

모바일 웹 표준 기반 경량화 프레임워크1)

신승우*, 박은주*, 김행곤*

*대구가톨릭대학교 컴퓨터정보통신공학부
e-mail:{selab, ejpark, hangkon}@cu.ac.kr

Agile Framework for Mobile Web Standards

Seung-Woo Shin*, Eun-Ju Park*, Haeng-Kon Kim*

*Dept. of Computer information & Communication Engineering
Catholic Univ. of Daegu, Korea

요 약

최근 모바일 디바이스의 성능향상으로 모바일 웹 애플리케이션 구동환경이 개선되고 있다. 이에 따라 모바일 웹 애플리케이션 개발 수요가 국내외적으로 증가하고 있다. 하지만 기존 PC환경과 달리 개발 환경 및 표준에 대한 고려가 적어 많은 개발비용이 소모되고 있다. 본 논문에서는 MWeb-프레임워크를 통해 최근 각광 받고 있는 웹 개발 프레임워크 기반의 개발방법론을 제안하며 이와 더불어 다양한 디바이스에서 동일한 사용이 가능하도록 모바일 웹 표준을 준수하는 개발방법론을 제안한다. Ruby on Rails기반의 모바일 웹 개발 아키텍처를 정의하고 이를 바탕으로 모바일 웹 표준을 준수하는 애자일 개발방법론을 적용한 프로세스 모델을 제안하고 기존 웹 개발 프레임워크와 개발방법론을 비교, 평가한다. 제안한 프레임워크를 통하여 신속한 개발을 수행할 수 있으며 모바일 웹 표준 환경을 준수하여 모바일 웹 애플리케이션의 품질과 신뢰성을 획득할 수 있다.

1. 서론

최근 모바일 웹 환경이 변화하고 있으며 모바일 디바이스가 빠르게 보급됨으로 인해 모바일 디바이스의 동작 환경은 나날이 개선되고 있다. 이로 인해 모바일 디바이스의 통신방식이 TCP/IP 기반의 HTTP의 사용으로 PC 환경과 유사한 웹 환경을 갖추게 되었다[1, 2].

개선되고 있는 모바일 웹 환경과 달리 모바일 웹 개발에 관한 연구는 아직 미흡하다. 모바일 디바이스에서 HTTP 기반의 모바일 웹 애플리케이션에 특화된 개발방법론에 대한 연구가 미비하다. 더군다나 웹 개발 프레임워크 기반으로 개발된 모바일 웹 표준을 준수하는 애플리케이션은 아직까지 찾아보기 힘들다[3]. 본 논문에서는 루비 온 레일스(Ruby on Rails) 웹 프레임워크 기반 모바일 웹 개발방법론을 제안한다. 다양한 디바이스에서 정상적인 동작이 가능하도록 지원하기 위하여 모바일 웹 표준을 적용하여 개발을 지원한다. 또한 점점 빠르게 변화하고 있는 사용자의 요구사항을 신속하게 반영하여 애자일 개발방법론을 통해 생산성 및 품질을 보증할 수 있도록 지원한다.

2. 관련연구

2.1 웹 개발 프레임워크

최근 들어 다양한 언어를 기반으로 한 웹 개발 프레임

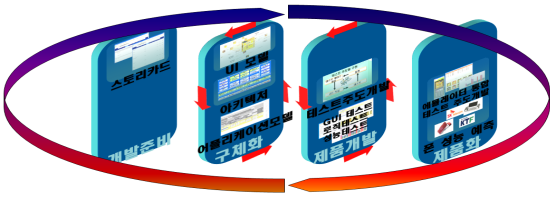
워크들이 각광을 받고 있으며 그중에 최근 많은 관심을 받고 있는 웹 개발 프레임워크는 아래와 같다.

- Struts(Java) : 비교적 오랜 역사를 가진 웹 프레임워크로 뷰와 비즈니스 로직을 분리하여 구현하여 유지보수를 용이하도록 지원하는데 초점을 맞춘 Java기반의 프레임워크이다[4].
- Ruby on Rails(Ruby) : Ruby on Rails는 스크립트형 객체지향 언어인 Ruby로 작성된 MVC 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 MVC템플릿을 자동 생성하는 기능이 있다[5, 6].
- CakePHP(PHP) : 쉬운 사용과 HTML의 처리에 강점을 가지고 있어 양한 오픈소스프로젝트에 사용되어 지는 PHP기반 프레임워크로 Ruby on Rails의 인기에 영향을 받아 개발된 웹 프레임워크이다[4].

2.2 MASAM

MASAM은 아키텍처 기반 모바일 응용 소프트웨어 개발방법론으로 모바일 애플리케이션을 개발함에 있어 애자일 개발방법론을 적용하여 경량화된 개발프로세스를 제공하는 개발방법론이다. 이를 통해 개발기간을 단축시키고 웹기반의 프로세스와 작업등을 공유할 수 있다. MASAM은 네 단계로 구성되며 개발준비→구체화→제품개발→제품화의 순서로 이루어지게 되며 이는 그림 1과 같다[7].

1) 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA-2009-(C1090-0902-0032))



(그림 1) MASAM 개발방법론의 개요

2.3 애자일 개발방법론

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 표준 프로세스에 가치를 두는 것과 달리 의사소통과 협동, 빠른 변화의 가치를 중시하는 90년대 후반에 태생된 개발방법론이다[8,9].

애자일 개발방법론들이 기존 방법론들과 다른 공통적인 특성은 다음과 같다[9] :

- 태생 : 경험에서 시작된 방법들이 많다.
- 가벼움 : 작은 단계로 시작해볼 수 있는 일들이 많다.
- 의사소통 : 대화와 협력을 통한 상호작용을 강조한다.
- 피드백에 기반한 적응적 시스템 : 짧은 반복(Iteration) 과정에서 얻은 경험으로 팀을 개선하도록 장려한다.

3. MWeb 프레임워크(MWeb-Framework)

3.1 프레임워크 아키텍처

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 아키텍처는 그림 2와 같다.

각 계층별로 간단히 설명하면 다음과 같다.

- Hardware - X86 Series : Ruby on Rails는 플랫폼에 독립적이므로 x86기반의 하드웨어가 아닌 환경에서도 동작하나 본 아키텍처에서는 개발환경으로 가장 보편적인 환경인 X86 Series 플랫폼을 기반환경으로 한다.
- Operating System - Any OS : x86기반 어떤 운영체제에서도 구동가능 하므로 자유선택이 가능하다.
- Web Server - Mongrel + Nginx : 개발과정에서는 잦은 변경이 발생하므로 프레임워크의 변화에 대응할 수



(그림 2) 모바일 웹 개발 아키텍처

있는 Mongrel을 개발환경 웹 서버로 운용하고 실제 애플리케이션의 배포에서는 경량화 웹 서버인 Nginx 웹 서버를 이용하여 응답속도를 향상 시킨다.

- DBMS - Sqlite3 : 경량화된 DBMS로 간단한 프로젝트에 많이 사용되고 있다. Rails 2.0의 경우 다른 DBMS에 비하여 편리하게 Sqlite3의 사용이 가능하다.
- Ruby on Rails Framework : 현재 Ruby on Rails 2.x에서 모바일 환경을 위한 Multiple View를 지원하여 좀 더 효율적인 개발이 가능하다.
- MWeb-Framework : MWeb-프레임워크는 본 아키텍처의 핵심요소로 개발환경의 일체를 제공하는 레이어로 하위 레이어들을 기반으로 하여 모바일 웹 애플리케이션을 개발할 수 있는 최적의 환경을 제공하기 위한 요소들로 구성된다. 각 요소 항목을 살펴보면 다음과 같다.
 - IDE - Aptana Studio(Radrails) : Ruby on Rails의 경우 별도의 통합 개발환경이 없이도 신속한 개발이 가능하나 형상관리 도구나 기타 테스트 환경을 고려하여 통합개발환경을 사용한다.
 - Mobile Device, Mobile Web Browser : 본 프레임워크는 모바일 디바이스에서의 동작을 보증해야 하므로 모바일 디바이스와의 연동이 필수적이다. 또한 개발과정에서 실행환경 테스트가 필요하므로 타겟 디바이스의 아키텍처 포함의 필요성이 있다.
 - Developer : 개발활동에 있어 가장 중심이 되는 것은 사람이다. 본 요소는 개발자들을 위한 환경을 구축해야 함을 나타내는 요소이다.
 - Validation Tool : 제안 아키텍처에서는 W3C의 모바일 웹 표준의 검증을 수행하는 mobileOK checker나 ready.mobi등의 검증도구로 표준화 검증을 수행한다.
 - Device Emulator : Aptana Studio에서는 모바일 디바이스를 위한 에뮬레이터 플러그인을 설치 할 수 있도록 지원한다. 모바일 디바이스의 Eclipse플러그인을 통하여 개발환경의 테스트가 가능하다.
 - Web Browser : 모바일 디바이스 이외에 PC 환경에서도 애플리케이션의 구동이 가능해야하므로 개발환경 내부 웹 브라우저를 통해 테스트를 해야 한다.

3.2 프로세스 모델

제안 MWeb-프레임워크 프로세스 모델의 각 단계별 흐름은 아래와 같으며 그림으로 나타내면 그림 3과 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. 이때 W3C Mobile Web Best Practices의 요소를 고려하여 작성하여야 한다.

2) Release Planning

모바일 사용자 스토리를 바탕으로 만든 요구사항들을 한번의 Iteration 단위로 나누고 필요할 경우 Architectural Spike나 Spike를 통해 구동환경이나 스펙 등을 정의하고 릴리즈에 대한 계획을 세운다. 이때 앞서 식별된 구현 불가능한 요소나 우선 구현 필요성이 낮은 단위는 계획에서 제거하거나 우선순위를 낮추어 차후 구현으로 Product Backlog의 차후 구현 순위로 변경한다. 만약 기존 개발된 요소를 변환해야 하는 경우 W3C Content Transformation Guideline을 참조하여 필요에 따라 변환하여 재사용한다.

3) Sprint Backlog

스프린트 백로그(Sprint Backlog)는 한번의 Iteration에 수행되는 단위로 실제적으로 구현 하는 단계라고 할 수 있다. 이때 XP의 다양한 Practice를 통하여 Iteration을 수행한다. 한번의 Iteration이 끝나면 해당하는 스프린트에서 수행된 결과를 바탕으로 다음 스프린트 백로그를 만들어내게 된다. Iteration동안 Test Scenario를 바탕으로 테스트를 수행하고 스프린트 단계에서는 모바일 웹 표준의 W3C XHTML Basic 1.1 과 W3C CSS Mobile 2.0 Profile을 참조하여 구현한다.

4) Acceptance Test

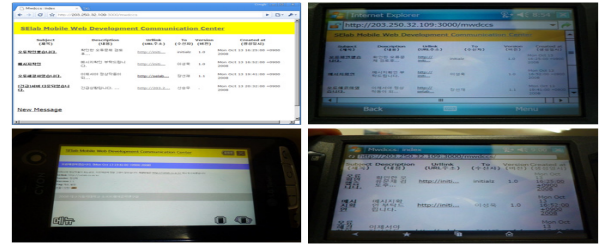
Iteration이 수행된 후 사용자와 함께 Acceptance Test를 통해 앞서 정의된 모바일 웹 테스트 시나리오를 바탕으로 하여 W3C mobileOK Basic Test와 함께 검토를 수행하며 W3C mobileOK Checker를 이용하여 모바일 웹 표준 준수여부를 검증한다. 추가적으로 Refactoring을 수행할 수 있다.

5) Small Release

모바일 사용자와 W3C mobileOK Basic Test의 요건을 충족한다면 Small Release한다.

6) Next Product Backlog Sprint

하나의 Small Release를 수행하였으면 다음 Product Backlog의 아이템이 스프린트에 들어가게 된다. Small Release 후, 모바일 웹 표준을 준수하지 못하는 요소가 모바일 사용자의 요구사항의 수용이나 모바일 웹 브라우저의 미지원 등의 이유로 발생할 경우 문제요소를 Product Backlog에 기록하여 유지보수 수행을 용이하게 한다.



(그림 4) 다양한 구동환경에서의 실행 예



(그림 5) mobileOK Checker를 통한 검증 및 성능 평가

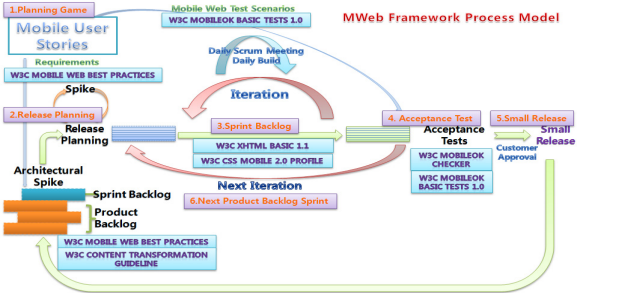
3.3 모바일 웹 지원 프레임워크 적용

그림 4는 MWeb-프레임워크를 적용하여 프로토타입 애플리케이션을 구현한 예이다. PC, PDA, 휴대전화등 다양한 디바이스에서 개발한 애플리케이션의 동작여부를 테스트하고 이를 통해 신속한 개발과 표준을 준수하는 애플리케이션을 제공할 수 있다. 그림 5는 애플리케이션의 모바일 웹 표준 준수를 검증하고 성능을 평가하는 도구인 ready.mobi를 통해 검증 및 신뢰도를 확인할 수 있다.

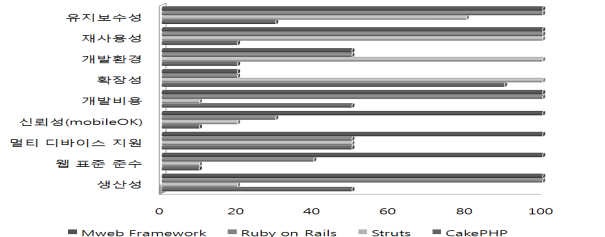
4. 평가

4.1 기존 웹 개발 프레임워크와의 평가

- 각 평가 항목별 평가내용은 아래와 같다.
- 유지보수성 : MWeb-프레임워크는 기본적으로 MVC 모델을 기반으로 하고 있고 간결한 코드가 작성되므로 차후 식별성이라는 측면에서 탁월하다.
- 재사용성 : CakePHP의 경우 이 부분에 대한 해결책을 제시하고 있으나 PHP의 언어 특성상 개발자의 노력이 필요한 부분이 많다.
- 개발환경 : 개발환경의 경우는 가장 다양한 개발환경을 지원하는 Java 진영이 탁월하다. MWeb-프레임워크의 경우는 Eclipse를 기반으로 한 Aptana Studio에 Radrails라는 Rails 프레임워크 플러그인을 통한 개발 환경을 적용하고 있으나 아직까지 버전별 환경지원 등의 문제점들이 존재한다.



(그림 3) 모바일 웹 표준 기반 애자일 프로세스 모델



(그림 6) 기존 웹 개발 프레임워크와의 평가

- 확장성 : 확장성은 본 프레임워크가 해결해야 할 제 1의 과제이다. 만일 데이터의 연산이 필요할 경우 단연 Java기반의 환경이 우수하다. 하지만 모바일 웹 애플리케이션의 경우는 이러한 확장성에 대한 고려도가 낮더라도 전체적인 성능저하 요소는 적을 것이다.
- 개발비용 : 간결한 코드와 뛰어난 DB와의 연동기법은 초기비용을 획기적으로 줄여주며 프로토타이핑을 수행하여 짧은 시간 안에 가능하며 개발비용을 줄여준다.
- 신뢰성(mobileOK) : mobileOK 인증을 통해 애플리케이션을 개발하는 측과 사용하는 측 모두에게 신뢰성을 안겨줄 수 있는 요소라고 볼 수 있다.
- 멀티 디바이스 지원 : 모바일 웹에 대한 지원을 Ruby on Rails 버전 2.x에서부터 고려하고 있다. 기본적으로는 CSS를 통한 디바이스별 환경에 대한 처리가 가능하나 Multiple View를 지원하고 있어 모바일 디바이스와 동시에 다양한 플랫폼에 맞춘 웹을 개발가능하다.
- 웹 표준 준수 : 모바일 웹 기반의 환경에 최적화 하기 위해 XHTML Basic의 코드로 일부 변경해야 하나 DTD나 CSS profile의 일부요소만 고려하면 되므로 다른 프레임워크에 비해서 우수하다.
- 생산성 : Ruby on Rails의 가장 큰 특징요소는 초고속 웹 개발이다. 제안한 프레임워크인 MWeb-프레임워크도 이러한 특성을 그대로 계승한다.

4.2 제안 프로세스 모델과 MASAM과의 비교

MWeb-프레임워크의 경우 모바일 웹 애플리케이션을 지원하는 프레임워크이고 MASAM의 경우 모바일 애플리케이션을 지원하는 개발방법론이므로 MASAM의 경우가 좀 더 넓은 도메인을 지원할 수 있다. 모바일 애플리케이션을 지원하는 개발방법론이 아직까지 많지 않으므로 이 두 모델의 장단점을 평가한다.

- 대상 : 두 모델 모두 모바일 디바이스를 타겟으로 하나 MWeb-프레임워크의 경우는 웹기반환경에, MASAM의 경우는 모바일 플랫폼들에 최적화 되어있는 개발방법론이다. 그러므로 범용적인 면에서는 MASAM이 유리하나 기존 웹 환경에서 모바일 웹으로의 확장을 고려하는 경우에는 제안 모델이 유리하다.
- 표준의 준수 : 기본적으로 두 모델 모두 표준에 입각한 모델이라고 볼 수 있다. 하지만 모바일 디바이스의 경우 다양한 플랫폼이 사실상의 표준으로 나타나고 있어서 다양한 디바이스에 대한 커버리지의 고려는 중복비용을 만들게 된다. 이러한 측면에서 웹을 플랫폼으로 하는 경우 디바이스 성능의 향상에 따라 점점 가치있는 플랫폼으로 거듭날 것이므로 Mweb-프레임워크의 표준 준수도가 좀 더 높다고 할 수 있다.
- 모델의 구체성 : MASAM의 경우 구체적이고 바로 적용이 가능한 장점이 있다. 하지만 경량화 모델이라는 측면에서 보았을 때 개발하는 모델에 따라 테일러링을 통해 간략화하여야 한다. 제안 프레임워크의 경우는 기

본적인 표준준수와 큰 틀을 제공하기 때문에 상세성은 낮다고 할 수 있으나 반복적이고 경험적인 애자일 개발방법론의 특성상 짧은 반복 안에서 효율적인 모델로 테일러링이 가능할 것이다.

- 문서화 : MASAM의 경우 애자일 개발방법론임에도 불구하고 비교적 많은 문서화를 요구하고 있다. 기존방법론과의 절충안으로 여겨지나 이 부분은 조직별 도입시 꼭 필요한 문서만 남기고 나머지는 줄이는 방향으로 테일러링 하여야 할 것이다.

5. 결론 및 향후 연구

본 논문은 모바일 웹 애플리케이션의 효율적인 개발을 가능하도록 지원하는 MWeb-프레임워크MWeb-프레임워크는 다양한 디바이스의 성능을 가진 모바일 디바이스를 지원하기 위한 모바일 웹 애플리케이션 개발 프레임워크로 사용자의 요구사항을 신속하게 반영하면서도 개발속도와 품질을 모두 고려할 수 있는 장점을 가진 프레임워크이다. 제안한 MWeb-프레임워크를 적용 사례연구를 통해 모델의 우수성을 검증하였다.

향후 연구로는 본 논문에서 비교한 MASAM과의 구체적인 비교를 통하여 본 프레임워크의 커버리지를 확장시켜 도메인별로 특화된 프레임워크로 발전시키고 구체화시키도록 하고자 한다.

참고문헌

- [1] 애니빌, “모바일 웹 산업 동향 및 개발 환경” 애니빌, 2008, <http://www.anybil.com/vx>
- [2] Ajit Jaokar and Tony Fish, “Mobile Web 2.0” Futuretext, 2006
- [3] 박상우, “Web 2.0환경과 모바일 서비스의 가능성” 모바일 웹 2008 컨퍼런스, 2007
- [4] Wikipedia, “Web application framework” Wikipedia, 2008
http://en.wikipedia.org/wiki/Web_application_framework/
- [5] David Heinemeier Hansson, “Ruby on Rails” Ruby on Rails, 2008, <http://www.rubyonrails.org/>
- [6] Dave Thomas, David Heinemeier Hansson, “Agile Web Development with Rails 2/E” Pragmatic Programmers, 2006
- [7] mopad, “MASAM”, <http://www.mopad.org>, 2009
- [8] 강석천, 강규영, 김창준, “변화를 꿈꾸는 개발방법론 애자일(Agile)” 월간 마이크로소프트웨어 2007년 3월호, 2007
- [9] Ron Jeffries, “What is Extreme Programming?. XProgramming.com : an agile software developmentresource” 2001 <http://www.xprogramming.com/xpmag/whatisxp.htm>