

차세대 모바일 웹 애플리케이션 표준화 동향

Standardizations for Future Mobile Web Application

IT 융합 정책 및 표준화 동향 특집

전종홍 (J.H. Jeon) 서비스융합표준연구팀 선임연구원
이승윤 (S.Y. Lee) 서비스융합표준연구팀 팀장

목 차

-
- I . 서론
 - II . 모바일 2.0 기술 동향
 - III . 모바일 웹 애플리케이션 표준화
 - IV . 결론

최근 스마트폰의 도입/확산과 함께 본격화되고 있는 모바일 인터넷의 변화는 하드웨어, 운영체제, 애플리케이션, 서비스까지 모든 면에서의 변화로 나타나고 있다. 아이폰과 안드로이드의 등장은 모바일 2.0 시대로의 전환을 알리는 신호로 새로운 단말과 플랫폼의 무궁무진한 가능성, 그리고 모바일의 미래에 대한 가능성을 보여주고 있다. 과거 단말 위주의 경쟁은 어느덧 플랫폼과 응용 위주의 경쟁으로 바뀌었으며, 애플리케이션에 대한 경쟁력이 핵심 경쟁력이 되었다. HTML5, Device API 등의 진화된 웹 애플리케이션 기술과 결합된 모바일 애플리케이션은 위젯, 오프라인 지원, 하이브리드 애플리케이션과 같은 새로운 애플리케이션 형태와 가능성들을 보여주고 있다. 본 고에서는 이러한 모바일 웹 애플리케이션 기술 동향과 관련 표준화 동향을 고찰함으로써 미래 모바일 애플리케이션 기술의 진화 방향과 그 의미를 살피고자 하였다.

I. 서론

전세계 모바일 인터넷 가입자 수가 최근 급속히 증가하는 가운데, 모바일 데이터 트래픽은 가입자 수 증가세를 훨씬 넘어설 것으로 전망되고 있다. Ovum의 자료에 따르면 전세계 모바일 인터넷 가입자 수는 연평균 50%의 성장률을 보이며, 2008년 1억8천만 명 규모에서 2014년에는 2008년 대비 1024%인 20억 명 수준으로 증가할 것으로 예측하고 있다. 이러한 모바일 인터넷 가입자 수의 증가와 함께 무선 데이터 트래픽도 급격히 증가할 것으로 예측되고 있다. Cisco는 4G의 보급과 함께 전세계 모바일 트래픽이 2013년까지 연평균 131%의 증가율로 증가하여 2008년 대비 66배까지 증가할 것으로 예상하고 있다[1].

이처럼 모바일 산업은 음성통신 위주에서 데이터 통신 위주로 산업의 중심이 옮겨오고 있고, 산업 패러다임의 변화에 따라 음성 중심의 휴대폰도 인터넷 통신을 위한 기기로 탈바꿈하며 스마트폰의 시대를 맞고 있다.

스마트폰의 성장은 다음과 같은 변화들을 촉발시키고 있다. 첫째, 다양한 “스마트”한 기능을 제공하기 위한 지능형 기술을 발전시키고 있다. 둘째, 디바이스의 지능화와 고도화에 따라 플랫폼의 진화와 발전도 필요하게 되면서 고급 기능을 제공하는 개방형 플랫폼의 확산과 관련 경쟁으로 이어지고 있다. 셋째, 플랫폼 기술의 확산과 성장은 하드웨어의 종속성을 낮추고, 애플리케이션 개발 편의성을 높이며, 애플리케이션 사용 편의성을 중심으로 하는 소프트웨어 생태계의 체계모니 이동과 개방형 모바일 소프트웨어 생태계의 확산을 촉진시키고 있다. 넷째, 스마트폰과 플랫폼, 그리고 모바일 애플리케이션의 중요성의 부각은 애플리케이션을 효과적으로 배급할 수 있도록 하는 앱스토어와 같은 모바일 애플리케이션 마켓플레이스에 대한 관심과 중요성을 높이는 계기로 이어지고 있다.

이처럼 모바일 산업의 경쟁력은 단말과 네트워크 중심의 하드웨어 경쟁에서 플랫폼과 애플리케이션

을 중심으로 하는 소프트웨어 경쟁으로 급속히 이동하고 있다. 지금까지 가장 큰 경쟁력이었던 하드웨어를 중심으로 하는 차별화와 부가가치 비중은 점점 줄어들면서 소프트웨어를 기반으로 한 유용성과 부가가치의 비중이 급성장하고 있다[2].

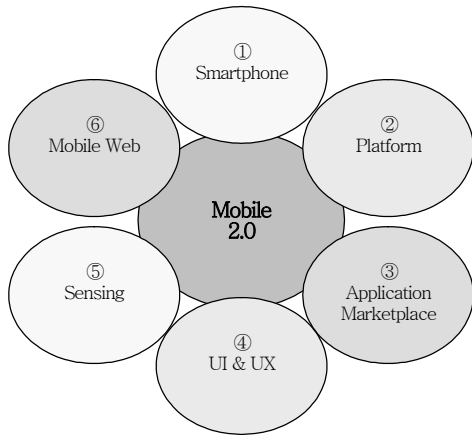
모바일 소프트웨어 경쟁력의 핵심은 플랫폼의 기능성, 인터넷 자원의 효과적 이용, 신속한 개발, 편리한 개발환경을 통한 보다 많은 개발자의 확보, 애플리케이션 배급 경로, 투명한 소프트웨어 생태계의 보장 등과 같은 모바일 애플리케이션 사용과 개발 환경을 어떻게 효과적으로 제공하는가로 모아지고 있다. 모바일 애플리케이션은 보다 빠르게 보다 편하게 모바일 인터넷 상에 있는 다양한 웹서비스와 인터넷 서비스를 결합하여 제공하기 위한 형태로 진화하고 있으며, 전통적인 네이티브 중심의 애플리케이션 형태가 아니라 HTML5, 위젯, 하이브리드 애플리케이션과 같이 웹 기술과 결합된 다양하고 새로운 형태로 나타나고 있다.

본 고에서는 이러한 모바일 웹 애플리케이션 기술 동향과 관련 표준화 동향을 살펴보고 미래 모바일 애플리케이션 기술의 진화 방향과 그 의미를 전망해 보고자 한다.

II. 모바일 2.0 기술 동향

스마트폰과 개방형 플랫폼, 모바일 브로드밴드의 확산을 통해 모바일 환경은 다양한 콘텐츠와 애플리케이션을 빠르고 자유롭게 활용하는 환경으로 발전해가고 있다. 이를 통해 과거 모바일 환경이 읽기 전용(read-only)의 환경이었다면 이제는 자유롭게 읽고 쓰는(read-write) 진정한 의미에서의 모바일 2.0 환경이 되고 있다[3].

모바일 2.0에 대한 주요 기술 변화는 (그림 1)과 같이 6가지 핵심 키워드들로 요약할 수 있으며, 이 중에서도 스마트폰을 중심으로 하는 다섯 가지의 변화가 가장 두드러지고 있다. 모바일 2.0을 위한 기술과 서비스 개발은 모바일 단말 제조사, 이동통신 사업자, 인터넷 기반 서비스 사업자들의 치열한 삼각



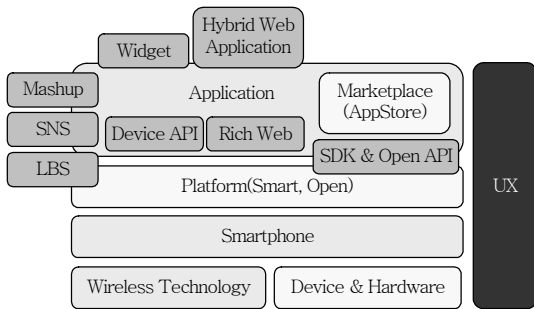
(그림 1) 모바일 2.0의 6가지 핵심 동향[3]

구도 경쟁 속에서 진행되고 있다[4],[5].

1. 스마트폰

2007년을 기점으로 2008년 이후로 모바일 분야에서 가장 많은 변화를 주도하고 있는 키워드는 단연 스마트폰이다. 2007년 2G 아이폰과 2008년 3G 아이폰의 등장 이후 단순히 업무용 PDA 정도로 인식되던 스마트폰에 대해 획기적인 인식 개선이 이루어졌다. 아이폰의 뛰어난 사용자 인터페이스와 지능형 처리를 위한 센서 기술들, 그리고 웹 브라우저와 인터넷 소프트웨어 기술들은 “스마트하지 못했던” 이전 스마트폰에 대한 인식을 바꾸고 새로운 관점을 갖게 하면서 (그림 2)와 같이 많은 변화들을 촉발시키기 시작했다.

스마트폰에 대한 변화는 인식의 변화뿐 아니라, 모바일 단말 시장의 실질적인 변화로도 나타나고 있



(그림 2) 스마트폰과 플랫폼, 애플리케이션의 변화

다. 시장조사업체 Canalsys 자료에 따르면 2008년 3사분기(7~9월)의 전세계 스마트폰 출하대수는 전년 동기 대비 28% 증가한 3,390만 대로 나타났으며, 주니퍼리서치의 전망에 따르면 스마트폰 판매량은 2013년에 3억 대에 이르고(가트너의 예측 자료는 5억5천만 대), 2013년에는 전체 모바일 단말 시장의 23% 이상을 차지할 것으로 전망하고 있다.

스마트폰의 성장은 세 가지 의미를 갖고 있다. 첫째, 모바일 단말 시장이 스마트폰을 중심으로 하는 고급 시장과 일반 휴대폰의 시장으로 세분화된다는 점이다. 둘째, 스마트폰의 확산과 함께 단순 음성통화 위주의 단말에서 벗어나 인터넷 복합 단말기로 발전하면서 데이터 서비스 중심의 모바일 인터넷으로 발전한다는 점이다. 셋째, “스마트”한 단말 기능을 제공하기 위해 센서, 터치스크린과 같은 지능형 UI 기술과 단말 기술, 지능형 처리 기술이 빠르게 발전한다는 점이다[2],[4],[6].

2. 모바일 플랫폼

2008년 애플이 3G 아이폰과 함께 아이폰 애플리케이션 개발용 SDK와 API를 공개하고, 구글이 안드로이드 플랫폼 개발을 시작하면서 모바일 플랫폼(mobile platform) 기술 경쟁은 본격화되기 시작하였다.

구글은 OHA를 중심으로 안드로이드라는 개방형 플랫폼의 개발을 선도하고 있다. OHA의 안드로이드는 “Open Software, Open Device, Open Ecosystem”이라는 목적 아래 운영 시스템, 미들웨어, 사용자 인터페이스, 응용으로 구성되며 자유로운 형태의 개방형 라이선스도 함께 제공되는 것이 특징이다.

애플의 SDK 공개와 개방형 안드로이드의 등장으로 노키아 Symbian도 개방형 플랫폼 경쟁에 뛰어들기 시작하였다. 노키아는 Symbian의 나머지 지분 52%를 인수하고 Symbian Foundation을 설립한 다음, 2~3년 이내에 오픈 플랫폼으로 공개하겠다는 계획을 발표하였다. 2008년 11월에 발표된 심비안 오픈소스화 일정에 따르면, 2009년 상반기까지 Sym-

bian OS의 미들웨어 'S60'과 'UIQ', 'MOAPS'를 통합해 Symbian Foundation 회원사(현재 78개 업체)들에게 무료로 제공하고, 2010년 6월부터 소스코드를 공개할 예정이다.

이처럼 플랫폼이 중요해지고 있는 이유는 수요와 공급 측면에서 살펴볼 수 있다. 수요 측면에서는 디바이스가 지능화되고 고도화되면서 고급 관리 기능이 요구된다는 점이다. 공급 측면에서는 동일한 소프트웨어 플랫폼을 사용하는 환경간의 상호호환성과 교류성이 높아지는 것과 같은 네트워크 효과를 기대할 수 있다는 점이다[7].

플랫폼 기술의 확산과 성장은 하드웨어의 종속성을 낮추고, 애플리케이션 개발 편의성을 높이며, 나아가 애플리케이션을 통해 사용 편의성을 높이고 있다. 결국 차세대 모바일 환경의 관건은 얼마나 진보적이고 진화된 다양한 모바일 애플리케이션을 제공할 수 있는가를 결정짓는 모바일 플랫폼에 있다고 할 수 있으며, 이를 위한 치열한 경쟁이 시작되었다고 할 수 있다[2],[4].

3. 애플리케이션 마켓플레이스

스마트폰과 플랫폼, 그리고 모바일 애플리케이션의 중요성이 부각되면서 이를 효과적으로 배급할 수 있도록 하는 앱스토어와 같은 모바일 애플리케이션 마켓플레이스(application marketplace)의 중요성도 증대되고 있다.

애플 앱스토어의 성공 이후에 수많은 단말사업자와 플랫폼 사업자, 통신사업자들은 앞다투어 마켓플레이스 구축에 나서고 있다. 애플의 AppStore의 뒤를 이어, 구글 안드로이드의 Android Market, 노키아의 OVI Store, 마이크로소프트의 Sky Market, 삼성전자의 Mobile Application Store를 비롯 SKT, KT, LG전자, 모토로라까지 많은 신규 마켓플레이스들이 개설되거나 개설이 추진되고 있다[8].

모바일 애플리케이션 마켓플레이스가 갖는 의미는 세 가지로 요약할 수 있다. 첫째는, 모바일 산업 구조의 변화로, 과거 통신사 중심의 폐쇄적 Walled

Garden의 붕괴를 의미한다. 이제는 이동통신사 망 뿐만이 아니라 WiFi, USB 등과 같은 우회적인 채널을 통해 자유롭게 소프트웨어를 얻고 설치할 수 있는 형태가 되었다. 둘째는, 모바일의 핵심 생태계가 “플랫폼과 소프트웨어”로 바뀐다는 점이다. 사용자는 이동통신사에 의해 가이드된 소프트웨어들이 아니라, 다양한 개발자가 자유롭게 개발한 소프트웨어를 자유로이 구입하고 선택하여 사용하면서 소프트웨어 선택권이 보장되는 진정한 의미에서의 모바일 소프트웨어 생태계가 만들어진다는 점이다. 셋째는 “분절된 경쟁”에서 “연계된 경쟁”으로 경쟁 구조가 변화되었다는 점이다. 과거 서비스와 단말로 이원화되었던 모바일 시장 구조가 단말, 플랫폼, 콘텐츠가 연계되는 통합경쟁으로 바뀌고 있다는 점이다[9].

이처럼 애플리케이션 마켓플레이스의 등장은 개발자에게는 편리한 개발 수익 회수 구조 제공을 통한 개발 동기 부여, 사용자에게는 편리한 모바일 애플리케이션의 설치 편의성을 통한 사용 동기 부여를 하며 모바일 애플리케이션 생태계 활성화의 기반 역할을 하고 있다. 앞으로도 당분간은 유사한 형태의 다양한 모바일 마켓플레이스들이 등장하고 경쟁하면서, 마켓플레이스를 중심으로 한 모바일 애플리케이션 생태계 경쟁은 더욱 치열해질 것으로 보인다.

4. UX

최근 직관적이고 혁신적인 인터페이스를 채택한 제품이 시장에서 성공하고 실감형 인터페이스가 각광을 받는 것처럼, 모바일 분야에서도 진보된 사용자 인터페이스를 위한 장치와 소프트웨어 기술이 주목을 받고 있다.

아이폰 등장 이후로는 터치 인터페이스를 채용한 단말들이 급속히 확산되기 시작하였다. Microsoft의 Windows Mobile 7 버전에서는 터치스크린과 모션 센서 기반의 인터페이스를 기본으로 장착할 예정이며, 삼성전자와 LG전자 등도 햅틱(haptic) 인터페이스와 터치스크린을 활용한 다양한 휴대폰을 출시하고 있다.

멀티터치 스크린 기능 외에도 근접 센서, 조도 센서, 가속도 센서, 지자기 센서와 같이 다양한 센서들을 단말에 내장하고 이를 이용하여 UX를 개선하려는 시도들도 함께 진행되고 있다. 이 밖에도 마이크를 통해 바람을 식별하고, 이를 이용하는 인터페이스 방식을 비롯해 카메라를 다양한 인터페이스의 조합과 재창조가 진행되고 있다.

인터페이스 기술에 대한 중요성이 높아지면서, 관련 특허 출원과 특허 분쟁도 증가 추세에 있다. 애플의 멀티터치 분쟁 사례 등에서 예상할 수 있듯이 인터페이스 기술과 관련된 특허 분쟁의 소지는 점점 커지고 있다. 애플은 멀티터치 이외에도 다양한 인터페이스 관련 특허를 지속적으로 출원하고 있고, PCT를 통해 특허를 다수 출원하고 있다는 사실만으로도, 앞으로의 인터페이스 관련 기술 경쟁이 치열할 것임을 예상할 수 있다[4],[5].

Ⅲ. 모바일 웹 애플리케이션 표준화

모바일 2.0 기술 동향에서 가장 주목할 부분은 애플리케이션 기술 분야다. 최근 모바일 분야의 핵심 경쟁력은 모바일 애플리케이션의 경쟁력으로 나타나고 있고, 이런 경쟁력의 핵심은 플랫폼의 기능성, 인터넷 자원의 효과적 이용, 신속한 개발, 편리한 개발환경과 같은 모바일 애플리케이션 사용과 개발 환경에 대한 이슈들로 모이고 있다.

1. Native App. VS. Web App.

2005년 이후로 웹 2.0의 성장과 함께 콘텐츠 유통과 상거래 방식의 변화, 브라우징 방식의 변화, 웹 애플리케이션 환경의 변화, 서비스 제공 방식의 변화 등과 같은 변화와 함께 다양한 신규 응용과 기술들이 등장하기 시작하였다[10].

이 중에서도 가장 많은 변화를 일으킨 부분은 RSS/Atom 등의 XML 데이터 조각을 이용한 서비스 연동 기술과 AJAX와 같은 비동기식 처리 기술, 브라우저 및 JavaScript 가속화 기술, Open API와 매시업 기

술 분야 등이었고, 이러한 기술들을 종합하는 웹 애플리케이션 기술 분야에서 많은 발전이 있었다.

모바일 분야에서도 전통적인 VM 기반의 네이티브 응용(native application)과 더불어 웹 브라우저 기반의 웹 애플리케이션도 함께 발전해오고 있다.

일반적으로 네이티브 애플리케이션은 빠른 속도를 제공하고 단말의 기능들을 효과적으로 활용할 수 있다는 장점을 갖는 반면, 많은 단말을 지원해야 할 경우 각각 별도 개발을 해야 한다는 문제점과 함께 애플리케이션의 재활용과 업그레이드 등이 용이하지 않다는 단점을 갖고 있다.

반면 웹 애플리케이션의 경우 별도 설치 없이도 계속 업그레이드된 기능을 사용할 수 있고, Open API 등을 통해 손쉽게 매시업 할 수 있도록 기능을 제공하는 등 재활용을 할 수 있다는 장점을 갖는 반면, 오프라인 처리와 단말의 특성 정보를 활용할 수 없고, 브라우저의 성능에 좌우되며 대용량의 처리 등에 한계를 갖는다는 단점을 갖고 있다.

이에 두 애플리케이션들의 장점을 가질 수 있도록 하며, 보다 빠르고 손쉽게 애플리케이션을 개발할 수 있도록 하기 위해 <표 1>과 같이 네이티브와 웹 애플리케이션을 합성하는 하이브리드형 애플리케이션(hybrid application)들이 등장하고 있다. 최

<표 1> Native App, Web App, Hybrid App 비교

	Native App.	Web App.	Hybrid App.
Graphic Performance	상	하	상
AppStore 판매(Monetize)	가능	불가능	가능
Offline Mode	가능	일부 가능	가능
웹서비스 매시업	불가능	가능	가능
Multi-platform 지원	어려움	용이	중간
Storage	로컬	서버, 클라우드	모두
Device Capability 이용	용이	불가능(개선중)	용이
다중 사용자 공동 작업	불가능	가능	가능
소프트웨어 갱신 방법	재설치	사용중 수정	부분 재설치
애플리케이션 재활용성	소스/ Lib 활용만	소스 및 SaaS로	모두
UI 제작 난이도	상	하	중
UI 표현 능력	상	하	중

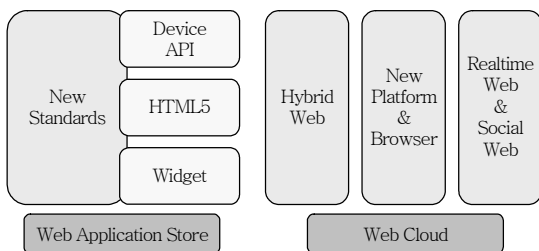
근 애플 아이폰, 구글 안드로이드, 팜 WebOS 등에서 좀더 빠르고 손쉽게 하이브리드형 애플리케이션을 개발할 수 있도록 하는 웹 런타임(web runtime) 엔진들이 개발되어 활용되고 있다.

2. 차세대 웹 애플리케이션 표준

일반적으로 “웹 애플리케이션”이라는 용어는 HTTP를 통해 전달되는 웹 페이지(XHTML 또는 그 변이형과 CSS, ECMAScript로 구성되는)의 집합체들이 웹 브라우저 내에서 애플리케이션 같은 환경을 제공하는 것을 말한다. 즉, 웹 애플리케이션은 여러 페이지를 거치는 대화형 처리 절차를 가지며, 이를 위한 상태 유지와 데이터 유지를 필요로 한다는 점에서 단순한 웹 콘텐츠와는 구분된다.

이 중에서도 “협정의 웹 애플리케이션”은 브라우저 상에서 동작되는 형태만을 고려하지만, “광의의 웹 애플리케이션”은 HTTP, (X)HTML, URI를 필수 동작 요소로 가지며, 브라우저뿐 아니라 독립 애플리케이션으로 동작하는 것도 포괄하는 개념으로 사용된다.

웹 애플리케이션을 효과적으로 개발할 수 있도록 하기 위한 차세대 웹 애플리케이션 기술 동향들은 (그림 3)과 같이 요약할 수 있으며, 이중 W3C를 중심으로 진행되고 있는 핵심적인 다섯 가지 기술 표준화 동향은 다음과 같다.



(그림 3) 차세대 모바일 웹 애플리케이션 기술 동향

가. HTML5와 새로운 마크업

웹 기술이 확산된 가장 큰 배경은 HTML이라는 언어를 이용하여 정보를 표현하고, 이를 다양한 단

말의 브라우저에서 효과적으로 활용할 수 있다는 점에 있었다. 1993년 HTML1.0 규격이 만들어지고 난 후, 1997년 HTML4.0과 1999년의 HTML4.01 규격이 만들어지기까지 웹 기술은 폭발적으로 성장하였다.

그러나 HTML 자체가 갖는 확장의 어려움으로 W3C에서는 1999년부터 좀더 다양한 확장성을 가질 수 있도록 하기 위해서 XML을 기반으로 하는 새로운 XHTML1.0 개발을 추진하였고, 2009년까지 XHTML2.0 개발을 진행하여 왔다.

HTML은 단순함을 가졌으나 확장이 어려웠고, XHTML은 확장성은 좋았으나 지나치게 복잡하다는 단점을 가졌었다. 이러한 이유로 XHTML 표준화는 계속 지연이 되었고, 이에 다양한 기술적인 진화 내역들을 흡수한 새로운 마크업 언어를 필요로 했던 업계 전문가들이 2004년 WHATWG을 구성하고 다양한 웹 애플리케이션에 효과적으로 사용할 수 있는 보다 단순하면서도 다양한 확장성을 갖는 HTML5.0 규격을 만들기 시작하였다.

이에 W3C는 2008년 새로운 HTML 규격을 만들기 위한 HTML WG을 구성하였고, WHATWG의 HTML5 규격을 기초로 한 새로운 표준안을 만들기 시작하였으며, 2009년에는 공식적으로 HTML5 표준화의 시작을 알리고, XHTML2.0 표준화 활동을 중단하기에 이르렀다.

HTML5는 HTML4와 XHTML1 문법과 호환되며 하위호환성을 고려하여 개발중에 있다. 현재 표준 개발중인 HTML5와 HTML4와의 차이에 대해서는 W3C의 기술문서로 정리되어 있으며, 주요 내용은 다음과 같다[11].

- 새로운 요소: section, article, aside, header, footer, nav, dialog, figure 등이 추가
- 새로운 속성: a와 area 요소에 media, ping, hreflang, rel 등을 추가. 이밖에 다양한 새로운 속성들이 추가
- 변경 요소: href, address, b, hr, I, label, menu, small, strong 등 요소의 의미를 변경
- 중단 요소: basefont, big, center, font, s, strike,

- tt, u, frame, frameset, noframes 등이 제외됨
- 중단 속성: HTML4에서 사용되던 속성 중 몇 개의 속성들의 사용을 중단함

이 밖에도 HTML5에서는 웹 애플리케이션 개발에 도움을 줄 수 있는 다양한 API를 제공하게 되고, 새로운 요소들과 함께 보다 손쉽게 애플리케이션을 개발할 수 있도록 하고 있다[12].

- 2차원 그래픽 API 사용을 위한 canvas 요소
- 내장 비디오 및 오디오 재생을 위한 video 및 audio 요소
- 내장 스토리지와 데이터베이스 지원 기능
- 온라인/오프라인 이벤트 기능, 네트워크 API

HTML5 표준안은 아직 초안 상태로 앞으로도 많은 수정과 보완 작업이 필요할 것으로 예상되고 있지만, 현재 스펙을 기준으로 한 구현은 <표 2>와 같이 이미 대부분의 브라우저에서 구현되어 동작되고 있다. 이 밖에도 구글/애플/모질라/마이크로소프트 등을 비롯한 많은 브라우저 개발사들은 브라우저 기능 개선과 함께 자사 규격을 표준에 반영하고 서비스 개발에 반영하기 위한 확장 노력을 병행하고 있다.

2009년 W3C에서는 HTML5 뿐 아니라 DOM3, CSS3, WAI-ARIA, XSLT를 비롯한 Xpath, XQuery 등에 대한 많은 새로운 마크업 관련 표준 개발을 진행하였다. 또한 구글 등을 중심으로 하는 웹 애플리케이션 진영에서의 HTML5 기반의 다양한 시도들

<표 2> 브라우저별 HTML5 지원 현황

	Chrome	Firefox	Safari	Opera
Canvas	√	√	√	√
Video	√	√	√	√
Geolocation	√	√	√ (iPhone)	√
AppCache	√	√	√	√ (mobile)
Database	√	√	√	√ (mobile)
Workers	√	√	√	√ (mobile)

을 하고 있다는 점은 주목할 만하다. 특히 Geolocation API와 다양한 Device API 표준들이 개발되고 있고, HTML5가 웹 애플리케이션에 좀더 초점을 맞추고 있다는 점은 HTML5를 중심으로 한 웹 애플리케이션 기술의 큰 변화를 예상하게 하는 점이다.

나. Device API

웹 애플리케이션이 갖는 가장 큰 단점 중 하나는 네이티브 애플리케이션과 달리 단말의 하드웨어와 관련된 제어를 할 수 없다는 점이라 할 수 있다. 예를 들어, 간단한 애플리케이션을 통해 배터리의 잔량, 주소록의 주소 정보, 단말에 저장된 일정 정보 등을 활용하고자 해도 할 수 없다는 점은 치명적인 약점으로 꼽혀 왔다.

이러한 웹 애플리케이션의 약점은 모바일 환경에서 더욱 치명적이라 할 수 있다. 데스크톱의 웹 애플리케이션과 달리 모바일 단말의 경우 좀더 많은 플랫폼으로부터의 제약을 갖고 있지만, 반면에 좀더 다양하게 디바이스 기능들을 활용할 필요를 갖고 있어 단말 기능 접근에 대한 요구가 훨씬 크다고 할 수 있다.

이에 W3C에서는 2008년 12월 Device API와 관련된 다양한 표준화 이슈들을 발굴하기 위해 관련 워크숍을 개최하였고, 워크숍 논의 결과를 기초로 WG 설립 작업을 진행하여 2009년 6월 DAP WG을 발족하게 되었다.

이에 앞서 OAA의 Mobile Device API, OMTP의 Bondi Activity, JIL 표준화 등으로 이어져 오면서 디바이스 API 표준화에 대한 다양한 논의들이 이루어진 바 있어, 이러한 기존 작업 결과들이 W3C로 취합되는 형태가 되고 있다.

현재 W3C DAP WG은 OMTP의 Bondi 1.0 규격과 Nokia에서 제출한 Device API 규격들을 중심으로 1단계 표준화 활동을 2010년 말까지 마무리할 계획으로 있으며, 8개 이상의 핵심 API 문서와 요구 사항 문서를 개발할 예정으로 있다.

- API 문서: Contact API, System Information

and Events API, Camera API, User Interface API, Tasks API, Messaging API, Gallery API, File System API, Communication Log API, Device Interface

- 요구사항 문서: Device API Requirements, Device API Policy Requirements

DAP WG은 2009년 11월 TPAC 회의에서 제1차 대면회의를 개최하고 WG 활동과 관련한 전반적인 계획을 재조정하고, 본격적인 표준화 작업을 시작하였다. 또한 1단계 작업 범위를 확정하고, 향후 작업 일정을 위한 로드맵을 만들었으며, 주요 API들에 대한 우선 순위를 선별하고 표준화 작업을 진행중에 있다. 우선 작업 대상으로는 현재 5개의 API(Contact API, Calendar API, Filesystem:File Writing API, Capture(audio/video) API, Messaging API)를 선정하여 우선 표준화 초안 작업중에 있다.

- Contact API: 2009년 10월에 1차 초안이 나온 상태로 2010년 9월 완료를 목표
- Calendar API: 2010년 1월 1차 초안 작성을 목표
- File System API: 2009년 12월 1차 초안을 기초로 2010년 9월 완료를 목표
- Capture API: 2009년 12월 1차 초안을 기초로 2010년 9월 완료를 목표
- Messaging API: 2009년 12월 1차 초안을 기초로 2010년 9월 완료를 목표

Device API와 중첩된 영역이지만, Geolocation API와 관련된 표준화는 DAP 구성 이전에 W3C의 Geolocation API WG을 통해 표준화가 마무리 단계에 있어, DAP WG의 활동 범위에서는 빠지게 되었다. 또한 구글이 제안하였던 Notification API는 웹 애플리케이션 WG으로 넘기고, 가로, 세로 전환 및 가속 센싱과 관련되는 API는 Geolocation WG으로 넘겨서 작업하기로 하였다.

Device API와 관련된 표준화는 W3C DAP WG을 중심으로 진행하되, OMTB Bondi와 JIL 등 다양한 조직들에서 적극적이고 빠른 표준화를 진행할 예정으로 있어 앞으로 많은 논의와 빠른 진행이 예상

되고 있다. 또한 모바일 웹 애플리케이션과 관련하여 가장 많은 영향을 미칠 수 있는 표준으로, 2010년 W3C DAP의 1단계 표준화가 완료되고 다양한 모바일 브라우저에서 구현된다면 훨씬 강력한 기능을 제공하는 다양한 모바일 웹 애플리케이션이 등장할 것으로 예상된다.

다. Web Application Standards

W3C는 2006년 Rich Web Client Activity를 시작하며 Web Application WG과 Web API WG을 만들어 표준화 작업을 진행한다. 2008년 Web Application WG으로 통합하여 표준화 작업을 진행해오고 있다[13].

<표 3>에서 보는 바와 같이 현재 약 20여 개 이상의 Web Application 관련 표준안들이 Web Application WG 내에서 검토되고 협의중에 있으며, 여기에는 XHR, Widget, Web IDL, Web Socket API, CORS 등이 포함되어 있다. 이 밖에도 HTML5 규

<표 3> WebApps WG 작업 문서 현황

Name of Spec	Last Publication	Type
Cross-Origin Resource Sharing (CORS)	2009-03-17	WD
DataCache API	2009-10-29	FPWD
DOM Level 3 Events	2009-09-03	WD
Element Traversal	2008-12-22	REC
File API	2009-11-17	FPWD
Indexed Database API	2009-09-29	FPWD
Progress Events	2008-05-21	WD
Selectors API	2008-11-14	LC #2
Server-Sent Events	2009-10-29	WD
Web SQL Database	2009-10-29	WD
Web IDL	2008-12-19	WD
Web Sockets API	2009-10-29	WD
Web Storage	2009-10-29	WD
Web Workers	2009-10-29	WD
XBL2 Spec	2007-03-16	CR
XBL2 Primer	2007-07-18	WD
XmlHttpRequest	2009-11-19	LCWD
XmlHttpRequest Level 2	2009-08-20	WD

격과 연관된 Web Storage, Web Workers, Data-Cache API, DOM Level 3 Events 등도 작업 대상으로 포함되어 있다.

이러한 Web Application WG의 주요 표준화 활동은 다음과 같이 6가지 내용으로 요약할 수 있다. (추가적으로 Widget에 대한 규격 집합도 있으나, 향후 Widget 표준들은 Widget WG으로 독립하여 작업할 예정에 있으므로, 위젯 표준 현황에 대해서는 별도로 기술하였다.)

1) XHR

XHR은 AJAX와 같은 비동기식 웹 애플리케이션 개발 기법의 핵심 요소, 서버와 클라이언트 사이의 데이터 전송을 위한 기능을 정의한다. XHR 1.0 버전은 최종 초안으로 조만간 권고안 제정 단계로 진입할 예정으로 있으며, 이러한 XHR 1.0을 확장하는 XHR 2.0 표준에 대한 초안 작업을 진행하고 있다.

2) Web IDL

Web IDL은 브라우저에서 구현되어 웹 상에서 인터페이스를 설명하기 위한 용도로 사용될 수 있는 IDL(인터페이스 정의 언어)을 정의한다. Web IDL은 인터페이스의 정의와 더불어, 인터페이스와 ECMAScript, 그리고 자바 바인딩에 대한 명료한 적합성 요구사항을 제공하는 데 이용된다.

3) Web Socket

웹 소켓 API 규격에서는 원격 서버와의 양방향 통신을 가능하도록 하는 웹 소켓을 이용하는 웹 페이지를 가능하도록 API를 정의한다. 웹 소켓에 대한 규격은 IETF에서 표준화 작업을 진행중에 있다.

4) Web Storage

웹 저장소 규격에서는 웹 클라이언트 내에 구조화된 키-값 쌍 데이터의 영구적 데이터 저장을 위한 API를 정의한다.

5) Web Workers

웹 워커 규격에서는 웹 애플리케이션 작성자가 메인 페이지 내에서 병렬적으로 스크립트 백그라운드 작업을 생성하여 실행할 수 있도록 하는 API를 정의한다. 이를 통해 장시간 실행되는 스크립트를 인터럽트 없이 수행 가능하도록 할 수도 있다.

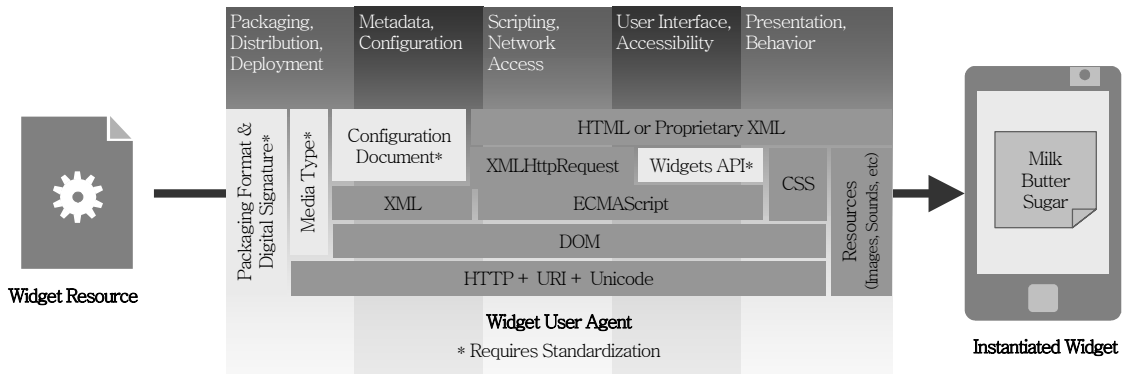
6) DataCache API

데이터 캐시 API 규격에서는 정적/동적 응답을 이용하는 HTTP 리소스 요청에 대한 오프라인 제공을 위한 API를 정의한다. 연관된 규격으로는 HTML5 내의 AppCache 규격이 있다.

Web Application WG의 작업 표준 현황에서 알 수 있듯이, 다수의 웹 애플리케이션 관련 규격들이 개발되고 있다. 특히 주로 HTML5와 관련하여 스토리지 처리, 백그라운드 처리, 소켓 처리, 비동기 데이터 처리 등과 같은 새로운 규격들이 개발중에 있다는 사실에 비추어 앞으로 차세대 웹 애플리케이션의 기능과 형태에 많은 변화가 있을 것으로 보이며, 이에 대한 대비가 필요할 것으로 예상된다.

라. Widget

‘위젯(Widget)’이란 용어에 대해 아직 다양한 정의와 인식의 차이들이 있기는 하지만, “사용자 기기 또는 모바일 단말에 다운로드 하거나 설치할 수 있으며 간편히 쓸 수 있도록 만든 작은 창(window) 형태의 응용” 개념으로 정의되고 있다. 위젯은 그 실행 유형과 구동 플랫폼, 구동 방식에 따라 다양한 유형으로 구분된다. 보통 웹 위젯은 웹 기술을 사용하여 구동되는 위젯 형태를 의미하며, 모바일 위젯은 모바일 단말에서 구동되는 위젯을 부른다. 물론 위젯이란 용어도 gadget, badge, module, webjit, capsule, snippet, mini, flake 등과 같은 다양한 이름들로 불리기도 하지만, 최근에 와서는 대체적으로 위젯이란 이름으로 통일되고 있는 추세에 있으며, 그 유형도 웹 위젯으로 대표되고 있는 추세라 할 수 있다.



(그림 4) 웹 위젯 기술 구성도(W3C)

위젯 표준화에 대한 필요성은 2006년부터 제기 되기 시작하였다. 위젯에 대한 관심이 높아지고 다양한 위젯 플랫폼들이 개발되면서, 위젯 플랫폼 간의 위젯 호환성을 높이고, 기 개발된 위젯 애플리케이션들을 공유하여 사용할 있도록 하자는 필요성이 제기되었기 때문이다. 예를 들어 A사가 개발한 위젯과 B사가 개발한 위젯 정의에 사용되는 마크업 언어가 틀리고, 구동 방법이 달랐기에 상호 호환되는 동작을 할 수 없었기 때문이다.

이에 2007년부터 W3C의 웹 애플리케이션 WG을 중심으로 표준화 작업이 시작되었다. 초기에는 단순히 2개의 표준안(위젯 요구사항과 위젯 언어) 작성 계획으로 출발하였지만, (그림 4)의 위젯 아키텍처 참조 모델을 기초로 9개 문서로 나누어 현재 작업중에 있다.

- Widget 1.0 Landscape: 위젯 관련 규격 및 제품 등을 총괄 정리(WG Note로 2010년 완성 예정)
- Widget 1.0 Requirements: 위젯 표준화에 관한 요구사항들을 총괄 정리(2010년 초안 완료 예정)
- WPC: 위젯 배포를 위한 패키징 및 환경설정 규격(2010년 중 권고안 예정)
- Widget 1.0: Digital Signatures: 안전한 위젯 리소스 배포를 위한 전자서명 규격(2010년 권고안 예정)
- Widget 1.0: Widget Interface: 위젯 데이터 메타데이터 액세스를 위한 API 규격(2010년 권고안 예정)

- Widgets 1.0: Updates: 위젯 버전 관리를 위한 규격(2010년 초안 완료 예정)
- WARP: 위젯으로부터의 네트워크 액세스 제어를 위한 보안 모델 규격(2010년 초안완료 예정)
- Widgets 1.0: Widget URIs: 위젯 URI 스킴 규격(2010년 최종초안 완료 예정)
- Widgets 1.0: View Models Media Feature: 뷰 모델과 표현 모드에 대한 규격(2010년 최종 초안 완료 예정)

이 밖에도 다국어 환경을 위한 Widget I18N, 위젯 상호호환성 검증을 위한 테스트 스위트 등과 같은 새로운 규격들에 대한 논의들도 진행되고 있다. 앞으로 또한 HTML5를 비롯하여 오프라인 처리를 위한 웹 스토리지, 웹 애플리케이션을 위한 Web Worker, Web Socket, Device API 등의 규격이 발전함에 따라 위젯의 응용 범위와 용도도 지속적으로 확장되고 개선될 것으로 보인다. 이를 통해 데스크톱, 모바일, 정보가전의 위젯 환경을 아우르는 통합 위젯 표준화에 대한 요구 또한 증가할 것으로 예상된다.

마. Mobile Web Application Best Practices

지난 1999년 웹에서 아이디어를 얻어 만들었던 WAP 환경이 발전하지 못했던 가장 근본적인 이유는 폐쇄적 서비스와 비 표준화된 환경에 있었던 것처럼, 모바일 웹 활성화를 위해서는 모바일 웹 표준화가 필수적이다[10].

2005년부터 시작한 W3C MWI Activity에서는 유무선 환경에 상관없이 일관된 웹 사용 환경을 만들기 위한 모바일OK 표준화를 추진하였고, 이를 통해 다음과 같은 문제점들을 해소하면서 상호호환성 있는 모바일 웹 환경을 만들기 위해서 노력해오고 있다[3],[14].

- 사용자들은 모바일 단말을 이용하여 손쉽게 다양한 웹 콘텐츠를 볼 수 없다.
- 개발자들은 각각 서로 다른 이동통신사와 단말에 맞도록 웹 콘텐츠를 수작업으로 만들고 유지 보수하는 등 많은 비용을 들여야 한다.
- 이동통신사들은 다수의 고유 규격들을 사용함으로써 웹 콘텐츠와 애플리케이션 간의 호환성이 없고, 중복 개발해야 하는 문제를 갖는다.
- 콘텐츠 제공자는 단말의 성능과 기능에 대한 특성 정보를 공유할 수 없어 단말 적응형 응용을 만들고 제공할 수 없다.
- 모바일 브라우저가 필요한 웹 표준을 구현하지 않고 있거나, 상호 호환되지 않는 방식으로 구현하고 있어, 호환성이 없다.

W3C의 MWBP 워킹그룹은 2005년부터 2007년까지의 1단계 작업을 통해 모바일 웹을 위한 기술적 모범 사례 표준화를 진행하였고, 2008년부터는 모바일 웹 애플리케이션 호환성 확보를 위한 모범 사례 표준과 관련 기술 표준을 만들기 시작하여, 현재 두 개의 표준안을 개발하고 있다[15].

- 1) CT Guideline 1.0: 웹 콘텐츠 변환 시에 동작하는 콘텐츠 변환 서버와 프록시의 동작 방식과 그 결과에 대한 표준
- 2) MWABP: 모바일 웹 애플리케이션의 개발 및 활용에 관한 모범 사례 표준

MWABP 표준에서는 모바일 단말 상에서 구동되는 “모바일 웹 애플리케이션”의 개발과 배포에 관한 모범 사례들을 정리하는 것을 목적으로 하고 있다. 기존의 MWBP는 정적인 문서와 콘텐츠를 중심으로 하는 모범 사례를 정리하였다면, MWABP에서는 동적인 모바일 웹 애플리케이션에 초점을 맞춘 모범

사례들을 정리하고 있다.

MWABP에서는 기존 표준인 MWBP 1.0과 차별성을 갖는 새로운 모범 사례를 중심으로, 아래와 같이 애플리케이션 데이터, 보안 관련, 사용자 인식성, 사용자 경험, 단말 기능 활용 등과 관련된 총 39개 정도의 모범 사례들을 정리하고 있다.

- 애플리케이션 데이터: 대부분의 애플리케이션은 다양한 형태의 데이터를 저장해야 하는데, 이러한 웹 애플리케이션의 데이터와 관련된 적절한 기술 및 기법에 관한 모범 사례
- 보안 및 프라이버시: 신뢰성 있는 정보 사용과 프라이버시 보호를 위한 모범 사례
- 사용자 인식 및 제어: 애플리케이션의 동작 방식에 관해 사용자가 편리하고 효과적으로 인식하고 제어할 수 있도록 하기 위한 모범 사례
- 사용자 경험: 복잡한 상호작용을 단순화 시키고 최적의 사용자 경험(UX)을 제공하기 위해 고려해야 할 모범사례들

2010년 중에 최종 표준안으로 확정될 예정인 모바일 웹 애플리케이션 모범사례는 다양한 모바일 웹 애플리케이션과 관련된 모범사례를 정리하였는데 그 의의가 있으며, 앞으로 모바일 웹 애플리케이션의 확산과 함께 모바일 웹 애플리케이션의 개발/활용과 관련하여 중요한 기초 가이드로 활용될 것으로 예상된다.

바. 국내 표준화 동향

국내의 모바일 웹 표준화는 TTA PG605(웹 프로젝트 그룹)와 산하의 WG6051(모바일 웹 실무반)에서 진행하고 있고, 모바일 웹 2.0 포럼에서 모바일 OK 표준화를 포함한 다양한 모바일 웹 표준화를 추진하고 있다.

모바일OK 표준화의 경우, 2008년까지 진행되었던 표준화 및 시범사업 방향이 모바일 웹 콘텐츠 호환성에 초점을 맞추었다면, 2009년부터는 모바일 웹 애플리케이션 중심 표준화 및 이를 기반으로 하는 2단계(2010년 3단계) 모바일OK 표준화를 추진

중에 있다. 모바일 웹 2.0 포럼에서는 모바일OK 콘텐츠 표준 이외에도, 모바일OK 애플리케이션 표준화, Device API, 위젯 표준화, 한국형 MWABP 표준 개발 등을 진행하고 있다[16].

이 밖에도 모바일 웹 2.0 포럼은 차세대 모바일 웹 애플리케이션과 모바일 2.0 분야의 국내/국제 표준화를 선도하기 위해, 주요 표준화 이슈 발굴과 협력을 위한 MWAC를 개최하고 있으며, W3C MWI 멤버 등으로 활동하며 국내외의 다양한 기업과 포럼 등과의 표준화 협력을 강화하고 있다[17].

IV. 결론

지금까지 모바일 2.0의 동향과 함께 빠르게 변화하고 있는 모바일 소프트웨어 생태계와 차세대 모바일 웹 애플리케이션 표준화 동향에 대해 살펴보았다.

이를 통해 차세대 모바일 웹 애플리케이션 기술과 관련하여 주요하게 두 가지 특징이 나타나고 있다는 점을 알 수 있었다.

첫째는 유무선의 경계가 사라지고 있으며, 모바일 애플리케이션에서 웹 기술과 웹 서비스의 매시업과 협력이 두드러지고 있다는 점이다. 구글, 야후, 이베이, 마이크로소프트를 비롯한 다수의 인터넷 기업들이 기존 웹 서비스를 개방형 웹 플랫폼의 형태로 발전시키고, API를 개방하면서 협력 발전 모델을 만들어가고 있다. 예를 들어 애플 아이폰의 주요 기능들은 구글 웹 서비스를 이용하고 있고, 이 밖에도 Facebook, Twitter, Amazon에서 제공하는 다양한 개방형 API를 이용한 많은 애플리케이션들이 빠르게 등장하고 있다는 점이었다. 또한 유선과 무선의 차이들은 없어지고, 단지 고유한 매체적 특성만 일부 남는 형태가 되고 있다는 점이다.

둘째는 하이브리드형 매시업 애플리케이션과 개발 방법이 급속하게 확산되고 있다는 점이다. 과거와 같이 Native Application과 Web Application이 따로 떨어지는 것이 아니라, 서로가 서로의 장점을 취하며 하이브리드 되는 형태로 진화하고 있었다.

특히 아이폰, 안드로이드, Palm의 WebOS와 같은 플랫폼에서는 보다 손쉽게 하이브리드 애플리케이션을 개발할 수 있도록 하는 다양한 도구들이 등장하고 있으며, 많은 웹 애플리케이션들은 하이브리드 애플리케이션으로 재포장되어 앱스토어를 통해 판매되고 있다.

이처럼 최근의 모바일 애플리케이션 환경은 플랫폼, 클라우드, 웹 서비스가 더욱더 밀접하게 연계되는 형태로 진화하고 있다. 다양한 개발도구는 하이브리드 애플리케이션을 손쉽게 개발할 수 있도록 하고, 개방형 API와 웹 클라우드는 다양한 기능과 서비스를 손쉽게 연계할 수 있도록 하고 있으며, 모바일 브로드밴드와 고성능 단말은 새롭고 뛰어난 모바일 애플리케이션 UX를 제공할 수 있도록 하고 있다.

향후에는 지금까지 살펴보았던 차세대 모바일 웹 애플리케이션 관련 표준과 기술이 안정화되고 정착되는 과정에서 훨씬 다양한 형태의 모바일 애플리케이션들이 등장하도록 할 것으로 예상된다. HTML5와 스케줄러, 그리고 브라우저 내의 데이터베이스를 이용하여 복잡한 캐시와 오프라인 처리를 손쉽게 하면서 다양한 단말의 기능을 제어하고, 사용자 단말의 대기화면과 다양한 서버 사이를 자유롭게 넘나드는 웹 표준 기반의 차세대 웹 애플리케이션과 하이브리드 웹 애플리케이션이 등장할 것으로 보인다. 이와 함께 일반 사용자들도 손쉽게 화면 드래그만으로도 원하는 맞춤형 모바일 애플리케이션을 제공할 수 있는 것도 가능할 것으로 보인다.

특히 스마트폰이 확산되고 플랫폼 경쟁이 심화되면서 보다 많은 종류의 모바일 단말과 플랫폼이 등장하게 될 것으로 예상되며, 이러한 다양한 단말 환경에서 효과적으로 애플리케이션을 개발하고 제공하기 위해서는 장치독립적인 모바일 웹과 하이브리드형 애플리케이션 기술에 대한 관심이 증가할 수밖에 없을 것으로 예상된다.

이런 차세대 모바일 애플리케이션 기술 전망과 함께 국내상황을 돌이켜보면 상당히 부족함이 많아 보인다. PC 위주, 그중에서도 MS의 윈도 IE에서만 구동 가능한 ActiveX 기반의 국내 웹 환경은 모바

일 환경뿐 아니라 차세대 단말 환경에 대해서도 자연스러운 적용이 불가능하다. 이처럼 인터넷 강국의 콘텐츠와 애플리케이션 중에서 차세대 단말 환경에서 사용할 수 있는 것이 채 5%도 못된다면 이견 정말로 심각한 문제라 할 수 있다.

우리는 다시 한번 한국의 인터넷 산업과 모바일 산업이 표준과 국제적 기준에 맞추어 함께 경쟁하고 어울릴 수 있는 구조인지 진지하게 살피고, 좀더 미래지향적인 구조와 기술로 변경해야 할 때라 생각한다. 그런 면에서 지금부터의 스마트폰 환경과 모바일 웹 애플리케이션을 중심으로 하는 변화의 기회는 국내 모바일 산업과 인터넷 산업이 일신우일신(日新又日新) 할 수 있는 소중한 기회라 생각한다.

● 용 어 해 설 ●

모바일OK: 모바일 웹 콘텐츠 호환성을 위한 콘텐츠 호환성 인증 마크. W3C에서는 모바일 웹의 호환성 기준을 개발하여 호환성을 가진 콘텐츠, 사이트, 브라우저 등에 호환성 인증마크를 사용하도록 함으로써 모바일 웹에서의 비표준화 문제를 해결하고 활용성을 높이고자 하고 있다. 국내에서는 모바일 웹 2.0 포럼을 통해 모바일OK 인증 체계와 기준을 만들고 있다.

위젯(Widget): 사용자 기기 또는 모바일 단말에 다운로드 하거나 설치할 수 있으며 간편히 쓸 수 있도록 만든 작은 창(window) 형태의 응용을 말한다. 웹 위젯은 웹 페이지에서 구동되는 위젯 형태를 의미하며, 모바일 위젯은 모바일 단말에서 구동되는 위젯 형태를 의미한다.

약어 정리

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CORS	Cross-Origin Resource Sharing
CSS	Cascading Style Sheets
CT	Content Transformation
DAP	Device APIs and Policy
DDR	Device Description Repository
DOM	Document Object Model
HTML	Hypertext Markup Language
I18N	Internationalization
IDL	Interface Definition Language

IRI	Internationalized Resource Identifier
MAR	Mobile Augmented Reality
MWABP	Mobile Web Application Best Practices
MWAC	Mobile Web Apps Camp
MWBP	Mobile Web Best Practice
MWI	Mobile Web Initiative
OAA	Open AJAX Alliance
OHA	Open Handset Alliance
OMTP	Open Mobile Terminal Platform
RIA	Rich Internet Application
RSS	Really Simple Syndication
SNS	Social Networking Service
URI	Uniform Resource Identifier
UX	User eXperience
VM	Virtual Machine
W3C	World Wide Web Consortium
WAI-RIA	Web Accessibility Initiative - Accessible Rich Internet Applications
WARP	Widget Access Request Policy
WebIDL	Web Interface Definition Language
WHATWG	Web Hypertext Application Technology Working Group
WPC	Widget Packaging and Configuration
XHR	XML HTTP Request
XHTML	Extensible Hypertext Markup Language

참 고 문 헌

- [1] 전수연, 김태현, “해외 모바일 브로드밴드 동향,” KISDI 이슈 리포트, 제21권 19호, 2009년 10월.
- [2] 유지은, “스마트폰의 Key Enabler: 스마트폰,” 소프트웨어 Insight, 2009년 4월.
- [3] W3C Mobile Web Initiative, <http://www.w3.org/Mobile/>
- [4] 전종홍, “모바일 2.0 서비스 동향 및 전망,” 한국통신학회지, Vol.26, No.4, 2009, pp.3-11.
- [5] 전종홍, 이승윤, “모바일 웹 2.0 기술 전망,” SK Telecom Review, 제17권 4호, 2007.
- [6] 권기덕, “스마트폰이 IT시장에 미치는 영향,” 소프트웨어 Insight, 2009년 4월.
- [7] 박동욱, “소프트웨어 플랫폼 경쟁의 전면화와 대응 방안,” LG경제연구원, LG Business Insight, 2008년 8월.
- [8] 권지인, “국내외 모바일 애플리케이션 마켓 현황

- 과 시사점,” KISDI 이슈 리포트, 제21권 13호, 2009년 7월.
- [9] 정제호, “스마트폰 마켓플레이스, 도전과 기회,” 소프트웨어 Insight, 2009년 10월.
- [10] 전종홍, 이승윤, “웹 2.0 기술 현황 및 전망,” 전자통신동향분석, 제21권 제5호, 2006년 10월, pp. 141-153.
- [11] W3C, Working Draft, “HTML 5 Differences from HTML 4,” <http://www.w3.org/TR/2009/WD-html5-diff-20090423/>
- [12] W3C, Working Draft, “HTML 5,” <http://www.w3.org/TR/html5/>
- [13] W3C Rich Web Clients Activity, <http://www.w3.org/2006/rwc>
- [14] 전종홍, 이승윤, “모바일 웹 2.0과 모바일OK 표준화 동향,” 전자통신동향분석, 제22권 제6호, 2007년 12월, pp.84-95.
- [15] W3C Mobile Web Best Practices WG, <http://www.w3.org/2005/MWI/BPWG/>
- [16] Mobile Web 2.0 Forum, <http://www.mw2.or.kr>
- [17] Mobile Web Apps Camp, <http://mobilewebapps-camp.com>