

UML을 적용한 모바일 고객관리 시스템 설계 및 구현

심 갑 식[†] · 정 태 영^{††}

요 약

최근 웹 기술이 급속하게 발달함에 따라 웹 기반의 많은 어플리케이션이 개발되고 있지만, 웹 어플리케이션의 개발 방법론 및 품질개선 속도는 이를 따르지 못하고 있다. 대부분의 웹 어플리케이션의 생성은 체계적인 개발 방법론 없이 개발자의 지식과 경험에 의존하여 개발되고 있다. 그러나 웹 기반의 응용시스템은 다양한 개발 기법이 필요하며 설계 및 개발 프로세스를 위해 풍부한 개발 방법론이 요구된다. 본 논문은 객체지향 개발 방법론을 지원하는 UML(Unified Modeling Language)을 모바일 고객관리 시스템 설계 및 구현에 적용하였다. 즉, 웹 개발단계에서 필요한 스토리 보드와 요구사항 기술서를 바탕으로 시스템을 모델링하는데 UML을 적용하였다. 그리고 본 논문에 적용된 개발 방법론을 통해서 산출된 컴포넌트들은 이와 유사한 웹 어플리케이션 개발 시에 필요한 컴포넌트로 재 사용될 수 있다. 이 시스템에 UML을 적용한 결과 효율적인 웹 어플리케이션의 분석 및 설계가 이루어졌다.

Design and Implementation of a Mobile Customer Management using UML

Gab-Sig Sim[†] · Tae-Young Jung^{††}

ABSTRACT

Recently many applications are being developed as web technologies are evolved rapidly, but the development methodology and quality improvement speed of the web applications is not so. The creations of Most web applications have developed depending on the developer's knowledge and experience without a systematic development methodologies. But the web-based application systems require the various developing techniques and the rich developing methodologies for a design process. This paper applies the UML (Unified Modeling Language) supporting an object-oriented development methodology to the design and implementation of a mobile customer management system. That is, it applies the UML to the system modelling based on a story board and a requirement specification at the web development step. And the produced components through this development methodology can be reused at the development of the similar web applications. Applying UML to this system achieved the analysis and the design of an web application efficiently.

키워드: UML, Story Board, Use Case Diagram, Activity Diagram

1. 서 론

최근 웹의 짧은 역사에 비하여 웹 관련 기술의 급속한 발전으로 웹 어플리케이션의 효율적인 개발방법이 필요하다[1-3]. 1990년 중반 이후, 정보시스템의 새로운 패러다임으로 등장한 웹 시스템은 메인 프레임 및 클라이언트/서버 환경을 대체하면서 빠른 기술적인 발전을 하고 있다. 웹 아키텍처와 비즈니스 로직을 구현하는 기술 및 프로그래밍 언어, 웹 페이지를 디자인하는 그래픽들이 빠른 발전을 하는 반면에, 개발 라이프 사이클을 제시하고 경로를 구성하는 프로세스 또는 방법론에 대한 논의는 상대적으로 활발하지 못하다.

오늘날의 웹 어플리케이션은 더 복잡해지고 관리하기 어

려워지고 있다. 이러한 복잡성을 관리하는데 도움을 주는 체계적인 웹 어플리케이션이 필요하다. 따라서 웹 어플리케이션을 개발하기 위한 적절한 모델링 방법과 이를 이용한 분석과 프로세스에 관한 체계적인 연구가 요구된다. 그 하나의 방법으로 모델링은 주요한 부분을 차지하고 있다. 웹 어플리케이션에서 시스템 모델은 웹 페이지나 하이퍼링크와 같은 특정한 웹 구성요소들을 모델링하는 것이다[4]. UML(Unified Modeling Language)은 웹 어플리케이션 모델링을 하기에 부족한 부분이 있지만, 확장된 방법을 통하여 이를 해결하고 있다. 본 논문은 UML을 사용하여 소프트웨어 재사용을 목표로 모바일 고객관리 시스템을 설계 및 구현한다. 다시 말해서 본 논문에서는 소프트웨어 재사용을 목표로 스토리보드를 통한 사용자 요구사항을 기술하고, 기술된 요구사항을 통하여 시퀀스 다이어그램 및 클래스 다이어그램을 설계한다. 또한 고객관리 시스템에 요구되는 데이터는 객체 관계형 데이터베이스로 설계한다.

[†] 종신회원: 진주산업대학교 교양학부 부교수

^{††} 성회원: 에이스코리아 대표

논문접수: 2003년 5월 12일, 심사완료: 2003년 8월 19일

2. 관련 연구

2.1 기존 방법론

웹 어플리케이션이란 인터넷 상에서 동작하는 어플리케이션을 말한다[5]. 웹 어플리케이션의 개발방법으로 OOHD(M Object-Oriented Hypermedia Design Model) 개발단계[6], Takahashi 개발단계[7] 등이 있다. 그러나 이들 방법론은 분석 및 설계단계에서 행해져야 할 작업에 대한 상세한 정의가 없다. 그리고 산출물에 대한 정의가 빈약하고 산출물에 사용된 표기법이 웹 어플리케이션을 표현하기에는 심볼의 정의가 빈약하다. 이들을 해결하기 위해 개발 초기단계에서부터 구현에 이르는 전 과정에서 산출물이 서로 연관성을 가져야 한다. 또한 사용자 중심의 요구사항 분석이 가능해야 하며 이를 적절히 표현할 수 있는 충분한 심볼을 지원해야 한다.

소프트웨어 개발에 있어서 가장 기본적인 과정은 고객과의 면담을 통해 고객의 요구사항을 분명하게 기술하는 것이다. 복잡한 현실 세계의 요구사항을 보다 정확하게 파악하기 위해 반복적인 분석과 설계과정을 수행하여 모델링한다. 이때 많은 모델을 만들어 낼수록 상대적으로 많은 뷰를 나타내며 요구사항에 근접해 간다. 이런 다양한 관점을 다룰 수 있는 모델링 도구인 UML을 가지고 개발 프로세스에 적용하고 있다.

2.2 웹 어플리케이션 개발 방법론

UML은 객체지향 시스템을 모델링하기 위한 표준 다이어그램들과 표기법들을 규정하며 심볼들의 의미와 관련된 기본적인 의미론을 기술하고 있다. 또한 시스템 개발에 필요한 가공물들에 대하여 모델링을 위한 구성 요소 제공, 이를 이용한 추상화 방법 그리고 결과물 산출을 개발자들이 쉽게 이해할 수 있도록 가시화 하는 방법 등을 포괄적으로 정의한 모델링 언어이다[8].

UML은 이러한 다양한 관점을 다룰 수 있는 효과적인 모델링 도구로서 일반적으로 표준으로 제정된 UML을 가지고 개발 프로세스에 적용하고 있는데, 요즘 가장 많이 사용되고 있는 개발 프로세스가 RUP(Rational Unified Process)이다. RUP는 '유스케이스 중심'의 개념을 가지고 있어 개발자는 시스템의 기능적 요구사항들을 결정하고 시스템이 무엇을 할 것인가에 대한 상세한 설계로서 유스케이스 모델링을 수행하게 된다[9, 10].

웹을 응용한 웹 어플리케이션 아키텍처 패턴은 모두 웹 페이지를 사용한다. 웹 페이지는 클라이언트 즉 사용자와 접촉점을 이루는 부분이다. 즉, 웹 페이지는 사용자의 인터페이스 컨테이너를 일반화시킨 것으로 볼 수 있다. 또한 이것은 브라우저와 나머지 시스템을 결합해 주는 접착제와 같은 역할을 한다. 그러므로 모델링을 통한 개발이 시스템 설

계에 도움이 되려면, 소프트웨어 시스템의 모든 요소들과 그 협력을 파악하여 시스템에 대한 모든 질문들에 대한 모든 질문들에 답할 수 있어야 한다. 이때 가장 중요한 것은 웹 페이지이다. 즉, 웹 페이지를 모델에서 가장 중요한 요소로 파악하고 클래스나 컴포넌트들과 함께 표현하는 것이 매우 중요하다[11].

시스템 설계 초기단계에서 스토리보드를 통해서 웹 페이지 프로토타입을 만들 수 있고, 웹 어플리케이션에서 필요로 하는 웹 어플리케이션의 구조도 설계할 수 있다. 이때 생성되는 산출물을 중심으로 재사용 가능한 모듈을 만든다. 그리고 이러한 웹 페이지를 바탕으로 개발한 시스템의 주요 특징들은 다음과 같다.

- 계층적인 네비게이션 구조를 가지는 사용자 중심의 설계
- 요구사항의 동적인 특성
- 객체 지향적인 접근방법에 따른 컴포넌트 재사용 요구
- 비교적 짧은 주기로 개발
- 신속한 유지보수 및 리모델링이 필요
- 급속한 최신기술의 발달 및 적용이 필요
- 웹 프로젝트에 대한 관리 및 개발 표준이 미약

현재의 상황에서, 웹 아키텍처와 비즈니스 로직을 구현하는 기술 및 프로그래밍 언어, 디자인 그래픽 요소 등은 지속적인 발전을 거듭하는 반면, 개발 라이프 사이클을 제시하고 경로를 구성하는 프로세스 또는 방법론에 대한 논의는 상대적으로 활발하지 못하다. 이러한 요소로 인해 개발 현장에서는 기존에 사용해온 구조적인 기법 및 정보 공학적 접근방법을 웹 시스템 개발에 적용하고 있고, 웹 프로젝트는 개발자 개인의 경험과 숙련도에 의존한 진행, 임의적인 프로세스 경로수정, 개발 프로세스와 산출물의 불일치와 이로 인한 시스템 유지보수 및 확장의 어려움 등을 반복하고 있다.

본 논문에서는 소프트웨어 재사용을 목표로 스토리보드를 통한 사용자 요구사항을 기술하고 기술된 요구사항을 통하여 시퀀스 다이어그램 및 클래스 다이어그램을 설계한다. 또한 고객관리 시스템에 요구되는 데이터는 객체 관계형 데이터베이스로 설계한다.

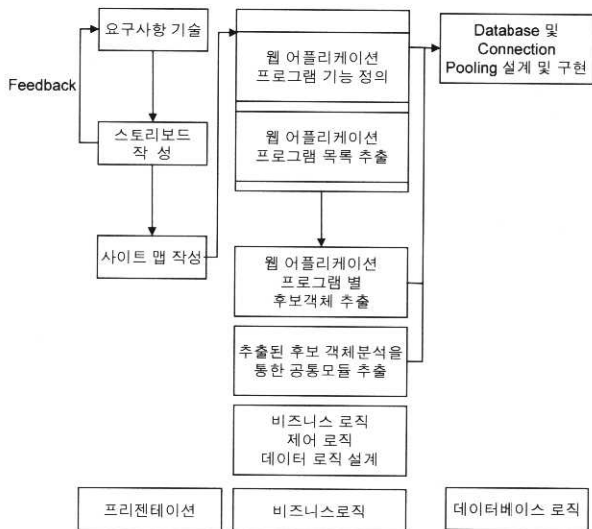
2.3 본 연구의 시스템 개발 방법론

본 연구에서 사용한 시스템 개발 방법론은 시스템 설계자, 웹 디자이너, 웹 프로그래머 등 웹 어플리케이션 설계자들이 의사소통을 공유할 수 있는 스토리보드를 기본으로 하는 것이다. 웹 어플리케이션 설계 시에 산출되는 요구사항을 바탕으로 스토리보드는 웹 어플리케이션 구축 시에 반드시 만들어져야 하는 것이고, 이 스토리보드는 사용자 인터페이스를 의미한다. 이러한 스토리보드는 대부분의 사용자의 요구사항을 반영하고 있으며 웹 어플리케이션의 특성

과 성격을 규정하고 있다. 그러므로 스토리보드를 분석하면 웹 어플리케이션에서 필요로 하는 컴포넌트를 추출할 수 있다. 웹 어플리케이션의 일반적인 구조가 거의 유사하므로 추출된 컴포넌트들은 다른 웹 어플리케이션 구축시에 전달되는 매개변수만 바꾸어서 사용할 수 있다.

3. 모바일 고객관리 시스템의 웹 어플리케이션 분석 및 설계

웹 어플리케이션 설계시에 산출되는 요구사항을 바탕으로 스토리보드를 설계한다. 설계된 스토리보드를 바탕으로 데이터의 흐름을 분석하여 필요한 메서드 및 관련 데이터를 추출한다. 이때 분석하는 데이터 흐름 및 데이터 처리에 대한 부분을 클라이언트와 서버로 나누어서 설계한다. 필요한 메서드 추출 및 데이터를 바탕으로 시퀀스 다이어그램을 설계하고 재사용 가능한 클래스를 설계한다. (그림 1)은 모바일 고객관리 시스템을 개발하는데 적용된 시스템 구조이다.



(그림 1) 개발 프로세스

프리젠테이션 영역은 사용자의 요구사항을 조사하여 스토리보드를 작성한다. 작성된 스토리보드를 가지고 사이트 맵을 작성한다. 요구사항 기술과 스토리보드 사이에 서로 피드백 되는 관계가 있으므로 반복해서 사용자의 요구사항을 스토리보드에 반영하게 된다. 스토리보드의 작업완료 후 사용자 인터페이스 부분인 사이트 맵을 작성한다.

비즈니스 로직에서는 프리젠테이션에서 작성된 산출물을 가지고 이 시스템에서 필요한 웹 어플리케이션 프로그램 목록과 클래스를 추출한다. 그리고 데이터베이스 로직은 데이터베이스관련 클래스와 객체 관계형 데이터베이스를 설계하는 부분으로 이루어진다.

3.1 웹 어플리케이션 구조 설계방법

웹 어플리케이션 개발을 위하여 전체적인 구조를 정의하고 설계한다. 일반적으로 웹 어플리케이션을 개발할 경우 대부분 두 가지의 개발 패턴형식을 가지고 있다.

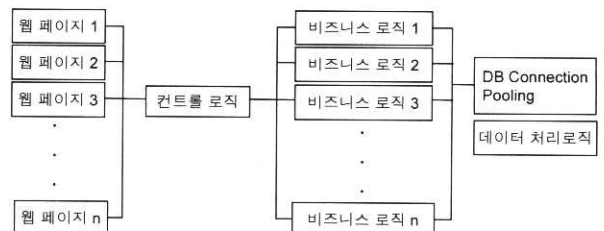
첫째, 페이지 중심의 설계 방법이다. 웹 사이트는 많은 문서들로 이루어져 있다. 정적 HTML 문서와 동적 HTML 문서들의 집합이 하나의 웹 사이트를 이루고 있다. 그러므로 웹 사이트 구축시 초기 개발단계에서 쉽게 접근하는 방법이 웹 페이지 중심의 설계방법이다. 각 웹 페이지마다 어떤 내용을 보여줄 것인지를 생각해서 각각의 페이지를 디자인하고 코딩에 들어가는 방식이다.

페이지 중심의 설계방식은 직관적으로 감을 잡기 쉽고 HTML 문서코딩과 연계작업이 어렵지 않아서 선호되는 방법이다. 더욱이, 코딩 단계마다 브라우저에서 결과를 바로 바로 보면서 기능을 추가하고 디버깅할 수 있기 때문에 개발속도도 빠른 편이다. 그러나 웹 페이지 중심의 설계 방식은 각각의 페이지에서 사용한 코드를 다시 사용하기가 어려운 단점이 있다.

웹 페이지 중심의 설계방식은 복잡하지 않으며 빨리 만들어야 할 필요성이 있는 웹 사이트의 경우에는 강점이 있으나 복잡한 기능을 구현할 경우에는 적합하지 않다.

둘째, 어플리케이션 중심의 설계 방법이다. 본 논문에서 지향하는 방법으로 웹 사이트 전체를 하나의 어플리케이션으로 보고 설계하는 방법이다. 각 웹 페이지가 전체 웹 사이트의 한 부분을 담당하는 기능으로 인식하는 것이다. 그러므로 각 웹 페이지마다 필요한 기능과 속성을 정의하여 공통적인 작업 모듈을 만든다. 그러면 웹 페이지에서는 필요한 기능들과 속성만 호출하여 사용할 수 있다[12].

본 시스템의 개발한 구조 (그림 2)와 같다. 즉, 각 페이지를 제어하는 컨트롤 로직이 있으며, 이 컨트롤 로직은 연관된 비즈니스 로직을 매핑(mapping)하는 구조이다. 하나의 컨트롤 로직이 전체의 웹 페이지를 제어하는 구조이다.



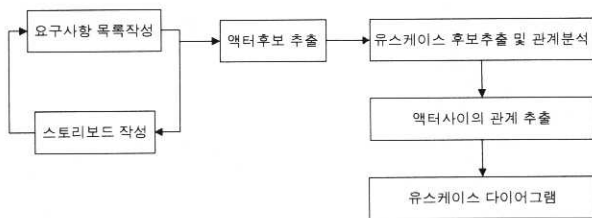
(그림 2) 어플리케이션 중심의 설계방법

3.2 요구사항 기술

요구사항 추출은 고객, 시스템 사용자, 그리고 시스템 개발에 관련된 사람들과 서로 의견을 교환함으로써 실제로 개발하고자 하는 시스템에 대한 요구들을 찾아내는 공정으로 시스템의 감추어진 정보를 관련된 모든 사람에게 분명하고

정확한 문장으로 언어내는 과정이다[10].

요구사항 추출단계를 통해서 요구사항 획득 후 분류한 요구사항과 추출한 공동어휘를 기반으로 시스템의 영역을 정의하고 UML 도구인 유스케이스와 액터를 사용하여 요구사항을 구조화한다. (그림 3)과 같은 단계를 통해서 요구사항을 추출 및 분석하여 고객, 시스템 사용자, 시스템 개발자간의 상호 커뮤니케이션을 위한 유스케이스 다이어그램을 추출한다.



(그림 3) 요구사항 및 유스케이스 추출

먼저, 구축하고자 하는 시스템의 목적을 설정한다. 그리고 의뢰인과의 대화를 통해서 기본적인 요구사항 목록을 작성한다. 작성한 목록을 바탕으로 스토리보드를 작성한다. 액터 및 유스케이스를 추출하기 전에 먼저 스토리보드를 작성하여 의뢰인의 의도를 정확히 파악한지를 점검한다. 이러한 피드백을 통하여서 요구사항 목록을 재 작성하며 스토리보드를 수정 보완한다. 이러한 과정을 통해서 요구사항과 스토리 보드를 작성하게 되면 액터 후보 및 유스케이스 후보를 추출하고 유스케이스 다이어그램을 작성한다. <표 1>은 요구사항 목록을 기술하는 양식이며 본 시스템에 대한

<표 1> 요구사항 목록기술서

개요-1
본 시스템 사용자에 대한 접근 권한 레벨을 설정한다. 접근 권한 레벨에 따라서 본 시스템의 접근을 허용하거나 거부한다. 사용자는 관리자, 기업, 일반고객이 될 수 있다.
업무영역 기술-BA1
<ul style="list-style-type: none"> 본 시스템과 관련된 사용자를 모두 관리한다[BA1-1]. 등록된 사용자를 본 시스템 사용목적에 따라 분류한다[BA1-2]. 등록된 사용자는 관리자, 기업, 기업관리자, 일반고객등으로 분류된다[BA1-3]. 분류한 사용자는 분류된 레벨에 적합한 접근 권한을 설정한다 [BA1-4].
To be Model-TM1
<ul style="list-style-type: none"> 관리자 모드 <ul style="list-style-type: none"> TM1-1. 관리자는 관리자 모드에서 접근 권한 레벨을 생성한다. TM1-2. 관리자는 생성된 접근 권한 레벨에 따라 사용자를 할당한다. 사용자 모드 <ul style="list-style-type: none"> 본 시스템을 사용하기 위하여 고객은 로그인 과정을 통하여 권한을 부여 받는다. TM1-3. 일반고객은 아이디와 비밀번호를 입력한다. TM1-4. 일반고객의 아이디와 비밀번호를 검증한다. TM1-5. 검증된 일반고객은 관리자 모드에서 설정한 레벨에 따른 접근 권한을 받는다.

실례를 보여 주고 있다.

<표 1>에서 개요는 사용자 요구사항에 대한 주 도메인 부분을 기술하며, 업무영역은 주 도메인에서 기술된 내용을 수행하기 위한 기능을 분류한다. 마지막으로 To Be Model은 주 도메인의 기능이 수행될 프로세스를 기술한다.

위의 요구사항 목록 기술서 및 액터 후보 추출은 다음과 같은 몇 가지 일반적인 규칙에 따라서 추출한다.

- 각 요구사항은 명백하고 간단해야 한다. 긴 문장은 잘못 해석할 가능성이 있다.
- 모든 요구사항은 반드시 검증할 수 있어야 한다.
- 주어가 생략된 부분은 의미를 파악하여 주어를 기술한다.
- 요구사항의 기술서 문장을 주어 + 목적어 + 서술어 형태로 기술한다.
- 동일한 기능을 하는 역할의 명칭은 동일한다.

<표 2> 액터 후보 목록관리 양식

일련번호	문장번호	액터후보	역 할
RBATM1-1	BA1-1	시스템	
RBATM1-2	BA1-1	사용자	시스템을 사용하는 사람
RBATM1-3	TM1-1	관리자	시스템을 관리하고 권한을 부여하는 사람
RBATM1-4	TM1-3	일반고객	시스템 사용자

<표 2>는 위의 일반적인 규칙에 따라서 추출한 액터 후보 목록관리이다. 각 액터마다 일련번호를 부여한다. 문장번호는 요구사항 기술서에 나타난 부분과의 연결고리이다. 그리고 <표 2>는 각 액터의 주요역할을 기술하고 있다.

3.3 스토리보드 작성

원래 스토리보드는 영화나 TV, CF, 만화 그림 등의 영상물이나 창작물을 제작할 때 사용되었던 방법으로 의뢰인과 제작자가 작품을 제작하기 전에 제작 전반에 걸친 내용을 간단하게 여러 개의 화면에 전달하고 싶은 영상을 시간적 흐름에 따라 그림으로 표현한 것이다[13].

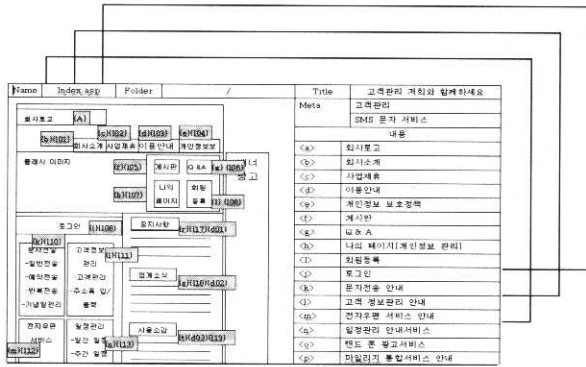
3.3.1 스토리보드의 역할

웹 사이트 제작시 스토리보드의 역할은 웹 사이트 기획을 제작자들과 의뢰인에게 표현하는 것이다. 또한 의뢰인, 기획자, 디자이너, 프로그래머가 사이트 제작되기 전에 미리 웹 사이트의 내용을 머릿속에 어느 정도는 같은 그림으로 그릴 수 있고 웹 사이트의 웹 페이지마다 설계도로서의 역할을 수행한다. 그러므로 다른 기획문서들이 표현하지 못하는 부분을 보완해 주는 역할을 한다.

3.3.2 스토리보드 요소

본 시스템의 스토리보드에 들어가는 요소는 (그림 4)와 같이 문서번호, 화면배치, 파일이름과 폴더이름, 문서 타이

들 및 메타태그, 주석, 링크 등이다. 문서번호는 웹 어플리케이션의 프로그램 목록과 일치한다. 화면배치는 클라이언트와 접촉점을 형성하는 인터페이스 역할을 한다. 그리고 문서 타이틀 및 메타태그는 각 페이지의 주요 특성을 기술한다.



(그림 4) 스토리보드

3.4 Use Case 분석

스토리보드 및 요구사항 목록 기술서를 바탕으로 Use Case를 분석하여 작업에 필요한 산출물을 작성한다. Use Case 분석은 구축할 업무 시스템의 요구사항으로부터 시스템의 영역과 시스템의 기능을 정확히 파악하는데 목적이 있다. Use Case 분석을 통해서 나오는 산출물은 Use Case 모델, 부가사항 기술서, 관련용어집(Glossary)이 있다. 이 산출물들은 구성원들이 이해하기 쉬운 용어를 사용해야 한다. 그 이유는 Use Case 분석을 통한 산출물을 이용해 사용자와 개발자, 그리고 시스템 구축에 관련된 모든 구성원들이 대화하고, 전체 프로젝트를 진행하는 기준이 되기 때문이다.

3.4.1 Use Case 모델

Use Case 모델은 Use Case Diagram과 Use Case Report로 구성된다. 즉 Use Case 모델은 액터와 Use Case, 그리고 그들간의 관계형성을 바탕으로 구현될 시스템의 업무 영역을 나타내는 Use Case Diagram과 Use Case 이벤트 흐름으로 이루어진다.

시스템 설계자는 시스템을 사용하는 사용자가 누구인지, 시스템이 제공하는 서비스는 어떠한 것이 있는지, 시스템과 인터페이스 하는 외부 시스템이나 기계 등은 어떠한 것이 있는지 등을 파악하면서 Use Case 모델을 작성해야 한다. 업무영역의 이벤트 흐름은 Use Case Report를 작성하는 방법과 Activity Diagram을 통해 표현하는 방법이 있다. 그림 6의 스토리보드는 고객의 요구사항을 반영한 것이다. 이를 통하여 Use Case Diagram과 Actor를 추출한다. 이 추출된 Use Case Diagram의 이벤트 흐름을 Activity Dia-

gram으로 표현할 수 있다.

Use Case 모델에서 Use Case를 구성하는 Use Case와 Use Case, Use Case와 Actor, Actor와 Actor의 관계가 존재한다. 이런 관계들은 시스템을 구성하는 기본 흐름이 되고 시스템을 이루고 있는 요소들간의 인터페이스를 분석하는데 중요한 역할을 한다.

Use Case는 적어도 하나 이상의 Actor와 관계를 가진다. 왜냐하면, Use Case는 시스템의 기능적 요구사항을 모델링한 것이기 때문에 그러한 요구사항을 초기화하거나 이 시스템을 사용하는 외부의 어떤 객체가 있어야 하기 때문이다. 이때 Use Case를 초기화하는 Actor를 능동적 Actor라 하고 Use Case로부터 결과를 넘겨받는 Actor를 수동적 Actor라 한다.

그리고 Use Case 모델링에서 이벤트의 흐름이 두 개 이상의 Use Case에 걸쳐 진행되는 경우에는 Use Case와 Use Case사이의 관계로 표현한다. 이들 사이의 관계는 크게 세 가지로 나누어진다.

첫 번째, 포함(Include)관계이다. 포함관계는 서로 다른 두 개 이상의 Use Case에서 공통된 이벤트 흐름이 존재할 때, 공통된 이벤트 흐름을 별도의 Use Case로 구성하고, 이 Use Case를 기존의 Use Case에서 참조하도록 설정해 놓은 관계를 의미한다.

두 번째, 확장(Extend)관계이다. 확장관계는 기존의 Use Case를 수정하지 않고 새로운 요구사항을 추가로 표현하고자 할 때 사용한다. 즉 새로운 요구사항에 해당하는 이벤트 흐름을 별도의 Use Case로 구성하여 기존 Use Case로부터 참조하도록 설정하는 관계를 의미한다.

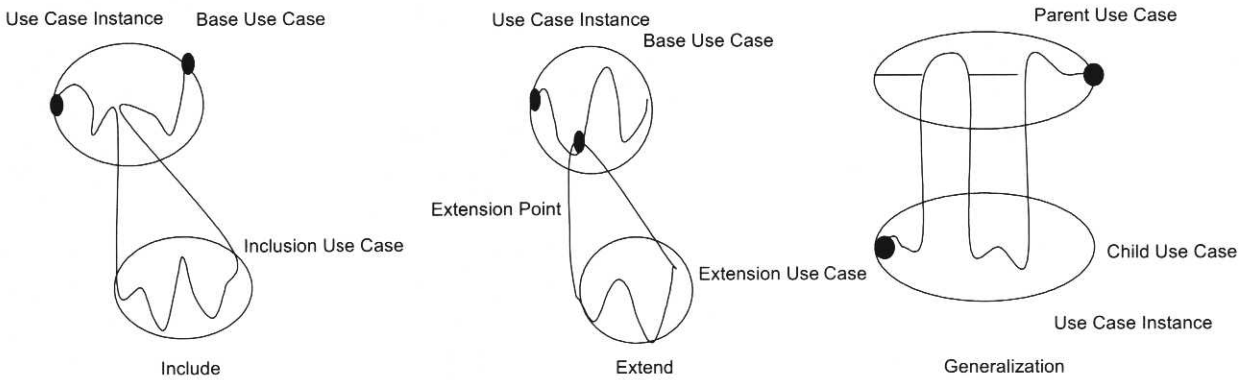
세 번째, 일반화(Generalization)관계이다. 일반화의 관계는 클래스의 상속관계와 유사한 개념이다. 자식 Use Case는 일반화된 Use Case(Parent Use Case)의 기능과 의미를 상속받고, 자신만의 고유기능을 추가하거나 재정의 하여 완전한 Use Case를 만든다.

(그림 5)는 Use Case 사이의 일반적인 관계를 그림으로 나타낸 것이다.

Use Case와 Actor사이의 관계표현을 사용하여 스토리보드 및 요구사항 목록 기술서를 가지고 (그림 6)과 같은 Use Case Diagram을 산출할 수 있다.

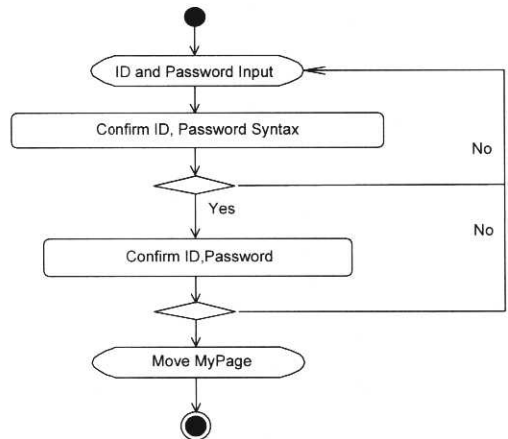
(그림 6)에서 나타난 스토리보드를 보면 모바일 고객관리 시스템의 기능요구 사항들이 잘 나타나 있다. 시스템의 기능요구 사항들이 대부분 Use Case Diagram에서 도출 된다는 것을 알 수 있다. 로그인 Use Case는 ID와 비밀번호를 확인하는 두 개의 Use Case와 일반화의 관계를 형성하고 있다. 그리고 Login Process Use Case는 다른 Use Case와 포함관계를 형성하고 있다.

Use Case Diagram을 가지고 각 Use Case의 업무 이벤트 흐름을 Activity Diagram으로 나타낼 수 있다. Activity

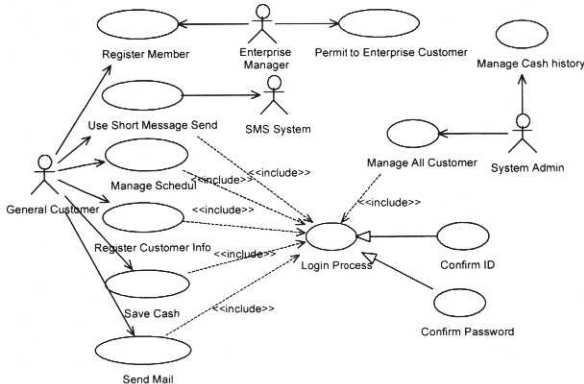
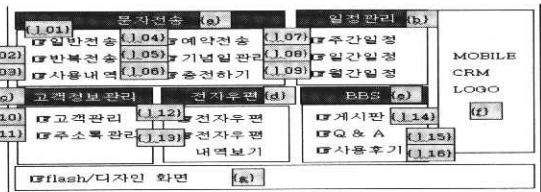


(그림 5) Use Case 사이의 관계

Diagram은 이벤트의 흐름을 각각의 동작 상태(Action State)로 구분하고 동작 상태간의 변화(Transition)을 다이어그램으로 표시한다.



(그림 7) 회원인증 Use Case에 대한 Activity Diagram



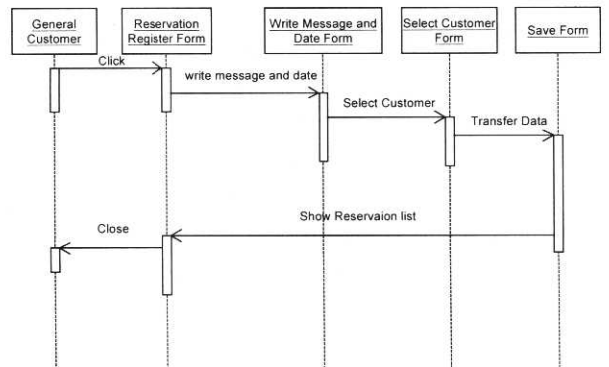
(그림 6) 스토리보드와 Use Case Diagram

동작상태는 이벤트의 흐름에서 발생되는 단계들을 표현하고 이러한 동작상태의 변화를 화살표를 사용하여 표현한다. 이때 각각의 상태변화를 표현할 때 조건(Guard Condition)을 줄 수 있으며, 이러한 조건들에 대한 분기가 발생하는 경우 분기 표시도를 이용한다. (그림 7)의 Activity Diagram은 회원인증에 관한 Use Case Diagram에 대한 이벤트 흐름이다.

이 Activity Diagram은 아이디와 비밀번호를 입력하면, 입력한 아이디와 비밀번호가 일정한 규칙에 맞는지 검사한다. 아이디와 비밀번호가 일정한 규칙에 맞게 입력되면 비밀번호와 아이디가 일치하는지 확인한다. 비밀번호와 아이디가 일치하는 자신의 페이지로 이동하면 하나의 프로세스 흐름이 끝난다.

3.5 시퀀스 다이어그램

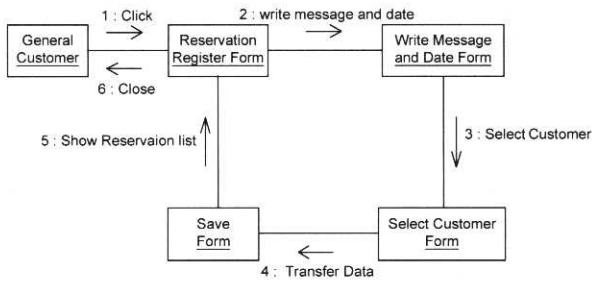
UML에서 객체를 찾아내고 객체간의 메시지를 파악하는 Use Case 실체화는 인터랙션 다이어그램으로 표현한다. 인터랙션 다이어그램은 시퀀스 다이어그램과 콜러보레이션 다이어그램이 있다. 시퀀스 다이어그램은 작성된 시나리오로부터 객체를 찾아내며 객체와 객체가 주고받는 메시지를 파악하여 진행순서에 따라 표현한 것이다. 콜러보레이션 다이어그램은 객체와 객체사이의 연결과 메시지 자체에 초점을 두고 작성되어 진다.



(그림 8) 문자예약 전송 시퀀스 다이어그램

시퀀스 다이어그램은 액터의 인스턴스와 각각의 식별자를 갖는 객체 그리고 그들이 주고받는 메시지로 이루어진다. 시퀀스 다이어그램을 처음으로 작성하는 경우에는 모든 객체에 대한 클래스가 아직 지정되지 않았으며 이후 작업을 통하여 클래스가 추출되면 해당 객체에 대해서 클래스를 지정한다. 이러한 작업의 반복을 통하여 업무영역에 존재하는 모든 객체와 모든 클래스를 파악하게 된다. 그림 8은 클라이언트가 문자를 예약 전송하는 과정에 대한 메시지의 흐름을 나타낸 것이다. 즉, 클라이언트가 문자예약 전송 폼을 클릭하면 문자예약 전송 폼이 나타난다. 사용자가 문자예약 전송 폼에서 문자메시지와 날짜를 기록하고 보내고자 하는 상대방을 선택한다. 상대방을 선택 한 후 기록한 모든 데이터를 저장 객체에 전달하면 저장객체는 데이터를 저장하고 예약목록을 클라이언트에게 출력한다.

콜러보레이션 다이어그램에서 객체의 표기는 기본적으로 시퀀스 다이어그램과 동일하다. 다만 시퀀스 다이어그램에서 표기되었던 생명주기는 표현하지 않는다. 객체들을 평면으로 배열하고 배열된 객체들 간의 연관성과 전달되는 메시지를 중심으로 표현한다.



(그림 9) 문자예약 전송 콜러보레이션 다이어그램

(그림 9)는 문자예약 전송 시퀀스 다이어그램을 콜러보레이션 다이어그램으로 변환한 그림이다.

3.6 클래스 설계

요구분석 단계에서 유스케이스 모델과 스토리보드를 통해서 객체들과 객체들간의 메시지를 파악한 것을 바탕으로 객체와 클래스를 추출하고 클래스들 간의 관계성을 정의하는 클래스 분석단계를 진행한다. 사용자의 요구사항 분석을 바탕으로 스토리보드를 작성하면 클라이언트에서 처리할 프로세스와 서버 측에서 처리할 프로세스 목록이 만들어진다. 클라이언트에게 보여지는 웹 페이지는 클라이언트의 입력을 받아서 이를 서버 측에서 처리하는 과정을 공통적으로 가지고 있다. 그러므로 각 웹 페이지에서 사용자의 입력을 받아서 서버 측에서 사용자의 요구사항을 처리하는 서버 측 프로세스는 동일한 기능이 반복적으로 요구됨을 <표 3>을 통해서 알 수 있다. <표 3>에 나타난 자료를 보면 하나의 웹 페이지가 하나의 객체 역할을 하는 것을 알 수 있다.

그러므로 스토리보드에서 설계된 웹 페이지를 하나의 객체 후보로 설정할 수 있다. 객체 후보 리스트를 만든 후 다시 피드백을 통하여 각 페이지마다 정확한 기능과 의뢰인의 요구사항을 정의하여 객체 후보리스트를 다시 점검하고 이 객체들을 추상화하여 클래스를 선정한다. (그림 10)은 이러한 과정을 그림으로 나타낸 것이다.

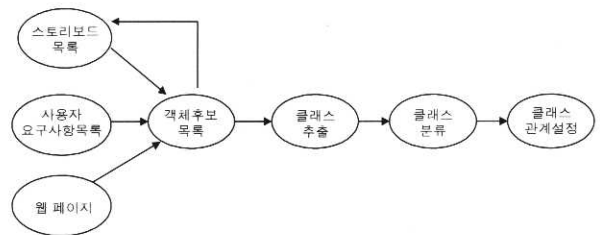
이때 산출된 객체 후보 리스트와 사용자 요구사항 그리고 스토리보드를 참고하여 클래스의 속성과 메서드를 정의한다. 이러한 작업을 통하여 클래스가 만들어지면 각 클래스 사이의 관계를 설정한다.

<표 3> 객체후보 리스트

클라이언트 페이지	액션	서버 측 페이지	처리 모듈
메인페이지 [main.html]	로그인하기	login_check.jsp	DB 연결 아이디 확인 비밀번호 확인
회원가입 [member.html]	회원 가입하기	member_insert.jsp	DB 연결 아이디 중복확인 회원등록[insert]
SMS사용 [sms_send.html]	메시지 작성	sms_send_ok.jsp	DB 연결 문자전송 내역저장
SMS사용내역 [sms_list.html]	사용내역 보기	sms_list_print.jsp	DB연결 select 쿼리 실행
SMS충전하기 [sms_cach.html]	캐쉬 충전	sms_cash.jsp	연결, insert 문 실행
SMS충전내역 [cach_list.html]	충전내역	cach_list.jsp	DB 연결 select 실행 cache 총계구하기

3.6.1 클래스 정의서

클래스 정의서는 클래스 설계과정에서 만들어진 클래스 목록이다. 클래스를 만들기 위한 프로세스는 (그림 10)의 과정을 따라서 진행된다. 그리고 클래스를 식별하는 작업은 많은 응용영역들에 대한 경험이 필요하지만, 클래스를 식별할 때 다음과 같은 기준을 적용한다.



(그림 10) 클래스 추출 프로세스

기준 1: 저장하거나 분석할 데이터를 찾는다. 이런 데이터는 시스템에 적용해야 하는 개념이나 특정한 순간에 발생하는 이벤트 혹은 트랜잭션으로 간주된다.

기준 2: 외부 시스템을 찾는다. 외부 시스템은 개발중인

시스템이 상호작용 하여야 하는 클래스로 간주할 수 있다.

기준 3 : 패턴, 클래스 라이브러리, 컴포넌트를 찾는다.

기준 4 : 액터가 어떤 역할을 수행하는지 찾는다. 사용자, 시스템 오퍼레이터, 고객 등과 같은 액터의 역할은 클래스로 간주될 수 있다.

요구사항 명세서나 스토리보드 등은 클래스를 식별하기 위한 기초 자료이다. 위의 4가지 기준을 가지고 <표 4>와 같은 PersonInfo 클래스 정의서를 작성 할 수 있다.

3.6.2 클래스 다이어그램

클래스가 추출된 후 각 클래스에 대한 관계를 설정하며 각 클래스간의 관계 설정과 관련된 여러 가지 디자인 패턴을 적용한다. 기본적인 클래스 다이어그램간의 관계는 다음 세 가지로 분류되며 이 세 가지 관계를 기본으로 하여 클래스 다이어그램을 작성했다.

① 종 속

종속은 클래스간의 사용관계를 나타낸다. 즉, 한 클래스가 다른 클래스를 사용할 경우 종속을 사용한다. 종속은 사용되는 클래스가 변경되면 클래스의 인터페이스나 행위가 달라질 수 있기 때문에 이를 사용하는 클래스가 영향을 받는다. 종속의 가장 일반적인 유형은 오퍼레이션의 매개변수로서 다른 클래스를 사용하는 클래스간의 연결이다. 이러한 사용관계를 모델링하기 위해서 오퍼레이션을 갖는 클래스로부터 오퍼레이션의 매개변수로 사용되는 클래스로 향하는 종속을 생성하여야 한다.

② 일반화

일반화는 슈퍼클래스와 보다 세부적인 서브클래스 간의 관계이다. 일반화는 “is-a-kind-of” 관계라고도 한다. 서브클래스는 속성이나 오퍼레이션 등과 같은 부모의 특성을 상속받는다. 부모, 자식 관계를 표현할 때 일반화를 사용한다. 일반화는 대부분 상속 관계를 표현하기 위해 클래스와 인터페이스간에 사용된다.

③ 연 관

연관은 객체와 객체를 연결하는 구조적이 관계를 의미한다. 연관은 이름, 역할, 다중성, 집산화 등고 같은 네 가지의 추가적인 표현 사항을 갖는다. 두 개의 클래스를 연결하는 연관에서 한 클래스의 객체로부터 다른 클래스의 객체로 네비게이션을 할 수 있다. 클래스가 연관 관계를 가질 때 각각의 클래스는 각자의 특정한 역할을 갖는다. 역할은 연관에 참여하는 하나의 클래스가 다른 클래스에게 제공하는 책임과 같다. 다양한 모델링 상황에서 객체가 얼마나 많은 연관에 연결되어 있는가를 표현할 수 있다. 이를 다중성(multiplicity)이라고 한다.

일반적인 두 클래스간의 연관은 동등한 수준에서 두 클래스간의 구조적인 관계를 표현한다. 그러나 때로는 하나의 클래스는 보다 큰 집합을 표현하고 다른 클래스는 보다 작은 집합으로 표현할 경우가 있다. 이러한 관계유형을 집산화(aggregation)라고 하며, 전체에 속하는 객체가 부분에 속하는 객체를 갖는 “has-a”의 관계를 나타낸다.

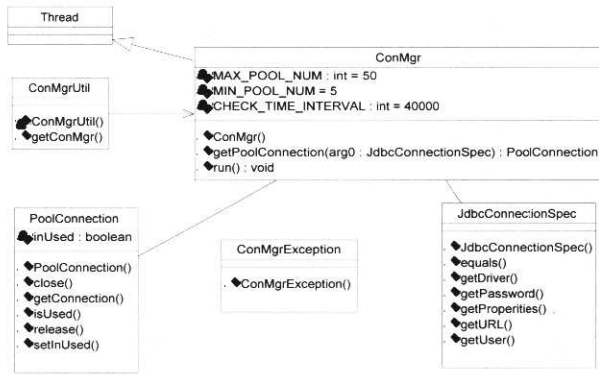
<표 4> PersonInfo 클래스 정의서

Class Name	PersonInfo	No.	1
Stereo Type		Refer StoryBoard	page 10
Super Class			
Sub Classes			
Class Description	이 사이트의 고객정보를 관리하는 클래스이다.		
Attribute		Attribute Name	
MemID	회원아이디		
ClassCode	회원분류코드		
CashCode	캐쉬적용코드		
CompCode	회사코드		
HP	핸드폰		
Name	이름		
Jumin	주민등록번호		
Pwd	비밀번호		
Email	전자우편		
BirthDay	생일		
Method		Method Name	
String getMemID()	회원의 아이디를 읽어온다.		
setMemID(String id)	회원 아이디를 등록한다.		
:	:		

(그림 11)은 데이터베이스 연결 및 쿼리를 관리하는 클래스들 사이의 관계를 표현한 것이다. ConMgr은 커넥션 풀을 관리하는 클래스로써 PoolConnection 클래스와 JdbcConnectionSpec 클래스 사이에는 연관관계가 성립한다. 이렇게 구현된 컴포넌트를 통하여 여러 가지 제품의 데이터베이스 상관없이 동일한 인터페이스를 통하여 데이터베이스 connection pool을 사용할 수 있다. 구현된 JdbcConnectionSpec 클래스를 통하여 데이터베이스 vendor에서 제공하는 jdbc 드라이버를 동일한 인터페이스를 통하여 사용할 수 있으므로 재 사용성이 높아지게 된다. 즉, 본 시스템에서 데이터베이스 커넥션을 효과적으로 관리하도록 하였다. jdbc를 사용하여 반복적인 커넥션 코드의 사용을 간단하게 처리하였고, 환경변수에 데이터베이스 커넥션 정보를 저장하여 데이터베이스 서버의 변경에 따른 코드 변경을 없앴다. 또한 둘 이상의 데이터베이스 커넥션에 대해서도 간편하게 설정이 가능하다.

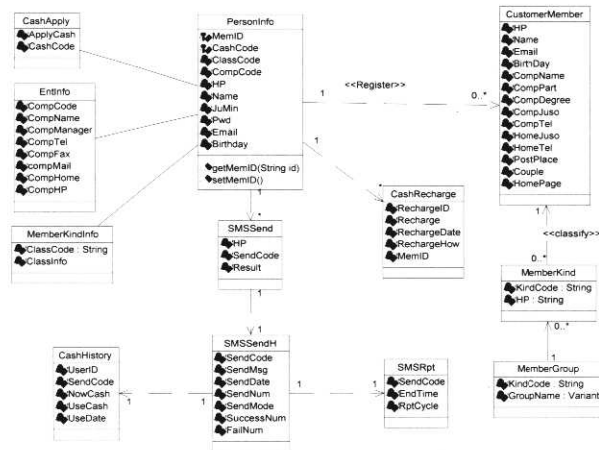
그리고 데이터베이스에서 조회, 갱신하는 쿼리 작업 코드와 데이터베이스 연결을 생성하고 쿼리 작업을 관리하는 코드를 분리하였다. 이는 쿼리 작업을 수행하는 코드는 데이터

베이스의 커넥션 환경에 대해서 알 필요가 없이 정해진 쿼리만 수행하여 결과를 반환하도록 하였다. 그래서 환경의 변화에 따른 코드 변경을 제거하였다. 이것은 데이터베이스의 커넥션 생성과 해체에 관련된 역할을 배제한 것이 되므로 연결을 열고 닫는데 자주 생기는 문제에서 해방시켜 준다.



(그림 11) DB Connection Pool Class Diagram

(그림 12)는 SMS문자 전송을 위한 클래스들 사이의 관계를 모델링 한 것이다. PersonInfo 클래스는 클래스의 중심에서 다른 클래스와 여러 가지 관계를 가지고 있다. 즉, PersonInfo 클래스는 문자전송 클래스인 SMSSend 클래스와 종속관계를 가지고 있으며 CashApply 클래스와는 연관 관계를 형성하고 있다. 그리고 CashHistory Class, CashRecharge Class 등 목록 형식으로 데이터를 보여 주는 클래스들은 동일한 구조를 가지고 있다. 그래서 맵과 리스트 기능을 동시에 가지는 클래스를 설계하였다. 이 클래스는 맵 기능과 리스트 기능을 선택적으로 편리하게 사용할 수 있으며 동시에 사용가능 하기도 한다. CashHistory Class, CashRecharge Class 등 목록 형식으로 데이터를 보여 주는 클래스가 게시판 목록을 가져올 때 게시물 10개와 현재 페이지 번호, 전체 게시물 수, 게시판의 특성 등을 가져온다면



(그림 12) SMS Class Diagram

이 클래스를 이용하여 하나에 담아 처리할 수 있게 된다.

4. 결 론

본 논문은 객체지향 개발방법론인 UML을 적용하여 모바일 고객 관리 시스템 설계 및 구현한 것이다. 즉 의뢰인의 요구사항이 잘 반영된 스토리보드와 객체지향 방법론인 UML을 이용하여, 모바일 고객 관리 시스템을 설계 및 구현하였다.

스토리보드를 통해서 의뢰인과 시스템을 설계 및 구축하는 개발자와의 의사소통이 원활했고, 스토리보드를 이용하여 객체의 후보를 추출하고 추출된 후보를 통하여 추상화된 클래스를 추출할 수 있었다. 또한, 이 시스템에서 개발된 클래스들은 이와 유사한 웹 어플리케이션을 구축할 경우 재사용 가능한 구조로 설계되어 있다. 그러므로 설계된 클래스를 이용하여 다른 플랫폼에서 동일한 모바일 시스템의 구축 시에도 구조 변경 없이 사용할 수 있어 생산성이 뛰어나다. 즉, 본 시스템에서 설계된 클래스들은 다른 시스템의 구축 시에도 파라미터 값만 조절하여 사용할 수 있으므로, 시스템 개발 기간을 단축시키는 장점이 있다. 결론적으로, 본 시스템을 개발하면서 사용된 개발 과정 및 여러 가지 산출물들을 이용하면 이와 유사한 시스템을 구축할 시에 비용 및 시간을 줄일 수 있다.

웹 어플리케이션 개발시에는 동일한 과정을 거쳐서 개발해야 되므로, 본 시스템을 통하여 개발된 라이브러리 및 시스템 아키텍처를 프레임워크로 정형화 시켜서 다른 프로젝트에 적용할 수 있도록 해야 한다. 이는 추후 연구과제로 본 시스템을 개발하면서 형성된 일련의 개발과정을 하나의 프레임워크로 정형화하는 것이다.

참 고 문 헌

- [1] J. P. Kuilboer and N. Ashrafi, "Software process and product improvement : an empirical assessment," Information and Software Technology, pp.27-34, 2000.
- [2] David Lowe, Richard Webby, "Web Development Process Modeling and Project Scoping : Work in Progress," WebE '98, 1998.
- [3] 최준용, 김영대, 김병기, "웹 어플리케이션의 요구분석에 관한 고찰", 한국정보처리학회 소프트웨어공학연구회지, pp. 3-13, 2000.
- [4] 김행권, 신호준, "UML + Navigation Diagram 기반 웹 도메인 응용 개발 프로세스에 관한 연구", 정보처리학회논문지, 제7권 제9호, pp.2846-2856, 2000.
- [5] 오해영, "정보공학방법론의 기술동향" 한국정보처리학회 소프트웨어공학연구회 소프트웨어공학연구회지, pp.1-20, 1995.

- [6] 배명남, 최 완, 양현택, “웹을 사용한 객체지향 설계정보 분석”, 정보과학회논문지 : 소프트웨어 및 응용, 제27권 제7호, pp.702-711, 2000.
- [7] D. Schwabe, G. Rossi, “The object-oriented hypermedia design model,” Communications of the ACM, 38(8), 1995.
- [8] Martin Fowler, Kendall Scott, UML Distilled, Addison-Wesley, 1997.
- [9] OMG, “Unified Modeling Language Specification Version 1.3,” <http://www.rational.com/>, 1999.
- [10] Lesz A. Maciasz, Requirements analysis and system design Developing information system with UML, Addison-Wesley Pub. Co., 2001.
- [11] Grady Booch, Building Web Applications with UML, pp. 191-222, 2001.
- [12] 김민식, 김세곤, JSP BIBLE, 정보문화사, 2001.
- [13] 오종혁, 할수 있다 특별판 성공적인 웹 사이트 구축을 위한 웹 기획 & 웹 프로젝트 매니지, 영진출판사, 2002.



심 갑 식

e-mail : gssim@jinju.ac.kr

1987년 전남대학교 계산통계학과(이학석사)

1993년 전남대학교 전산통계학과(이학박사)

1993년~현재 진주산업대학교 교양학부

부교수

관심분야 : 정보 보안, 무선인터넷, 데이터 모델링, XML



정 태 영

e-mail : garisan@lycos.co.kr

1996년 경상대학교 전자재료공학과(공학사)

1997년 경상대학교 전자재료공학과(공학석사)

석사)

1997년 (주)광성전자 선임연구원

2003년 에이스코리아 대표

관심분야 : 데이터마이닝, CRM, 무선인터넷