

초고속 모바일 웹 개발을 위한 경량화 프레임워크[☆]

Lightweight Framework For Supporting Mobile Web Development

신 승 우* 김 행 곤**
Seung-Woo, Shin Haeng-Kon, Kim

요 약

모바일 웹 애플리케이션들이 다양한 모바일 디바이스의 성능향상으로 형태로 변화하고 있다. 하지만 기존 PC환경과 달리 개발환경 및 표준에 대한 고려가 적어 애플리케이션을 디바이스 별로 구현하는 등의 이유로 개발비용이 소모되고 있다.

본 논문에서는 MWeb-프레임워크를 통해 최근 각광 받고 있는 웹 프레임워크기반의 모바일 웹 개발을 수행하는 개발방법론을 제안하며 이와 더불어 다양한 디바이스에서 동일한 사용이 가능하도록 모바일 웹 표준 준수를 지원한다. Ruby on Rails기반의 모바일 웹 개발 아키텍처를 정의하고 이를 바탕으로 모바일 웹 표준을 준수하는 애자일 개발방법론을 적용한 프로세스 모델을 제안한다. 제안한 모바일 웹 개발 프레임워크를 통한 사례연구를 수행하여 제안한 프레임워크를 검증한다. 이를 통해 개발 수행 시 애자일 개발방법론을 도입하여 신속한 개발을 수행할 수 있으며 모바일 웹 표준 환경을 준수하여 모바일 웹 애플리케이션의 품질과 신뢰성을 획득할 수 있다.

향후 연구로 제안한 MWeb-프레임워크의 다양한 사례연구를 통해 잠재적인 문제를 개선하고 이를 바탕으로 표준화를 제안한다.

Abstract

Mobile web applications are being used and changed rapidly due to the growth of mobile device performance. But, cost of development environment and standards make the high development cost and low productivity. It is main reason that the design and implementation of the applications are more time consuming than general computing environments.

In this paper, we propose MWeb(MobileWeb)-Framework based on the agile methodology and Ruby on Rails that is a kind of framework for supporting mobile web application development using mobile web standards. This work consists of the mobile web development architecture and agile process model. MWeb-Framework will support the same user experience to the different devices. We validates the framework by implementing the case studies through suggested mobile web development framework. As a result, we can develop the mobile web applications with productivity and quality.

In the future, we will suggest how to make the MWeb-Framework standardization and practically apply the frameworks the various case studies to improve framework potentially problems.

☞ keyword : 소프트웨어 프레임워크, 모바일 웹 개발, 애자일 개발방법론, 익스트림 프로그래밍, 스크럼, 모바일 웹 표준, 루비 온 레일즈, 모바일 모델링, Software Frameworks, Mobile Web Development, Agile Methodologies, XP(eXtreme Programming), Scrum, Mobile Web Standards, Ruby on Rails, Mobile Modeling

1. 서 론

최근 모바일 웹 환경이 변화하고 있으며 모바일 디바이스가 빠르게 보급됨으로 인해 모바일 디바이스의 동작환경은 나날이 개선되고 있다. 디바이스의 성능이 비약적으로 발전하고 있고 입력 방식도 터치센서나 Qwerty 타입의 입력방식으로 개선되어 조작이 용이해 지고 있다. 무엇보다도 네트워크의 접속속도가 와이브로(Wibro)나 HSDPA(High

* 준 회 원 : 대구가톨릭대학교 컴퓨터정보통신공학부 석사
selab@cu.ac.kr

** 정 회 원 : 대구가톨릭대학교 컴퓨터정보통신공학부 교수
hangkon@cu.ac.kr

[2009/01/02 투고 - 2009/01/09 심사(2009/04/14 2차 - 2009/05/25 3차) - 2009/05/28 심사완료]

☆ 이 논문은 2008년도 정부(교육과학기술부)의 재원으로 한국 과학재단의 지원을 받아 수행된 연구임
(No. R01-2008-000-20607-0)

Speed Downlink Packet Access) 등의 고속 접속방식으로 바뀌고 있다. 이로 인해 모바일 디바이스의 통신방식이 기존의 WAP(Wireless Application Protocol)에서 TCP/IP 기반의 HTTP(Hypertext Transfer Protocol)을 사용할 수 있게 되어 PC 환경과 유사한 웹 환경을 갖추게 되었다[1].

개선되고 있는 모바일 웹 환경과 달리 모바일 웹 개발에 관한 연구는 아직 미흡하다. 모바일 디바이스에서 HTTP 기반의 모바일 웹 애플리케이션에 특화된 개발방법론에 대한 연구가 부족하다. 또한 웹 개발 프레임워크 기반으로 개발된 모바일 웹 표준을 준수하는 애플리케이션은 아직까지 찾아보기 힘든 실정이다.

본 논문에서는 루비 온 레일즈(Ruby on Rails) 웹 프레임워크 기반 모바일 웹 개발방법론을 제안한다. 이와 더불어 다양한 디바이스에서 정상적인 동작이 가능하도록 지원하기 위하여 모바일 웹 표준을 적용하여 개발을 수행하도록 지원한다. 또한 점점 빠르게 변화하고 있는 사용자의 요구 사항을 신속하게 반영하기 위해 애자일(Agile) 개발 방법론을 적용하여 생산성 및 품질을 보증할 수 있도록 지원한다.

2. 관련연구

2.1 모바일 웹 표준

W3C는 웹에 관한 다양한 표준을 제정하는 단체이며 모바일 웹 이니셔티브(Mobile Web Initiative)를 통해 모바일 웹 표준에 관한 다양한 활동을 수행하고 있다. W3C는 이러한 모바일 웹 표준을 위해 mobileOK 라는 인증마크를 도입을 하여 모바일 웹 표준에 대한 것을 알리고 있다. 모바일 웹 이니셔티브의 활동을 통한 모바일 웹 표준화 활동은 아래와 같다[2][3].

- Mobile Web Application Best Practice - MWBP 1.0
- mBT(mobileOK Basic Test) 1.0
- Content Transformation Guideline 1.0

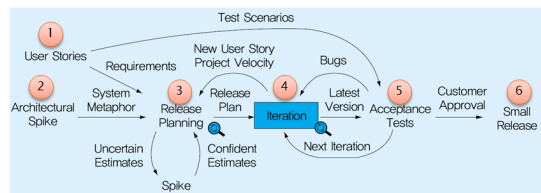
- mobileOK Checker & Library
- DDR(Device Description Repository)
- Web Compatibility Test for Mobile Browsers
- XHTML Basic 1.1
- CSS Mobile 2.0 Profile

2.2 애자일 웹 개발 방법론

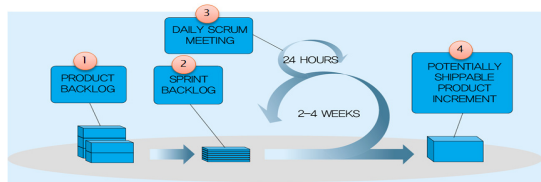
애자일(Agile) 소프트웨어 개발방법론은 켄트 벡(Kent Beck)이 소프트웨어 플래닝, 코딩, 디자인, 테스트의 가치와 원리와 방법론을 통해 eXtreme Programming (XP)을 시발점으로 하여 전통적인 개발방법론이 표준화된 프로세스에 가치를 두는 것과 달리 의사소통과 협동, 빠른 변화의 가치를 중시하는 90년대 후반에 태생된 개발방법론이다[4][5].

XP는 대표적인 애자일 기법을 실천한 응용 예로서 1999년 C3 프로젝트의 참여자인 켄트 벡(Kent Beck)과 론 제프리즈(Ron Jeffries), 마틴 파울러(Martin Fowler) 등의 경험에서 비롯된 방법이다. XP의 프로젝트의 진행과정은 그림 1과 같다[4][5].

Scrum(스크럼)이란 단어 자체는 본래 럭비에서 유래된 것으로, 제프 셔덜랜드(Jeff Sutherland)와 켄 슈와버(Ken Schwaber), 마이크 비들(Mike Beedle) 등에 의해 소개된 후 점차 확산되고 있는 방법이다. 그림 2는 이런 Scrum의 전반적인 진행도를 그림으로 도식화 한 것이다[4][6].



(그림 1) XP 프로젝트의 진행과정



(그림 2) Scrum의 전반적 진행도

2.3 웹 개발 프레임워크

웹 개발 프레임워크는 웹 개발을 수행함에 있어 생산성과 품질을 향상시킬 수 있도록 지원하는 일련의 단위로 동적인 웹 사이트나 애플리케이션, 서비스를 구축 할 수 있도록 만들어진다. 웹 개발 프레임워크를 통해 웹 개발을 수행 할 경우 유사한 개발을 유도하는 환경을 제공하게 되어 팀원간의 원활한 의견교환이 가능하고 애플리케이션의 개발과 유지 보수 성능 등에 일관성을 가질 수 있게 된다. 웹 개발 프레임워크는 검증된 개발 기반 환경을 제공하기 때문에 안정적으로 작동할 수 있는 프로그램을 개발하는 데 큰 도움이 된다.

그중에 최근 많은 관심을 받고 있는 웹 개발 프레임워크는 아래와 같다.

- **Struts(Java)** : 뷰와 비즈니스 로직을 분리하여 유지보수를 용이하도록 지원하는데 초점을 맞춘 Java기반의 프레임워크이다[8].
- **Ruby on Rails(Ruby)** : Ruby로 작성된 MVC(Model-View-Controller) 구조 기반의 오픈 소스 웹 프레임워크이다. Rails는 Ruby의 풀-스택, 오픈 소스 웹 프레임워크로 애플리케이션 구현이 쉽고, 코드는 적게 든다. Ruby on Rails는 MVC 아키텍처를 지원하며 MVC의 템플릿을 자동 생성하는 기능이 있다.[9][10].
- **CakePHP(PHP)** : 쉬운 사용과 HTML의 처리에 강점을 가지고 있는 PHP 기반 프레임워크로 Ruby on Rails의 인기에 영향을 받아 개발된 웹 프레임워크다[8].

3. 모바일 웹 표준 지원 애자일 프레임워크

3.1 MWeb-프레임워크 아키텍처

모바일 웹 애플리케이션 개발을 지원하기 위한 개발환경을 제안하면 그림 3과 같다.

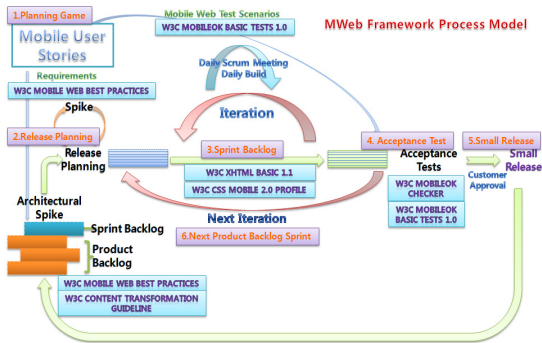
본 모바일 웹 개발 아키텍처는 Ruby on Rails 프레임워크에 그 기반을 두고 있으며 모바일 웹

개발을 지원하기 위한 다양한 환경을 고려한 MWeb-프레임워크를 포함한다. 이를 바탕으로 개발환경까지 고려하여 구성한다. MWeb-프레임워크의 계층별 구조는 다음과 같다.



(그림 3) 모바일 웹 개발 아키텍처

- **Hardware - X86 Series** : 본 아키텍처에서는 개발환경으로 가장 보편적인 환경인 X86 Series 플랫폼을 기반환경으로 한다.
- **Operating System - Any OS** : Windows나 Linux, Mac OS 등 운영체제는 자유롭게 선택이 가능하다.
- **Web Server - Mongrel + Nginx** : 개발과정에서는 잦은 변경이 발생하므로 프레임워크의 변화에 대응할 수 있는 Mongrel을 개발환경에서 운용하고 배포에서는 Nginx 웹 서버를 이용하여 응답속도를 향상 시키는 것을 지원한다.
- **DBMS - Sqlite3** : Ruby on Rails 2.0의 경우 개발과정에서 Migration 수행 시 월등히 편리한 사용이 가능하다.
- **Ruby on Rails Framework** : 개발하는 애플리케이션이 Ruby기반으로 구동되기 위한 기반환경 및 자동화 생성기능을 제공하여 생산성을 향상시킨다.
- **MWeb-Framework** : MWeb-프레임워크는 본 아키텍처의 핵심요소이다. 하위 레이어들을 기반으로 하여 모바일 웹 애플리케이션 개발환경을 제공한다.



(그림 4) 모바일 웹 표준 기반 애자일 프로세스 모델

3.2 모바일 웹 표준지원 프로세스 모델

모바일 웹 애플리케이션을 개발할 때 핵심적 고려사항은 다양한 디바이스에서 동일한 콘텐츠를 보여줘야 한다는 것이다. 물론 모바일 디바이스의 성능이 향상됨에 따라 PC와 유사한 환경을 지원하는 모바일 웹 브라우저도 개발되고 있으나 디바이스의 디스플레이 환경은 쉽게 커질 수 없고 입력 환경도 PC에 비해 불편하다.

따라서 본 논문에서는 모바일 웹 표준을 준수하여 사용자의 불편함을 최소화할 수 있도록 하면서도 빠른 모바일 웹 애플리케이션의 개발을 가능하게 하는 애자일 프로세스 모델을 제안한다. 그림 4는 모바일 웹 표준 지원 애자일 프로세스 모델을 나타낸다. 이는 대표적인 애자일 개발방법론인 Scrum을 기반으로 하여 XP의 프랙티스 요소를 결합한 형태이며 여기에 W3C에서 모바일 웹 표준으로 제안하고 있는 표준 문서, 가이드라인 그리고 테스트 요소를 개발 단계의 필요한 영역에 구성하였다. 이를 통해 빠른 개발속도와 더불어 향후 유지보수비용을 줄일 수 있는 기회를 제공해주며 Scrum과 XP 개발방법론이 공통적으로 가지고 있거나 유사한 개념의 요소는 두 방법론 중에서 좀 더 명확하게 명시된 요소를 우선적으로 고려한다.

제안 MWeb-프레임워크 프로세스 모델의 각 단계별 흐름은 다음과 같다.

1) Planning Game

프로젝트 팀은 사용자와 함께 모바일 사용자 스토리(Mobile User Story)를 작성하게 되는데 이를 통해 모바일 테스트 시나리오가 산출된다. 모바일 사용자스토리를 바탕으로 요구사항들을 Product Backlog로 쌓는다.

2) Release Planning

모바일 사용자 스토리를 바탕으로 만든 요구사항들을 한번의 Iteration단위로 나누고 필요할 경우 구동환경이나 스펙 등을 정의하고 릴리즈계획을 세운다.

3) Sprint Backlog

Sprint Backlog는 한번의 Iteration에 수행되는 단위로 실제적으로 구현 하는 단계라고 할 수 있다. 이때 XP의 다양한 Practice를 통하여 Iteration을 수행한다.

4) Acceptance Test

Iteration이 수행된 후 사용자와 함께 Acceptance Test를 통해 앞서 정의된 모바일 웹 테스트 시나리오를 바탕으로 하여 검토를 수행하며 모바일 웹 표준 준수여부를 검증한다. 추가적으로 Refactoring을 수행할 수 있다.

5) Small Release

모바일 사용자와 W3C mobileOK Basic Test의 요건을 충족한다면 Small Release를 한다. 개발자와 모바일 사용자가 서로 다른 의견을 가지고 있다면 다시 Iteration을 수행하게 된다.

6) Next Product Backlog Sprint

하나의 Small Release를 수행하였으면 다음 Product Backlog가 Sprint에 들어가게 된다. 애플리

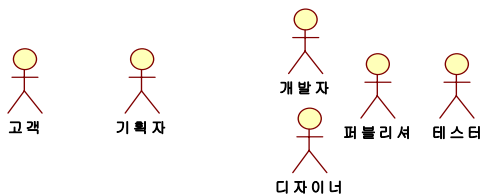
케이선의 크기에 따라 Iteration을 반복하여 개발을 수행한다.

3.3 적용 사례

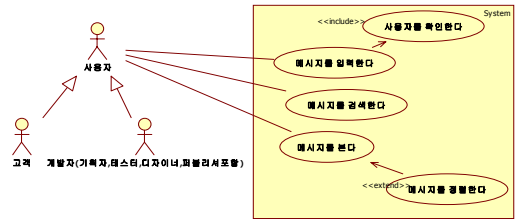
본 논문에서는 제안한 MWeb-프레임워크를 적용하여 개발자를 위한 모바일 웹 애플리케이션을 구현하는 것을 사례연구로 수행한다. 본 애플리케이션은 웹 애플리케이션을 개발하고자 할 때 커뮤니케이션을 통합하여 협업을 지원하기 위한 도구이며 다양한 환경에서 사용할 수 있도록 모바일 웹 표준을 준수하는 웹 애플리케이션이다. 개발한 애플리케이션을 ‘DeCA(Developer Communication Application)’라고 칭한다. 이를 통해 항상 온라인 상태를 유지하여 프로젝트의 진행을 원활하게 하고 의사소통을 효율적으로 수행할 수 있도록 해준다.

DeCA를 개발하기 위해 첫 번째로 수행해야 할 단계는 모바일 유저 역할을 식별하는 것이다. 모바일 유저를 식별하는 단계에서부터 첫 Sprint를 수행하기 이전에는 모든 프로젝트 인원들과 고객이 함께 토의하고 내용을 작성한다. 여기서는 기본적으로 개발자들이 도구를 사용하는 중심 사용자이므로 모바일 웹 개발자의 역할별로 분류하고 프로젝트를 의뢰하는 고객으로 분류한다. 개발자의 역할 분류는 기본적으로 실전 웹 표준 가이드 지침을 바탕으로 한다[11]. 본 애플리케이션에서는 6가지의 역할로 분류하며 그림 5와 같이 나타낼 수 있다.

이에 대한 상호간의 이해를 위하여 각 유저별 정의하는 역할 모델링을 수행하며 각 역할별로 정의된 내용은 표 1과 같다. 역할모델링은 표 1과 같은 방법 이외에도 그림 6과 같이 UML기반의 유즈 케이스 다이어그램과 정의서로 나타낼 수도 있다.



(그림 5) 모바일 Stakeholder



(그림 6) DeCA 유즈케이스

(표 1) DeCA 모바일 Stakeholder 모델링

역할	내용
고객	모바일 웹 애플리케이션을 개발할 때 의뢰를 하는 사람. 별도의 의뢰가 없는 경우는 기획자 및 개발자가 고객의 역할에 대응함. 다양한 요구사항이나 추가되는 기능을 빠르게 전달하기 위해 도구를 사용. 컴퓨터에 능숙하지 않을 수 있음.
기획자	모바일 웹 애플리케이션 의뢰를 받아 팀을 꾸리고 프로젝트를 진행할 사람. 기획자는 항상 프로젝트의 진행상황을 확인할 수 있도록 고객과 같이 PC 및 모바일 디바이스를 통해 접속을 할 수 있음. 고객의 요구사항을 확인하여 프로젝트에 반영하고 개발과정에서 문제점을 각 유저에게서 부터 보고받아 해결책을 제시함. 고객과 개발자들 간의 소통과 조율을 수행함.
디자이너	디자이너는 기본적으로 PC환경 및 모바일 디바이스 등의 다양한 환경에서 유저가 편리하게 이용할 수 있도록 디자인하는 역할을 수행하기 위해 개발자와 퍼블리셔와 함께 커뮤니케이션을 하기 위해 사용함.
개발자	개발자는 로직레벨을 처리하기 위한 프로그래머로 개발과정에서 디자이너와 퍼블리셔와의 협업을 수행하기 위해 도구를 사용함.
퍼블리셔	퍼블리셔는 웹 표준 개발을 수행할 수 있도록 XHTML 과 CSS를 통한 구조화를 수행하고 표준을 이행하는 코드가 될 수 있도록 디자이너와 개발자와 함께 협업을 수행하는 사람. 명세 되어 있는 모바일 웹 표준을 준수하며 Validator를 통해 검증하는 과정에 있어서 문제점을 기획자나 개발자, 디자이너와 커뮤니케이션하기 위해 도구를 사용함.
테스터	테스터는 고객의 요구사항이 제대로 적용되었는지, 초기명세와 같은 구현이 이루어졌는지 등을 확인하고 검증하여 문제점이 있을 경우 해당하는 역할유저에게 보고하거나 해결하는 사람. 개발과 관계된 모든 역할들과 협업을 수행해야 하며 각 역할 별로 따로 테스팅을 하도록 할 수도 있음.

역할 모델링을 수행한 후에는 스토리카드를 작성한다. 앞의 과정과 마찬가지로 모든 프로젝트 참여 인원이 수행을 하며 사용자 역할별로 작성을 하거나 이와 무관하게 작성할 수도 있다. 표 2는 스토리카드의 내용을 정리한 모바일 유저 스토리이다.

(표 2) DeCA 모바일 유저 스토리

스토리 텍스트
<ul style="list-style-type: none"> • 도구는 로그인을 통해 사용자의 구분을 한다.(실명, 아이디, 패스워드, 역할, 이메일) • 사용자는 요구사항을 입력할 수 있다. • 기획자는 모든 유저에게 메시지를 전달할 수 있다. • 기획자는 프로젝트를 생성하고 삭제할 수 있다. • 모든 글에는 고유번호가 존재하여 고유번호를 통해 연결 글을 만들 수 있다. • 사용자는 PDA나 휴대전화 PC등 웹브라우저가 존재하면 도구를 사용할 수 있다. • 도구는 다양한 웹브라우저에서 정상적으로 나타나야 한다. • 사용자가 메시지를 입력을 할 때에는 다음과 같은 속성을 적용할 수 있다. • 제목, 설명, 받는 사람(검토자), 중요도(긴급,중요,보통,낮음), 태그, 메시지 종류(공지, 제안, 결함, 질의, 보고, 기능 확장, 문제점), 마일스톤(이터레이션), 버전, 참조자, 확인여부, 파일첨부 • 새 메시지를 작성할 때 중요도가 긴급으로 설정될 경우는 받는 사람의 휴대폰으로 확인메시지를 보내며 해당사용자가 로그인 중일경우 알림 기능을 수행한다. • 작성한 메시지가 정상적으로 확인이 되었을 경우 완료되었음을 지정할 수 있다. • 사용자는 다양한 검색기능을 통해 메시지를 찾을 수 있다. • 사용자는 메시지에 댓글/답글을 달 수 있다. • 메시지내용에는 다양한 인용 및 하이퍼링크를 지원한다.(코드의 경우 별도표시) • 도구는 어느 환경에서든 동일한 사용자경험을 제공해줄 수 있도록 디바이스간 통일된 UI를 제공한다. • 도구는 모바일 디바이스에서도 10초 이내에 로딩이 가능해야 한다.(텍스트위주의 디자인) • 도구는 다음과 같은 모바일 웹 표준을 준수한다. • XHTML 1.1 Basic(W3C), CSS Mobile Profile 2.0(W3C), Mobile Web Practices 1.0(W3C) • PC환경의 사용자를 위해서는 다음과 같은 웹 표준을 준수한다. • XHTML 1.1 (W3C), CSS 2.0 (W3C) • 도구는 다음의 웹 브라우저에서 정상 동작해야 한다. • 모바일 디바이스 : Opera Mini, Minimo • PC : Internet Explorer, Firefox

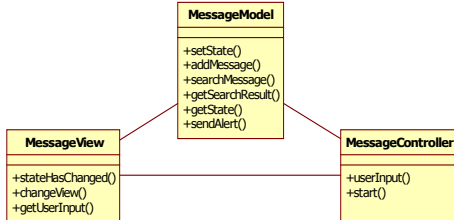
모바일 유저 스토리가 완성이 되면 이를 바탕으로 스토리 추정을 수행한다. 이와 더불어 모바일 웹 테스트 시나리오도 작성한다. 스토리 추정 후 Iteration 단위로 추정된 스토리를 나누고 우선순위를 정하여 Release Planning을 수립할 수 있다. 표 3은 스토리 추정을 거쳐 Release Planning이 산출된 것을 나타내고 괄호안의 숫자는 스토리 추정치(주단위)를 나타낸다. 그리고 나서 우선순위를 부여하여 우선적으로 Sprint에서 수행할 스토리와 다음 Iteration에서 수행할 수도 있는 스토리를 고려한다.

(표 3) 완료된 DeCA의 Release Planning

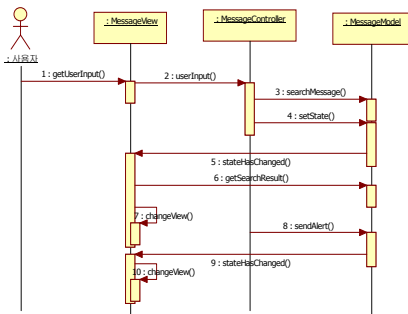
Iteration 1	우선 순위	Iteration 2	우선 순위
도구는 로그인을 통해 사용자의 구분을 한다.(1)	1	사용자는 메시지에 댓글/답글을 달 수 있다.(1/2)	1
사용자는 요구사항을 입력할 수 있다.(1)	1	기획자는 모든 유저에게 메시지를 전달할 수 있다.(2)	2
사용자가 메시지를 입력을 할 때에는 여러 속성을 적용할 수 있다.(2)	1	새 메시지를 작성할 때 중요도가 긴급으로 설정될 경우는 받는 사람의 휴대폰으로 확인메시지를 보내며 해당사용자가 로그인 중일경우 알림 기능을 수행한다.(3)	2
기획자는 프로젝트를 생성하고 삭제할 수 있다.(1)	1		
모든 글에는 고유번호가 존재하여 고유번호를 통해 연결글을 만들 수 있다.(1)	2	사용자는 다양한 검색기능을 통해 메시지를 찾을 수 있다.(2)	2
작성한 메시지가 정상적으로 확인이 되었을 경우 완료되었음을 지정할 수 있다.(1/2)	2	메시지내용에는 다양한 인용 및 하이퍼링크를 지원한다.(1)	3

Release Planning을 바탕으로 Sprint에 들어가기 이전에 구현할 애플리케이션을 설계한다. 이때에는 CRC 카드를 이용하거나 UML기반의 클래스 다이어그램을 이용할 수도 있다. 본 논문에서는 그림 7과 같이 클래스 다이어그램으로 설계하였

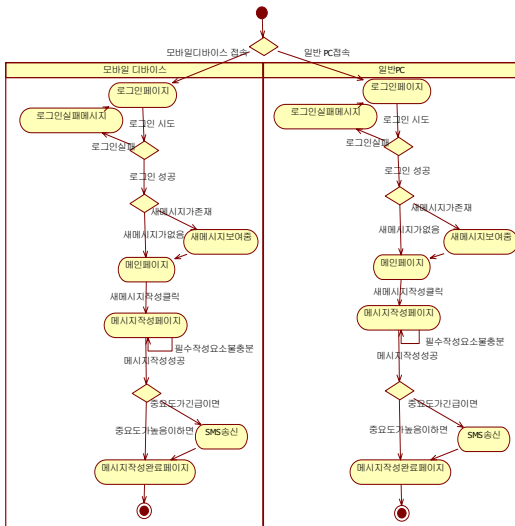
고 이를 바탕으로 그림 8과 같이 시퀀스 다이어그램을 작성한다. MWeb-프레임워크는 Ruby on Rails의 경우 MVC 모델을 바탕으로 하므로 클래스 다이어그램도 그에 맞추어 설계한다.



(그림 7) DeCA 클래스 다이어그램



(그림 8) DeCA 시퀀스 다이어그램



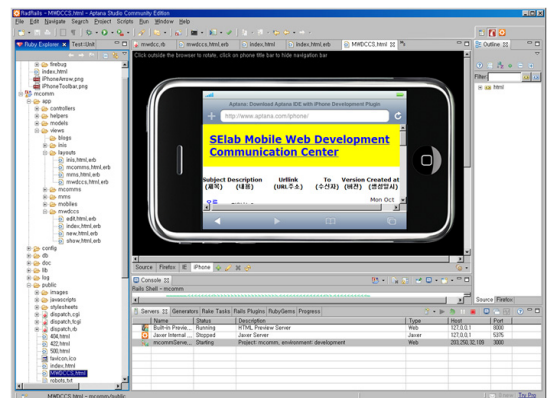
(그림 9) DeCA Activity Diagram

그림 9는 DeCA의 액티비티 다이어그램이다. 본 애플리케이션의 경우 기본적으로 다양한 디바이스에서 동일할 결과물을 얻는 것을 목표로 하고 있으므로 PC 기반에서 구동내용과 모바일 디바이스(PDA, 휴대전화)에서도 정상적인 구동이 가능한 흐름을 구성하기 위해 두 흐름으로 구성 된다.

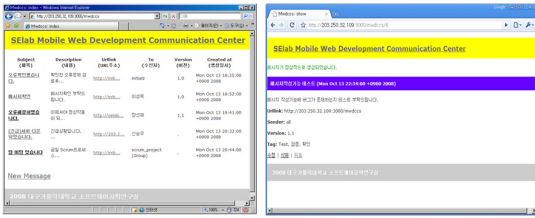
3.4 DeCA 실행 예

본 절에서는 개발한 애플리케이션을 다양한 환경에서 구동하면서 동작 상태를 확인한 결과를 통해 애플리케이션의 구동상태 및 개발한 모바일 웹 애플리케이션의 간단한 내용을 기술한다. 상단은 PC 기반의 웹 브라우저에서의 실행 예이며 하단은 모바일 디바이스에서의 웹 브라우저 실행 예이다. 개발환경인 PC에서 다른 웹 브라우저를 통해 정상적인 결과를 확인하고 PDA 및 휴대 전화에 웹 브라우저에서도 정상적인 결과화면이 나타나는 것을 확인할 수 있다.

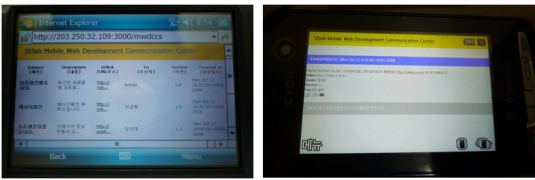
그림 10은 통합개발환경 내부에서 모바일 에뮬레이터를 통해 정상적인 동작여부를 확인하는 것을 나타내고 있다. 별도의 외부 웹 브라우저 없이 통합개발환경내부에서 편리하게 동작여부를 확인할 수 있다.



(그림 10) 통합개발환경 모바일 에뮬레이터 실행 예(Aptana IDE, iPhone)



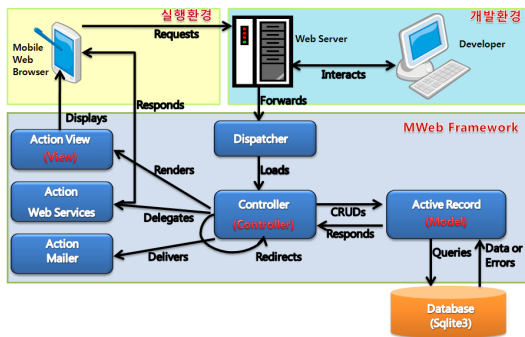
(그림 11) 개발환경 실행 예(PC, Windows XP, Internet Explorer 7, Chrome)



(그림 12) 실행환경 실행 예(PDA, Windows Mobile 6, Internet Explorer)

그림 11은 개발환경에서 다양한 웹 브라우저를 통해 애플리케이션의 정상적인 동작여부를 확인하는 실행 예이다.

그림 12는 모바일 디바이스의 정상적인 동작을 확인하기 위해 PDA와 휴대전화를 통해 개발한 애플리케이션을 실행한 예이다.



(그림 13) MWeb-프레임워크 기반 MVC 구동환경

그림 13은 개발한 모바일 웹 애플리케이션의 구동 구조를 나타낸다. 기본적으로는 MVC 형태를 따르고 있다[9]. 개발환경에서 에뮬레이터나 PC기반의 웹 브라우저로 확인 및 테스트를 수행

하고 모바일 디바이스 웹 브라우저로 실행환경에서 테스트를 수행할 수 있다.

다이어그램을 바탕으로 하여 Sprint를 수행하여 개발한 애플리케이션을 개발환경 내부 에뮬레이터에서 동작을 확인하면 그림 10과 같은 결과를 얻을 수 있다. 한 단계의 Iteration이 수행된 후 Acceptance Test를 수행하여 Small Release를 한다.

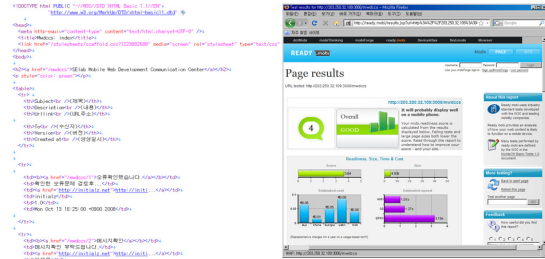
표 4는 구현된 DeCA 애플리케이션에 대한 mobileOK 검증여부를 확인하기 위해 mobileOK basic tests 1.0문서를 통한 항목별 평가결과이다. 총 24개의 항목 중 1개의 요소(TABLES_ALTERNATIVES)를 제외한 모든 요소가 수동 검증결과 합격판정을 받았으며 TABLES_ALTERNATIVES 항목의 경우 테이블 요소를 사용하였기에 경고요소(Warn)가 존재하지만 표준에서는 사용 금지(Fail)요소가 아니기 때문에 mobileOK 인증을 받을 수 있다는 결과를 얻을 수 있다. 스타일시트 항목의 경우 다양한 디바이스에서 구동을 지원하기 위해서는 반드시 지원할 필요성이 있다.

4. MWeb-프레임워크 평가

4.1 검증 및 성능

앞 절에서 DeCA의 정상적인 동작여부를 확인 하였으나 모바일 웹 표준의 준수여부는 브라우저 내부에서는 확인 할 수 없다. 따라서 본 절에서는 모바일 웹 표준 준수 여부를 쉽게 확인 할 수 있도록 개발된 mobileOK Checker library를 이용하여 mobileOK 여부와 성능평가를 동시에 수행할 수 있도록 지원하는 도구인 ready.mobi를 통해 도구의 검증을 수행한다. 그림 14는 검증결과를 나타내고 있는데 좌측은 검증한 XHTML Basic 1.1에 맞춰 작성된 소스코드이며 우측이 ready.mobi를 이용한 검증결과화면이다. 검증내용은 mobileOK 테스트요소를 통과하는지를 검증하고 페이지의 사이즈와 페이지 렌더링 성능 및 국가별 로딩속도를 확인하는 요소들로 이루어져 있다. 수행결과

Page Result는 Good으로 측정되었고 평점 4점으로 우수한 성능과 mobileOK 인증을 통과할 수 있음을 확인하였다.



(그림 14) mobileOK Checker(ready.mobi)를 이용한 검증 및 성능 평가

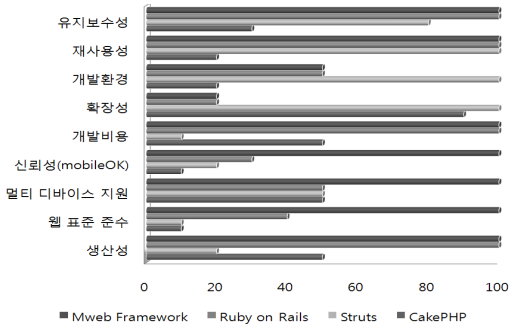
(표 4) mobileOK Basic Tests 1.0 항목 평가

항목	내용	결과
AUTO_REFRESH and REDIRECTION	자동 갱신이나 이동 항목 사용금지	없음 (OK)
CACHING	Meta-equiv를 통한 캐싱 지원	선언됨 (OK)
CHARACTER_ENCODING_SUPPORT and CHARACTER_ENCODING_USE	문자열 인코딩(utf-8)	UTF-8을 통한 인코딩 (OK)
CONTENT_FORMAT_SUPPORT and VALID_MARKUP	문서포맷지원(DTD) : Basic, MP	DTD Mobile Basic 1.1을 통한 선언 (OK)
DEFAULT_INPUT_MODE	Inputmode의 사용 (Basic)	Inputmode 사용함 (OK)
EXTERNAL_RESOURCES	외부 리소스의 사용 금지 (10,20)	외부 리소스 사용안 함 (OK)
GRAPHICS_FOR_SPACING	추적,공간을 위한 이미지사용금지	사용 안 함 (OK)
IMAGE_MAPS	Input 엘리먼트의 image사용금지 img 엘리먼트 usemap,ismap금지	사용 안함 (OK)
IMAGES_RESIZING and IMAGES_SPECIFY_SIZE	이미지 리사이즈 사용금지	사용 안함 (OK)
LINK_TARGET_FORMAT	링크 타겟 문서의 검증	링크 타겟 문서 무결함(내부링크만 활용) (OK)
MEASURES	CSS규격에 관한 측정	CSS규격 준수 (OK)

항목	내용	결과
MINIMIZE	문서 외부요소의 정도 (10% : warn, 25%:fail)	10%를 넘지 않음 (OK)
NO_FRAMES	Frameset, iframe 엘리먼트 사용금지	사용 안 함 (OK)
NON-TEXT_ALTERNATIVES	이미지의 대체 텍스트 사용(alt)	이미지 사용 안 함 (OK)
OBJECTS_OR_SCRIPT	Script사용의 자제 권유	Script 사용 안 함 (OK)
PAGE_SIZE_LIMIT	페이지크기 제한 (10kb이하)	제한을 넘지 않음 (main:5313bytes) (OK)
PAGE_TITLE	페이지 이름 부여	페이지 이름 존재 (OK)
POP_UPS	팝업의 사용금지 (_self, _parent, _top)	팝업 엘리먼트 사용 안 함 (OK)
PROVIDE_DEFAULTS	Form등에서 기본값 부여	기본값 부여 해 줌 (OK)
STYLE_SHEETS_SUPPORT	스타일시트 지원	스타일시트 사용함 (OK)
STYLE_SHEETS_USE	스타일시트 사용	스타일시트 사용함 (OK)
TABLES_ALTERNATIVES	테이블 사용시 경고 요소(금지하는 아님)	테이블 일부 사용 (Warn)
TABLES_LAYOUT	테이블	규격 준수 (OK)
TABLES_NESTED	테이블 태그 중첩	테이블 중첩하지 않음 (OK)

4.2 평가

본 논문에서 제안한 애자일 프레임워크와 기존 웹 개발 프레임워크를 비교한 결과는 그림 15와 같다. 본 평가의 기준은 Tim Bray[12][13]의 연구 결과를 참고하여 비교한다. 비교항목에 대한 절대적인 수치화가 불가능하기에 상대적인 비교내용을 그래프로 나타내었다. MWeb- 프레임워크는 Ruby on Rails를 기반으로 제안된 모바일 웹 개발 프레임워크이므로 기본적인 특성은 Ruby on Rails를 따른다. 그래프(그림 15)에서 나타나는 것과 같이 기존 웹 프레임워크보다 확장성을 제외한 분야에서 우수한 평가결과를 얻었으며 모바일 웹 개발의 요소도 추가적으로 고려하였기에 모바일 디바이스에서의 동작도 보증할 수 있다.



(그림 15) 기존 웹 개발 프레임워크와의 평가

각 평가 항목별 평가내용은 아래와 같다.

- 유지보수성 : PHP언어의 특성상 코드와 HTML이 스펙계티 코드로 나타나는 특징을 가지고 있다. MWeb-프레임워크는 기본적으로 MVC 모델을 기반으로 하고 있고 간결한 코드가 작성된다.
- 재사용성 : 유지보수성과 유사한 의미에서 재사용성의 측면도 고려할 수 있다.
- 개발환경 : 개발환경의 경우는 가장 다양한 개발환경을 지원하는 Java 진영이 탁월하다. MWeb-프레임워크의 경우는 Eclipse기반의 Radrails라는 개발환경을 적용하고 있다.
- 확장성 : 확장성은 본 프레임워크가 해결해야 할 제 1의 과제이다. Java기반의 환경이 확장성은 우수하나 웹의 경우 고도의 연산을 처리하지 않는 애플리케이션이 많기에 극복 가능한 요소이다.
- 개발비용 : 개발비용은 결국 목표 애플리케이션의 개발시간과 인력과 비례한다. 간결한 코드와 뛰어난 DB와의 연동기법은 개발 외적인 환경구성에 대한 초기비용을 획기적으로 줄여준다
- 신뢰성(mobileOK) : 모바일 웹 애플리케이션 개발에 특화되어 있으므로 애플리케이션을 개발하고 사용하는 측 모두에게 신뢰성을 안겨줄 수 있는 요소이다.
- 멀티 디바이스 지원 : Multiple View라는 특징을 지원하고 있어 모바일 디바이스와 동

시에 PC기반, 기타 플랫폼에 맞춘 웹 애플리케이션을 개발가능하다.

- 웹 표준 준수 : 기본적으로 MWeb-프레임워크는 XHTML기반으로 작성이 되므로 DTD나 CSS profile의 일부요소만 고려하면 되므로 모바일 웹 표준을 준수한다.
- 생산성 : 코드를 기준으로 PHP의 코드는 신속하지만 복잡하고 난해한 코드로 이루어지거나 MWeb-프레임워크의 경우 신속하면서도 깔끔한 코드를 획득할 수 있다.

평가과정에서 MWeb-프레임워크가 개선해야 할 요소들도 확인할 수 있었다. 향후 개선이 요구되는 사항은 아래와 같다.

모바일 웹 표준과 모바일 웹 브라우저의 불일치(완벽한 표준의 미지원)

- XHTML Basic 표준을 지원하는 전용에디터의 부재
- Ruby on Rails의 짧은 역사로 인한 개발과정간 문제 발생 시 대처정보 부족
- 모바일 디바이스 및 모바일 웹 브라우저의 급변으로 인한 잦은 수정 필요

본 프레임워크가 향후 많은 사용자가 모바일 디바이스를 통한 웹을 사용하면서 개발방법론을 적용하는 측면에서는 변화에 능동적으로 대응하여 프레임워크의 적용이 필요하다.

5. 결론 및 향후 연구

모바일 웹 환경에 적응적인 모바일 웹 표준을 준수하고 애자일 개발방법론을 적용한 Ruby on Rails 기반의 모바일 웹 개발 프레임워크인 MWeb-프레임워크를 제안하고 제안한 프레임워크를 통한 사례연구를 수행하였다. MWeb-프레임워크는 다양한 디바이스의 성능을 가진 모바일 디바이스를 지원하기 위한 모바일 웹 애플리케이션 개발 프레임워크로 사용자의 요구사항을 신속

하게 반영하면서도 개발속도와 품질을 모두 고려할 수 있는 장점을 가진 프레임워크이다. 제안한 MWeb-프레임워크를 적용 사례연구를 통해 모델의 우수성을 검증하였고 경량화 애자일 프레임워크를 통해 생산성과 품질을 검증할 수 있다. 검증 결과 뛰어난 생산성과 신뢰성을 가지며 유지보수성과 재사용성, 그리고 저렴한 개발비용과 멀티디바이스 지원 등의 우수성을 가지는 프레임워크임을 확인하였다.

향후 연구로는 본 논문에서 제안한 MWeb-프레임워크의 표준화 검증을 통한 표준 인증을 획득하고 다양한 모바일 웹 애플리케이션 개발 사례를 통해 모델의 추가적인 문제점을 파악하여야 할 필요가 있다. 또한 통합개발환경의 W3C 모바일 웹 표준 준수 코드 검증 기능 플러그인을 개발하여 좀 더 효율적으로 표준화를 수행할 수 있도록 고려되어야 한다.

참 고 문 헌

- [1] 애니빌, “모바일 웹 산업 동향 및 개발 환경”, 애니빌, 2008. <http://www.anybil.com/vx>
- [2] Mobile Web Initiative, “W3C Mobile Web Initiative”, W3C Mobile Web Initiative, 2008. <http://www.w3.org/Mobile/>
- [3] 이승윤, “한국형 모바일 OK 표준화”, 모바일 웹 2008 컨퍼런스, 2007.
- [4] 강석천, 강규영, 김창준, “변화를 꿈꾸는 개발방법론 애자일(Agile)”, 월간 마이크로소프트웨어 2007년 3월호, 2007.
- [5] Ron Jeffries, “What is Extreme Programming?. XProgramming.com : an agile software development resource”, 2001. <http://www.xprogramming.com/xpmag/whatisxp.htm>
- [6] Ken Schwaber, “What is Scrum?, Scrum: Its about common sense”, 2007. <http://www.controlchaos.com/about/>
- [7] Kent Beck, “Extreme Programming Explained : Embrace change”, 2nd ED, Pearson Education, 2005.
- [8] Wikipedia, “Web application framework”, Wikipedia, 2008. http://en.wikipedia.org/wiki/Web_application_framework/
- [9] David Heinemeier Hansson, “Ruby on Rails”, Ruby on Rails, 2008. <http://www.rubyonrails.org/>
- [10] Dave Thomas, David Heinemeier Hansson, “Agile Web Development with Rails 2/E”, Pragmatic Programmers, 2006.
- [11] 윤석찬, 신정식 외, “실전 웹 표준 가이드”, 한국소프트웨어진흥원, 2005.
- [12] Tim Bray, “Issues in web frameworks”, Tim Bray Ongoing, 2008. <http://www.tbray.org/>
- [13] Tim Bray, “Comparing Frameworks”, Tim Bray Ongoing, 2006. <http://www.tbray.org/ongoing/When/200x/2006/11/10/Comparing-Frameworks>

● 저 자 소개 ●



신 승 우

2007년 대구가톨릭대학교 컴퓨터공학과 (공학사)

2007년~2009년 대구가톨릭대학교 컴퓨터정보통신공학과 석사(공학석사)

관심분야 : SOA, Agile methodologies, Mobile Web Development, Web Frameworks, MDA

E-mail : selab@cu.ac.kr



김 행 곤

1985년 중앙대학교 전자계산학과(공학사)

1987년 중앙대학교 대학원 전자계산학과(공학석사)

1991년 중앙대학교 대학원 전자계산학과(공학박사)

1978년~1979년 미 항공우주국 객원 연구원

1987년~1989년 한국전기통신공사 전임연구원

1988년~1989년 AT&T 객원 연구원

1990년~현재 대구가톨릭대학교 컴퓨터공학과 교수

2000년~2002년 Central Michigan University 교환교수

2007년~2008년 미 SEITI 연구소 객원 연구원

관심분야 : 모바일 임베디드 시스템 개발기법, 모바일 시스템 모델링, 컴포넌트 기반 개발 기법,

임베디드 소프트웨어 테스팅, 프로세스 개선 공학, 요구공학 및 도메인 공학

E-mail : hangkon@cu.ac.kr