행하였다.

이들 두 가지는 학생들이 실수를 하거나, 또 실수를 했을 때 그 원인을 파악하고 그리고 어떻게 그것을 정정해서 해결해 가느냐의 과정을 배우게 되는 것이다. 말하자면, 준비된 사항들이 실수를 바로 잡기 위한 자연적인 방법을 준비하고 있다.

개발환경 및 개발도구는 Window XP 또는 Windows NT Server 2000 운영체제에서 Web Server인 Apache Server에 PHP Script와 MySQL database server를 탑재하였다. 개발 툴로는 EditPlus와 Visio2005, BlueJ (version 2.0), 그리고 Java (JDK 1.5.0 version) 사용하였다.

이 실습 프로젝트의 제한적인 범위와 UML에 대하여 사전 제시를 통해, 학생들에게 기초 준비를 통해 모델링과 개발과정에 맞게끔 준비했다.

4. 프리젝트 진행 절차

이 프리젝트는 소프트웨어 공학에서 실습 프리젝트(비디오 대여가게 - video rental store : VRS system)로 진행되었다. 시작하기 전에 전체적인 계획과 단계별 작업에 대하여 전반적인 이해를 구하였다. 응용프로그램 구축 과정은 마치 물리적인 건물 구조물을 세우는 것과 같이 비교 되고 있다. 집을 건축하는 것처럼, 소프트웨어를 개발에서도 유사한 점을 탐구해 볼 수 있다. 학생들은 요구사항들을 수집하고, 모형(prototype 또는 mock-up)을 생성하며, 비용예측을 계산하며, 그리고 모든 요구사항을 문서화(creating documents) 단계에 대하여 배우도록 맛을 보았다.

다음으로, 학생들은 객체(objects), 클래스(classes), 그리고 인스턴스(instances)를 먼저 개념적으로 배우고, 그때 좀 더 정확하게 Java 안에서 보충하였다. 화포에서 서로 다른 그림을 그리는 것처럼, 한 쌍의 클래스를 이용하여, 인스턴스를 생성하며, (데이터를) 다루며, 메시지를 호출하는 등등.. 일련의 연습을 마치게 되었다.

다음으로, 학생들은 클래스가 무엇인지, 그것을 어떻게 사용하는지 이해하고 있다면, 그들은 제로(0)에서 소규모 응용프로그램을 구축하고 설계 문제를 접하게 되었다. 우리는 프리젝트인 비디오 가게(VRS)의 설계과정을 소개하였다. 우리는 사용자(user)의 요구사항과 문서화를 먼저 이해하도록 하였으며, 그 중요성을 강조하였다. 학생들은 그 VRS와 서로 대화하는 actors 들을 확인하며, use case들을 확인하고, 그리고 use case diagrams을 그리게 된다. 그리고 그때 선택 대안을 포함하여 각 use case 별로, 그리고 단계별로 문서화를 하게 되었다. use case를 끝낸 학생만이 그 다음 단계로 진행하도록 지도하였다. 그 점에서 그들은 필요한 객체를 설계하고, 각각의 속성(attributes)과 오퍼레이션(behavior)을 정의하며, 클래스 다이어그램(class diagram)을 생성하게 되었다. 학

생들은 IDE(Integrated Development Environment) 개발환경에서 class diagram에 들어가, class, attribute, method, ..을 작성하기 위하여 Java 코딩을 하게 되었다. 다음으로, variables, data type, String, assignment, statement, 그리고 loops, .. 등등 같은 기본적인 코딩 개념을 소개하며, 각 개념을 표현하도록 제안하게 할 때, 학생들은 바로 그 VRS에 따른 코딩을 하게 되었고, 메서드도 작성하게 되었다. 그 project의 거의 마지막 단계에서 학생들은 그 시스템의 출력 형태인 웹형태 출력 방법을 배우게 되었다.

이 project에서는 최소의 강사 도움과 함께 자신들의 작업으로 마치도록 지도 하였다. 학생들은 여러 툴들을 사용하였으며, 단계별로 작업을 마쳤다.

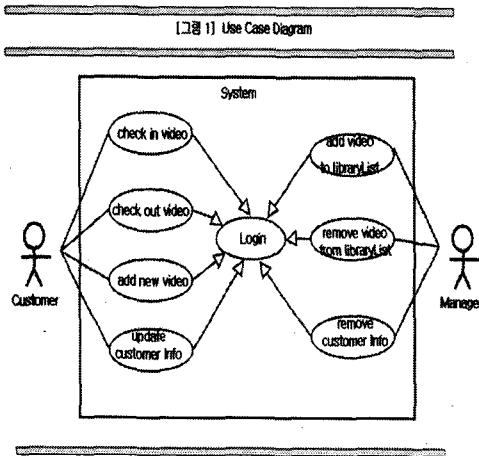
5. 분석과 설계 및 구현

실습 project수행의 성공을 가져오기 위하여 전체적인 프로젝트 원칙과 규칙을 이해하도록 구했다. 학생들은 프리젝트를 수행한 경험이 거의 없다. 따라서, 산업체와 같은 환경과 시간과 같은 조건을 가지지 않지만, 유사한 경험을 갖도록 노력하였다. 어떤 추가적인 process를 시작 하기 전 시스템에 대한 필요성, 목적, 이점, 그리고 범위 등을 묘사하는 project 계획에 대하여 토의를 가졌다.

먼저, 계획(planning) 단계의 목적은 새로운 시스템의 목적을 확인하고, 그 project가 쉽게 수행하도록 확인하는 것이고, 자원계획(resource), 예산, 그리고 스케줄(schedule)을 발전시키는 것이다. 반면에 다른 단계는 다음 단계가 될 때까지 use case를 작성하기 위하여 다음 단계까지 기다리며, 선택하는 것이다. 프리젝트 범위를 설정하는데 그룹 멤버를 통하여 어떤 use case의 다이어그램을 작성하는데 도움이 된다는 것을 알게 된다. [그림 1]에서 본 use case diagram은 비디오대여(video rental store)의 응용프로그램 기능을 기술하고 있다. 말하자면,

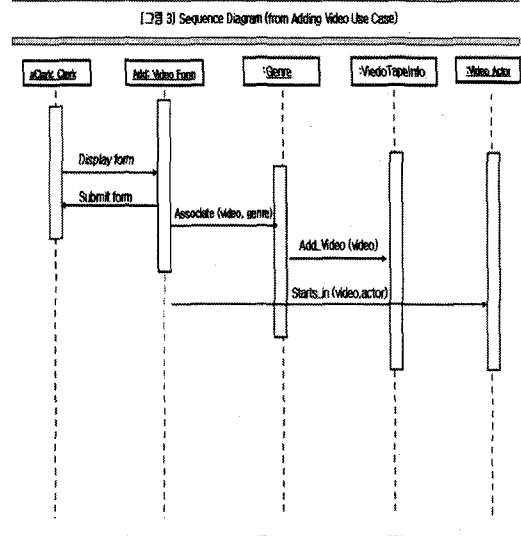
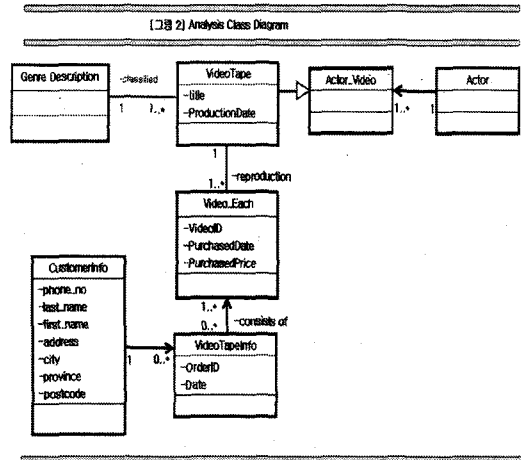
그 시스템의 특징들을 개별적으로 상세히 나타내기 보다는, 이용할 수 있는 기능들(서비스)을 보여주고 있다. use case diagram은 단지 4개의 주요 요소를 가지고 있다: 시스템과 대화하는 actors, system 그 자체, 그 시스템이 제공하는 service, 그리고 이들 요소들 간에 relationship를 나타내고 있다.

다음 단계인 분석단계는 개발자가 새로운 시스템의 business 필요성과 요구사항(requirements)을 이해하고 문서화 하여야 한다. 그 사항(requirements)을 설정하는 동안 그 주된 초점은 class, attributes, 그리고 그 classes들과 서로 간의 관계(relationship)를 확인하는 것이다. 대응하는 class diagram이 [그림 2] 에서 작성되었다면, 각각의 use case에서 기술한 actions을 나타내는 sequence diagram을 그리기 시작한다.



각각의 use case는 일치한 하나의 sequence diagram을 가지며, 그 sequence diagram의 actors는 그 use case로부터 온 것이다. 우리의 목적에 대해서, 우리는 그 sequence diagram에서 3개의 component로 제한하였다: actors, classes, 그리고 user interface (UI). actors는 실제로 키보드를 터치하는 사람들을 말하며, 또는 UI를 통해 activity를 시작한 사람들이

다. data 위주의 웹응용 프로그램에 대해서, sequence diagram에 있는 class 객체는 [그림 3] 에서처럼, database로 또는 database 부터 class 이동의 예이다. class 이름은 database의 table 이름과 같다. 학생들은 일련의 sequence diagrams을 사용함으로써 그리고 각 use case 대해 하나씩 사용함으로써 actors, interfaces, 그리고 data 가운데 서로 대화를 상세히 기술할 수 있어야 한다.



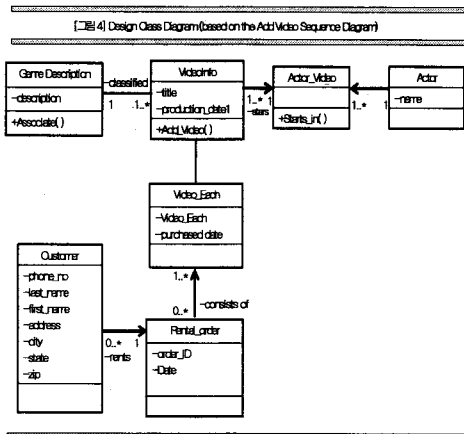
design. 해결책은 이미 정의한 요구사항 (requirements)을 분석하는 동안 만들어진 결정사항에 대하여 기초를 두고 있다. 분석 서류 (analysis documents)와 함께, 배달된 설계는 바로 직전 단계의 세공품을 발전시키는 것이다. sequence diagram에서, 모든 message는 소스(source) 객체와 도착지(destination) 객체를 가지고 있다. 어느 메시지 하나가 객체(object)로 보내질 때, 그 객체는 그 메시지를 허가하기 위하여 준비되어야만 하고, 또 몇 가지 활동을 시작해야 한다. 이 프로세스는 단지 객체에 대한 메서드를 요청하거나 간청한 하는 것이다. 다시 말하면, sequence diagram 에서 나타난 모든 message는 도착지(destination) 객체에서 메서드를 요청하고 있다. 설계 클래스 (design class)를 만들기 위하여, 학생들은 모든 sequence diagram을 통과 해야 되고, [그림 4]에서 처럼 각 class로 보낸 모든 메시지를 찾아야 한다. 이들 message들은 각각의 class를 위하여 요청한 메서드를 결정해야만 한다. 유사하게, 그 message가 사용자(user) interface로 보내진다면 PHP script는 요청한 action과 일치한 method를 포함하여야 한다. 직접 class map으로부터 SQL에 이르기 까지 대부분의 actions은 statement를 선택하거나 삽입하게 된다.

이 프로젝트 process 중에서 어려운 부분은 단계별 과정을 거쳐 끝까지 implementation 단계 까지 가도록 학생들에게 요청하였다. 분석과 설계단계가 끝나기 전에, coding을 시작할 수 있다면 시간을 절약할거라고 대부분 학생들은 생각하고 있다. 왜냐하면, 거기에는 각 단계들 간에 약간 중복되어 있고, 또 coding를 시작하기 원하면 학생들은 외부의 user interface에 초점을 맞춰야 한다고 생각하기 때문이다. User Interface는 sequence diagram을 개발하는 동안 그리고 요구사항(requirement) 사항들을 설정하는 동안 윤곽을 그려 낼 수 있다. 대부분의 경우에, interface는 시스템설계 (system design)가 진척될 때 몇 가지 반복적으로 경험을 하게 된다.

6. 요약(summary)

지금까지 우리는 학기 중에 이루어지는 하나의 실습 프로젝트로 이루어졌다. design-first approach 방식으로 주어진 VRS(비디오 대여가게) System을 구현하는데 단계별로 언급하였으며, 교육과 함께 각 단계의 공적작업의 원칙과 내용을 함께 언급하였다. UML을 이용하고 design-first approach 방식에 충실히 따르도록 시도하였다.

design-first approach는 학생들에게 소프트웨어 개발을 할 때 큰 그림("big picture")을 소개해줄 수 있기 때문에. 구조적인 프로그래밍 또는 object-first approach 보다는 더 효과적이다 라고 믿고 있다. 물론 학기중에 이루어지기 때문에 시간 제한과 또 소화내야 할 내용이 많기 때문에 애로점이 많았다. 무엇보다도 시간제약이 크다. 그러나, 가능한 소프트웨어 공학적인 접근을 통해서 VRS system을 구현할 때 생략부분도 있었지만, 분석, 설계, 등등.. 단계별로 학생들과 공정을 수행해 나갈 있도록 노력하였다. 특히, design-first approach 방식에 충실하면서, 소프트웨어 공학적인 접근을 충실히 가지려고 시도하였다. 학생들의 수행과



정은 그 완성보다는 그 과정 원칙을 충실히 지키며 순서를 지키도록 노력하였다. 또 project 경험이 없는 학생들이 주류를 이루기 때문에 현실 세계와 같은 감각을 느끼도록 하자는데 하나의 목적이기도 하였다.

이 paper는 하나의 완성된 구현 단계만을 중요시 하거나 또는 그 결과만을 보고 판단하도록 강조한 아니라, 그 수행 해 가는 각 공정을 충실히 지키면서 수행 과정에 주안점을 주었다. 따라서 만족할 만한 결과는 가져오지 못했지만, 그 과정에서의 교육은 긍정적인 효과를 가져왔다.

7. 결론

이 paper에서, 객체지향(object-oriented: OO) 분석과 설계의 기본개념을 학생들에게 조기에 적용할 필요성을 가졌으며, 학생들의 설계기술을 증진하는데 도움이 되었다.

학생들이 이행하는데 필요한 설계부분과 프로그래밍 작업부분 사이에 링크(link)를 눈으로 볼 수 있다면, OO 개념의 훌륭한 이해가 성취될 수 있을 것으로 보았다. 이러한 시각을 성취하기 위하여, 우리는 문제기술(problem statement)을 프로그래밍 작업(programming assignment)으로 변형시키는 설계에 기초를 둔 방법론을 더욱 더 발전시키면 design-first approach 방식과 같이, 더욱 더 높은 수준의 소프트웨어공학 코스를 교육하는데 도움이 될 것으로 본다.

틀은 기본적인 개념을 이해하는데 학생들을 도와주고 있다. 이 design-first approach에 기초를 둔 VRS는 여러 틀(tool)들에 의하여 작업을 하게 되었고, 학생들에게 그 개념을 이해하도록 도와 주었으며 설계와 프로그래밍 문제를 응용하도록 도움을 주었다.

가시적인(visual) 분석과 설계 지원은 모호한 문제 진술 (problems statement)로부터 소프트웨어 개발 작업을 지원 하도록 소프트웨어공학을 공부하는 학생들에게 도움을 주었다. 더욱

기 학생들은 작업한 코딩을 본인이 설명할 수가 있었으며, 또 하나는 소프트웨어 개발과정에서 상호 의사소통을 할 때 학생들이나 강사간에 선택적인 대안 또는 구체적인 대화를 나누게 될 때 통신방법에도 도움을 주었다.

참고문헌

- [1] Moritz, S. H., Wei, F., Parvez, S. M., and Blank G. D. From Object-First to Design-First with Multimedia and Intelligent Tutoring, Proceedings of Innovation and Technology in Computer Science Education (ITiCSE), 2005, Lisbon, Portugal June, 2005.
- [2] Moritz, S. H. and Blank, G. D. A Design-First Curriculum for Teaching Java in a CS 1 Course, SIGCSE Bulletin (inroads), vol.37, 2, July 2005.
- [3] Tabrizi, M.H.N., Collins, C., Ozan, E. and Li, K. Implementation of Object-Orientation Using UML in Entry Level Software Development Courses. Proceedings of the 5th conference on Information technology education October 2004.
- [4] DeLooze, L. L. Minimal UML Diagrams for a Data-Driven Web Site. Proceedings of the 6th conference on Information technology education SIGITE '05, October 2005.
- [5]. Barnes, D. & Kolling, M. Objects First With Java: A Practical Introduction Using BlueJ, Englewood Cliffs, NJ: Prentice Hall, 2002.
- [6] Hamza, H. S. From Problem Analysis to Programming Assignments: Teaching the Big Picture in Early Object-Oriented Courses. Journal of Computing Sciences in Colleges, 2004.
- [7] Moritz, S. H., Wei, F., Parvez, S. M., and Blank G. D. A "Design-First" Curriculum an

d Eclipse Tools. Journal of Computing Sciences in Colleges, Volume 22 Issue 3, January 2007.

[8] Ghafarian, A. Teaching design effectively in the introductory programming courses. Journal of Computing Sciences in Colleges, Volume 16 Issue 2 November 2000.

[9] Moderator, F. B. Objects-first - Does It Work? 2003. Journal of Computing Sciences in Colleges, 2003.

[10] Wei, F., Moritz, S. H., Parvez, S. M., Blank, G. D. A Student Model for Object-Oriented Design and Programming, Consortium for Computing Sciences in Colleges (CCSCNE), Providence RI, April, Online at <http://www.cse.lejgh.edu/~cimel/papers/CCSCNE05.pdf>. 2005

[12] Bruce, Kim. Controversy on How to Teach CS1: A Discussion on the SIGCSE-members Mailing List. In inroads - The SIGCSE Bulletin, December, 2004.

국 중 각

- 현재: 삼육대학교 컴퓨터학부 부교수
- 미국 오레곤 주립대학교 졸업(University of Oregon, Eugene, OR, U.S.A),
진공: 컴퓨터교육 및 소프트웨어공학, Ph.D.
- 교환교수: 미국 Southern Adventist University, Collegedale, TN).
- 미국 휴스턴 주립대학교(University of Houston), 연구원 및 조교수
- 관심분야: Software Engineering, Software Modeling,
Object-Oriented Programming, Mobile and Internet-Based Learning.