모바일 통합 웹 게시판 게이트웨이 개발

(Development of Mobile Integrated Web Board Gateway)

이신호, 권동우, 김현우, 주홍택

계명대학교 컴퓨터공학과

{ leeshinho, dwkwon, hwkim84, juht }@kmu.ac.kr

요 약

커뮤니티 사이트들은 다양한 정보를 제공하고 공유하기 위하여 게시판 형식의 사이트를 구축하고 있다. 오픈 소스 웹 게시판 프로젝트들이 활성화되어 있고, 이를 이용하여 간편하게 커뮤니티 게시판을 구성할 수 있다. 최근 스마트폰 사용이 증가함에 따라, 필요한 정보를 얻기 위하여, 여러 웹 게시판을 방문하게 되는 불편함이 존재한다. 우리는 이러한 서로 다른 여러 커뮤니티 사이트들의 게시판 정보들을 하나로 모아, 사용자가 설정한 개인화를 통하여 사용자가 원하는 정보를 한 번에 손쉽게 얻을 수 있는 방법인 모바일 통합 웹 게시판 게이트웨이 개발 방법을 제안한다.

Keywords: Application Gateway, Web Board, Integrated Gateway Service, MQTT

1. 서론

현재 스마트폰 사용의 증가로 인하여, 다양한 정보들을 언제 어디서든 편하게 습득할 수 있게 되었다. 인터넷이 보편화 되고 웹이 활성화되면서, 온라인 커뮤니티 사이트 또한 빠르게 형성되고 발전하였다. 이러한 커뮤니티 사이트들은 다양한 콘텐츠를 주제로 하며, 사용자에게 다양한 정보를 제공할 수 있도록 게시판 형식으로 사이트를 구성하고 있다. 현재 이를 위한 많은 오픈 소스 웹 게시판 프로젝트들이 활성화되어 있으며, 이를 이용해 많은 사이트가 커뮤니티를 위해 활용되고 있고, 이런 프로젝트 중 대표적으로는 XpressEngine, 그누보드, Technote 등이 있다.

인터넷의 급속한 발전으로 이러한 많은 커뮤니티 사이트들이 형성되면서 사용자들은 정보를 필요로 하는 사이트들을 검색을 통하여 활동하게 되었고, 이에 따라, 개인당 가입된 커뮤니티 활동 사이트 또한 많아지게 되었다. 2005 년 온오프코리아가 인터넷 개인정보보호 사이트인 이지스를 통해 국내 주요 1,000 개 웹사이트를 이용하는 네티즌의 자료를 분석한 결과 1 인당 평균 27 개의 웹사이트에 회원으로 가입되어 있으며, 사용하는 ID 종류도 평균 7.5 개인 것으로 조사되었다[1]. 또한, 한국인터넷진흥원(KISA) 인터넷통계정보시스템(ISIS)에서 2012 년 인터넷 이용실태조사에 따르면 주된 인터넷 이용 용도로 자료 및 정보 획득이 92.3%이며, 커뮤니티를 목적으로 하는 용도는 44.1%로 집계되었고, 가구당 정보통신기기 보유현황에서 스마트폰 보유율이 65%를 차지하고 있다[2].

이처럼 현재 스마트폰이 보편화됨에 따라, 다양한 콘텐츠를 주제로 커뮤니티 사이트들이 형성되면서 여러 사용자 간의 상호 의견과 정보를 교류할 수 있는 사이트들이 범람하게 되었다. 하지만 특정한 콘텐츠를 주제로 하여 형성된 커뮤니티 사이트에서 주로 활동을 하는 사용자들은 활동하는 사이트가 많을수록, 필요한 정보를 얻기 위해 자신이 활동하는 사이트들을 일일이 방문하여 확인해야 하는 불편함과 접속과정에서 발생하는 불필요한 트래픽 유발 및 시간적 소요들이 존재하게 된다. 또한, 모바일 환경을 제공하지 않은 사이트들이 아직 존재하고 있고, 모바일 환경을 제공하는 커뮤니티 사이트일지라도 일일이 방문하여 확인해야 한다는 점에 있어선 동일하게 불편함을 주고 있다. 특히, 입찰 정보나 경매, 증권 같은 민감한 정보를 제공하는 사이트에서 이러한 정보들을 빠르게 얻기 위해 사용자는 항상 게시판을 모니터링 해야 하는 단점을 지니게 되는데, 그와 반면에 Really Simple Syndication or Rich Site Summary(RSS)를 이용하여 사용자가 필요한 정보를 얻을 수 있지만, RSS 를 지원하지 않은 게시판에서는 그 한계점이 있다.

우리는 이러한 서로 다른 여러 커뮤니티 사이트들의 정보를 한곳으로 모아, 사용자가 설정한 개인화를 통하여, 원하는 정보를 빠르고 손쉽게 얻을 수 있는 방법으로 모바일 통합 웹 게시판 게이트웨이에 대한

본 연구는 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과임. (2012H1B8A2025942)

설계를 제안하고 구현 결과를 제시한다. 우리가 제안하는 모바일 통합 웹 게시판 게이트웨이는 모바일에 특화된 환경 제공을 하기 위해, 푸시 알림 서비스에 대한 개념을 도입한다. 모바일 환경에서 사용자는 게이트웨이에서 개인화 설정을 통하여 원하는 게시판의 정보를 통합하고, 실시간으로 푸시 알림 서비스로 정보를 제공한다면, 사용자는 여러 사이트에 방문하거나 모니터링 해야 하는 불편함과 불필요한 트래픽 유발 및 시간적 소요 없이 원하는 정보를 한번에 빠르고 실시간으로 간편하게 얻을 수 있는 장점이 있다.

2. 관련 연구

H. P. Huy 외 2 명은 HyperText Markup Language(HTML) 파싱을 이용한 웹 서비스 게이트웨이 방법을 제안하였다[3]. 이 방법은 각각의 인터넷 환경에서의 필요한 정보를 웹 서비스 게이트웨이에서 요청하고 파싱하여 필요한 정보를 추출하고 사용자에게 정보를 제공해주는 방법이다. 요청을 수신하여 적절한 형태로 요청 정보를 변환하고 반환 된 HTML 에서 필요한 정보를 파싱하여 트리 형태로 저장한 후, Web Service Wrapper 로 필요한 정보를 분석하여 사이트별로 데이터를 추출하는 방법을 제안하였다. 또한, 각각의 웹 사이트에 따라 원하는 데이터를 추출할 수 있도록 알맞은 형태의 파서를 추가할 방법을 제안하였다. 우리의 모바일 통합 웹 서비스 게이트웨이는 앞서 제안한 방법처럼 각각의 게시판마다 특화된 파서를 어댑터 형식으로 추가할 수 있는 개념을 도입한다.

현재 스마트폰의 애플리케이션에서는 사용자를 위해 정보를 송/수신하는 2 가지 방법으로 원하는 정보를 실시간으로 알려주는 푸시 알림 방식과 알려진 데이터 정보를 상세히 전달받기 위한 콘텐츠 로드 방식이 있다.

푸시(Push) 알림 방식은 일정 주기마다 아주 작은 패킷으로 서버가 클라이언트에게 통신이 끊어지지 않게 확인을 하다가, 해당 서버에서 클라이언트에 필요한 정보를 제공할 이벤트가 발생하게 되면 해당 클라이언트에게 정보를 전송하는 통신 방법이다. 이러한 방법은 클라이언트가 서버에게 항상 확인하는 폴링(Polling) 방식보다 배터리 및 트래픽 효율성을 높일 수 있다. 우리는 푸시 알림을 위하여 IBM 의 오픈 소스 프로젝트인 Message Queuing Telemetry Transport(MQTT) 를 활용한다[4]. MQTT 는 경량적인 프로토콜로써, 푸시 알림 서비스를 구현함에 있어서, 메시징 크기에 제약이 없으며 이기종 플랫폼 간에 개발이 용이하다. 그림 1 은 MQTT 의 Publish 와 Subscribe 간의 통신을 간략하게 표현한 것이다.

그림 1 에서 각각의 클라이언트들은 자신이 Publish 할 Topic 과 Subscribe 할 Topic 을 Broker 서버에게 알린다. Broker 서버는 Publish 클라이언트의 메시지를 받아, Subscribe 클라이언트들이 알린 적절한 Topic 을 확인하여, 해당 메시지를 전송한다[5].

P. Julio 는 OMNeT++ 시뮬레이션 환경에서 MQTT 프로토콜에 대하여 메시지 크기 및 개수, 다수 클라이언트 간의 통신 성능을 평가하였다[6]. 또한, M. Prihodko 는 MQTT 와 HTTP 간의 처리를 비교하여 배터리 소모 및 성능을 측정하여 분석하고, MQTT 의 장점을 소개하였다[7]. MQTT 는 대표적으로는 Facebook Messenger 에서 사용[8]하고 있으며, 많은 오픈 소스 프로젝트들이 진행되고 있고, 다양한 실험을 통하여 그 유용성이 검증되었다[9-11].

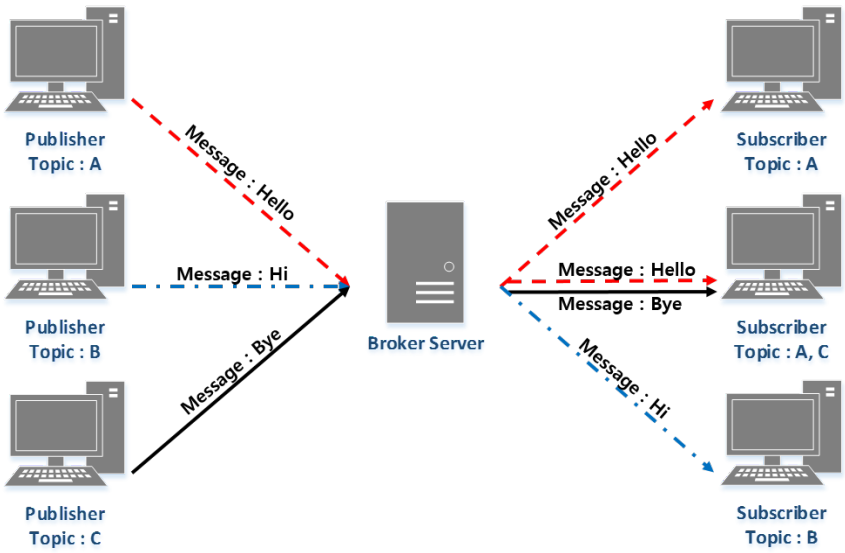

그림 1. MQTT를 활용한 통신 방법

콘텐츠 로드 방식은 이러한 푸시 알림을 사용자가 전달받게 되면 사용자가 푸시 알림에 대한 정확한 정보를 얻기 위하여 게이트웨이에 요청하는 통신 방법이다. 스마트폰의 애플리케이션에서는 이러한 파싱을 위한 작업으로 JavaScript Object Notation(JSON)을 활용하고 있다[12]. JSON 은 경량의 데이터 교환 포맷으로 데이터 타입이 자유로우며 읽고 쓰기가 용이하고, eXtensible Markup Language(XML)보다 데이터 형식을 간단하게 표현할 수 있는 장점을 지니고 있다[13-14]. 또한, XML 보다 경량적인 형식이기 때문에 모바일 환경에서 유용하게 사용이 가능하다.

3. 모바일 통합 웹 게시판 게이트웨이

이 장에서 우리는 모바일 통합 웹 게시판 게이트웨이의 개념과 설계 결과를 제시한다. 또한, 모바일 통합 웹 게시판 게이트웨이의 통신 방식인 푸시 알림 방식과 콘텐츠 로드 방식의 통신에 대하여 상세히 다루고 프로토콜 설계 결과를 제시한다.

3.1. 모바일 통합 웹 게시판 게이트웨이 개념

우리가 제안하는 모바일 통합 웹 게시판 게이트웨이는 사용자 클라이언트에 설치되는 모바일 통합 웹 게시판 애플리케이션과 각각의 정보를 필요로 하는 사이트들 사이에 위치한다. 그림 2 는 기존에 사용자가 직접 각각의 게시판에 접근하는 방법과 우리가 제안한 모바일 통합 웹 게시판 게이트웨이와 통합 애플리케이션을 통하여 접근하는 차이점을 보여준다.

게이트웨이는 각각의 게시판 서버에 인증하고, 해당 글들에 대하여 사용자의 설정에 따라 원하는 게시판의 정보에 대하여 푸시 알림 서비스를 해주게 된다. 사용자는 게이트웨이를 통하여 이러한 푸시 알림 서비스를 받게 되면, 사용자는 이에 대한 상세한 정보를 확인하기 위해 통합 애플리케이션을 이용하여 게이트웨이에 콘텐츠를 요청하게 된다. 사용자에 의해 콘텐츠들을 요청하게 되면, 게이트웨이는 사용자의 개인화 설정에 따라 각 게시판 서버에 콘텐츠를 요청하고 응답받은 정보들을 종합하여 사용자에게 제공한다.

이처럼 기존의 방법은 사용자가 수시로 게시판에 접근하여, 원하는 정보를 확인해야 하는 불편한 점이 존재하지만, 우리가 제안하는 모바일 통합 웹 게시판 게이트웨이는 원하는 게시판의 정보를 푸시 알림을 통하여 빠르게 얻을 수 있고, 사용자의 개인화 설정을 바탕으로 사용자가 원하는 게시판들을 통합하여 정보를 전달함으로써 여러 사이트를 방문할 필요 없이 빠른 정보를 한 번에 확인할 수 있다. 또한, 개인화된 웹 게시판 정보 제공과 실시간 웹 게시판에 대한 정보 제공으로 사용자가 수시로 게시판에 접근하여, 원하는 정보를 확인할 때 발생하는 불필요한 트래픽들을 감소할 수 있는 긍정적인 효과도 있다.

3.2. 모바일 통합 웹 게시판 게이트웨이 설계

본 논문에서 제안하는 모바일 통합 웹 게시판 게이트웨이 구성은 크게 웹 게시판 서버와 게이트웨이 간의 통신을 위한 Web Service Adapter 와 게이트웨이 계정 및 데이터 처리, 보안을 위한 Account Manager, Data Manager, Security Manager, 마지막으로 통합 애플리케이션과 게이트웨이 간의 통신을 위한 Broker, Contents Adapter 로 구성되어 있다.

우리는 각각의 게시판에 대하여 사용자의 인증을 받고 게시판 서버에 데이터를 요청하기 위한 방법으로 Web Service Adapter 를 이용한다. Web Service Adapter 는 각각의 서로 다른 오픈 프로젝트 게시판 종류에 대하여 서로 다른 인증 방식을 해결할 방안으로 게시판에 따른 인증 방식을 어댑터 형식으로 제공한다. 우리는 이러한 어댑터 형식을 이용하여 서로 다른 종류의 게시판 인증 방식에 유동적으로 대응할 수 있다.

게이트웨이는 각각의 사용자의 개인화 된 정보에 따라 해당 게시판들을 주기적으로 폴링하거나, 푸시

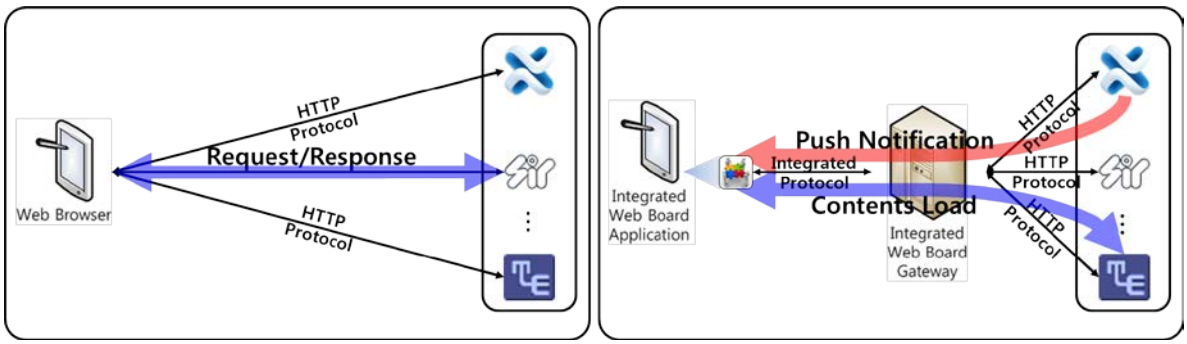

그림 2. 기존의 서로 다른 게시판의 접근 방법과 통합 웹 게시판 게이트웨이를 활용한 접근 방법

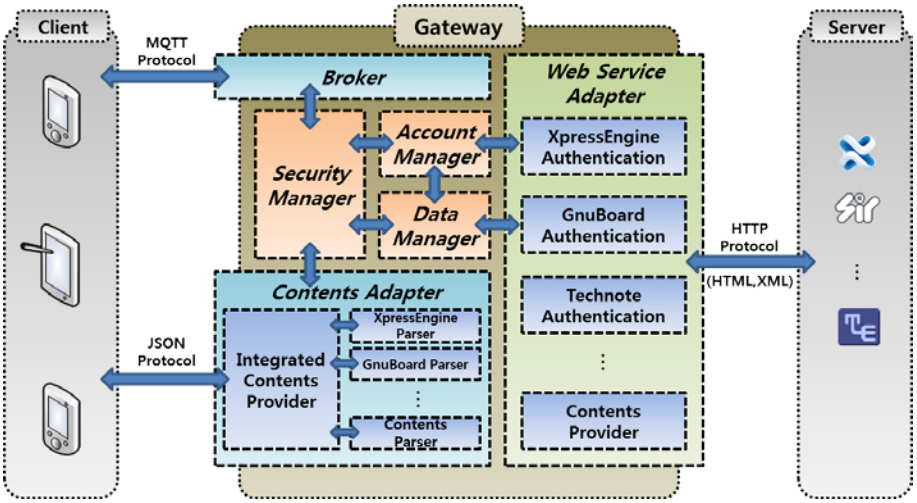

그림 3. 모바일 통합 웹 게시판 게이트웨이 구성도

알림을 받게 되면 Broker 모듈을 통하여 사용자에게 푸시 알림 서비스를 해주게 된다. 우리는 푸시 알림 서비스를 위하여 Broker 모듈로는 오픈 소스 프로젝트로 개발된 Mosquitto 를 활용한다[15].

그림 3 은 우리가 설계한 모바일 통합 웹 게시판 게이트웨이의 상세한 설계 구성도이다. 우리는 각각의 서로 다른 게시판에 대하여 데이터 추출에 알맞은 파서를 제공하기 위하여 Contents Adapter 개념을 도입하였다. Contents Adapter 는 사용자가 푸시 알림으로 서비스받은 정보를 상세하게 제공하기 위하여 기능을 수행한다. Contents Adapter 는 각각의 게시판에 대하여 데이터를 주고받으며, 각각의 게시판 파서를 통하여 데이터를 파싱하고 이러한 정보들은 Integrated Contents Provider 에서 종합하여 사용자에게 제공한다.

Security Manager 는 통합 애플리케이션과 게이트웨이 사이의 통신에 대한 보안을 담당하며, Account Manager 는 사용자의 통합된 ID 와 사용자 인증을 담당한다. Data Manager 는 통합된 ID 로 설정된 사용자의 게시판 개인화 설정 및 최신 글 알림을 위한 Check List, 푸시 알림의 Topictree 등 사용자가 게시판에 정보를 얻기 위한 모든 설정 정보들을 저장하고 관리한다.

3.3. 모바일 통합 웹 게시판 게이트웨이 푸시 알림 방식

이 절에서 우리는 푸시 알림 방식에 대하여 설명한다. 푸시 알림 방식은 사용자가 설정한 게시판에서 원하는 정보를 전달받게 되면 실시간으로 게이트웨이를 통하여 사용자에게 정보를 전달해줌으로써, 사용자는 사이트에 직접 수시로 방문하지 않고도 빠른 정보를 얻을 수 있다.

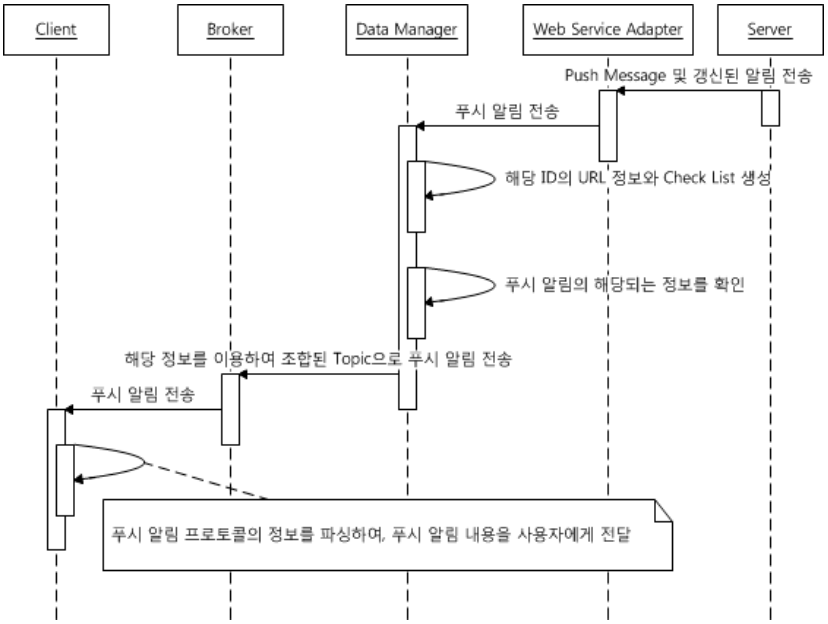

그림 4. 모바일 푸시 알림 시퀀스 다이어그램

그림 4 는 푸시 알림 방식에 대한 시퀀스 다이어그램을 나타낸 것이다. 게시판 서버에서 새로운 글들에 대해서 게이트웨이에 푸시 알림을 전송하게 되면, 게이트웨이의 Web Service Adapter 는 해당 푸시 알림의 내용을 받아 Data Manager 에게 전달하게 되고, Data Manager 는 이러한 내용을 분석하여 Check List 생성 및 푸시 알림에 해당하는 정보를 추출하여 데이터를 변환하게 된다. Data Manager 는 변환된 데이터를 우리가 설계한 푸시 알림 프로토콜 형식으로 정보를 생성하고 Broker 모듈을 통하여 사용자에게 푸시 알림을 서비스 해주게 된다.

3.3.1. 푸시 알림 Topictree 설계

MQTT 는 기본적으로 Publish/Subscribe 클라이언트와 Broker 서버 간의 푸시 알림 서비스를 위하여 Topic 을 사용한다. 그리고 MQTT 는 이러한 Topic 을 더욱 세분화할 수 있도록 Topictree 를 제공한다. 우리는 게시판에 대한 정보와 게시판의 URL 그리고 사용자의 ID 와 게시판의 정보들을 바탕으로 Topictree 를 더욱 세분화하여 정의하였다. 그림 5 는 푸시 알림을 위한 Topictree 의 예시이다.

Topictree 구조는 다음과 같다. 게이트웨이의 통합 ID 와 각각의 게시판 형식에 대한 정보, 그리고 각각의 게시판에 대한 URL 정보와 그 게시판의 사용자 ID 정보 및 게시판의 기능별로 세분화시켜 Topictree 를 구성한다. 이처럼 Topictree 는 각각의 게시판의 기능적 특징들을 게시물에 대한 작성자나 날짜, 카테고리별로 Topictree 를 설정할 수 있도록 세분화하였다. 사용자가 특정한 게시물 작성자나 정해진 날짜 및 원하는 카테고리에 새 글이 작성되면, 게이트웨이는 Topictree 를 활용하여 이러한 글들에 대한 푸시 알림 서비스를 사용자에게 제공하게 된다. 예를 들어 User/Integrate_ID/XpressEngine/Board_URL_Key/ID/Writer 로 Topictree 를 구성하여 Subscribe 를 요청하게 되면 해당 게시판의 Writer 가 작성하는 모든 글을 푸시 알림 서비스 받을 수 있게 된다.

그림 6 은 사용자가 원하는 정보에 대하여 새로운 Topictree 를 생성할 때의 수행되는 시퀀스 다이어그램이다. 앞서 설명한 대로 MQTT 는 기본적으로 Publish/Subscribe 간의 Topic 을 Broker 모듈에 알려줌으로써 통신이 이루어진다. 따라서 사용자가 게시판을 추가하고 게시판 기능별로 원하는 정보에 대하여 개인화를 설정할 때, 설정에 대한 Topictree 를 생성하고 난 후, Publish 역할을 수행하는 Data Manager 와 Subscribe 역할을 수행하는 사용자 모바일 통합 애플리케이션은 Broker 모듈에 Topictree 를 알려야 통신이 성립된다.

사용자가 새로운 게시판을 추가 요청하게 되면, Contents Adapter 는 요청을 받아 Account Manager 에게 전달한다. Account Manager 는 사용자에 대하여 인증을 받은 후, 게시판의 형태와 URL 등 게시판의 정보를 Data Manager 에게 전달한다. Data Manager 는 이러한 게시판 정보를 바탕으로 Topictree 를 생성하고, 사용자 데이터베이스 설정에 적용한다. 그리고 해당 Topictree 를 Broker 모듈에 알려줌으로써, Publish 역할을 담당하는 Data Manager 와 Broker 모듈 간의 Publish 통신이 성립된다. 다음으로 Data Manager 는 생성된 Topictree 를 Contents Adapter 에게 전달하여 사용자의 통합 애플리케이션에 Topictree 를 전송한다. 통합 애플리케이션에서 해당 Topictree 를 전송받게 되면, Topictree 를 게이트웨이 Broker 모듈에 알려줌으로써 통합 애플리케이션과 Broker 모듈 간의 Subscribe 통신이 성립하게 되고, Publish/Subscribe 간의 푸시 알림 방식의 통신이 가능하게 된다. 이렇게 Subscribe 역할을 수행하는 통합 애플리케이션에서는 사용자가 원하는 정보에 대한 Topictree 를 Broker 모듈에 알림으로써, 사용자가 원하는 정보를 푸시 알림으로 서비스를 받을 수 있다.

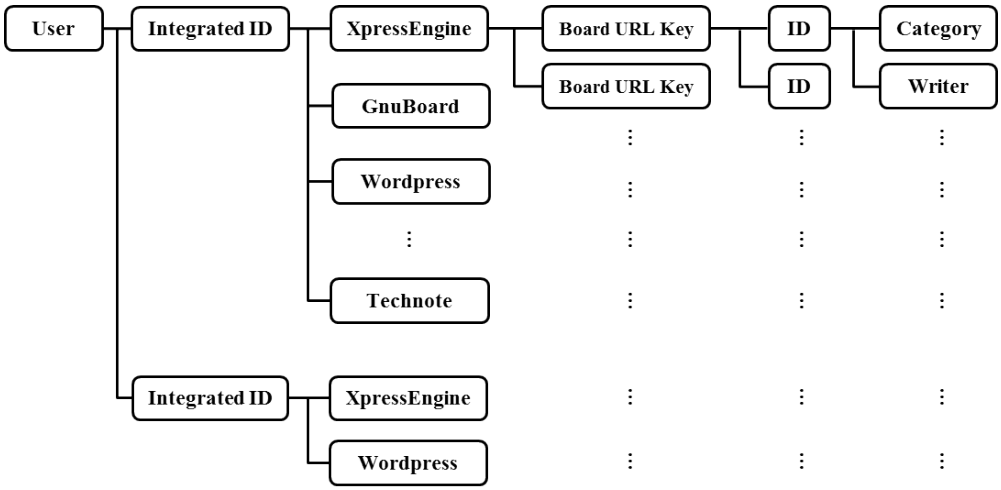

그림 5. 푸시 알림을 위한 Topictree 설계

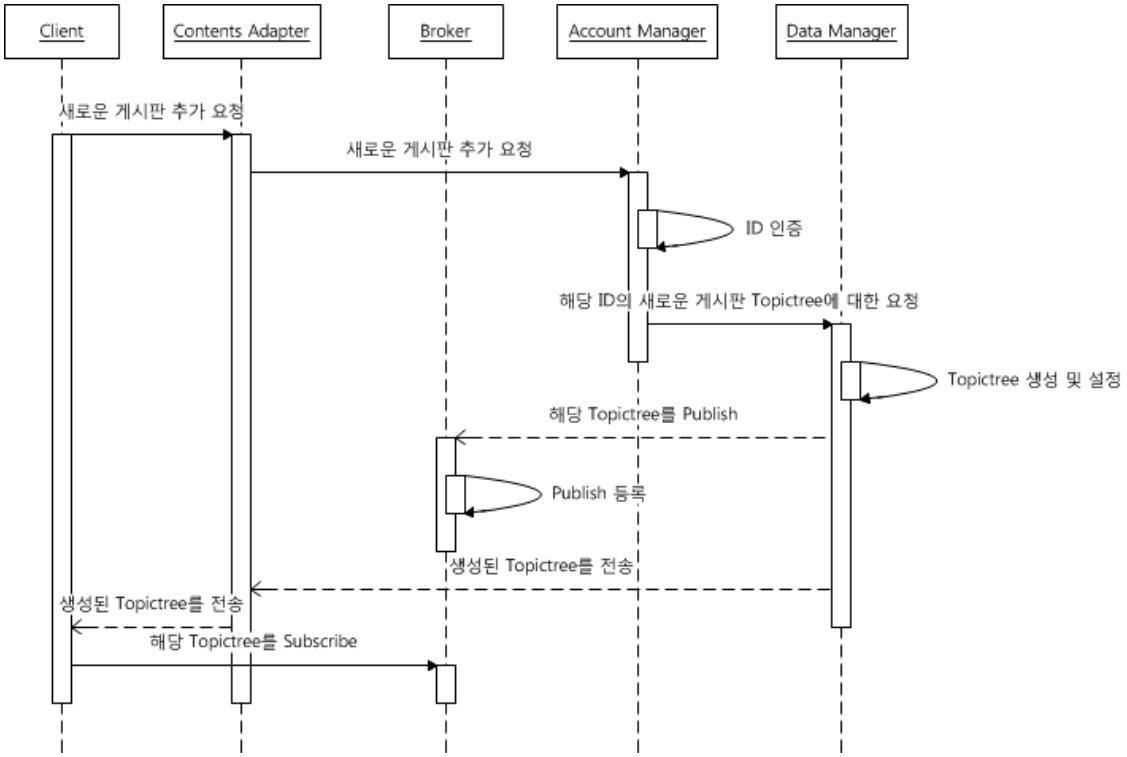

그림 6. Topictree 추가 시퀀스 다이어그램

3.3.2. 푸시 알림 프로토콜 설계

우리는 MQTT 를 이용하여 푸시 알림을 해줄 푸시 알림 프로토콜을 JSON 을 이용하여 설계하였다. 우리가 설계한 프로토콜은 MQTT 의 Payload 부분에 위치한다. 그림 7 은 JSON 을 이용하여 푸시 알림 프로토콜을 설계한 결과이다.

우리가 설계한 프로토콜은 사용자가 설정한 푸시 알림 서비스를 위하여 게시판 기능별로 세분화시킨 Topictree 의 정보들을 모두 포함한다. 웹 게시판 서버에서 새 글이나 새 댓글이 작성되면 게시물에 대한 정보를 게이트웨이에 전송한다. 게이트웨이는 웹 게시판 서버에서 수신받은 정보를 분석하고, Data Manager 에서 웹 게시판 서버의 수신 받은 정보를 바탕으로 등록된 Topictree 를 이용하여 Broker 모듈에 정보를 Publish 한다. Broker 모듈은 Data Manager 에 정보를 수신받게 되면, 해당 Topictree 로 Subscribe 요청한 통합 애플리케이션들은 푸시 알림을 Subscribe 함으로써 푸시 알림 서비스가 이루어지게 된다. 이렇게 게이트웨이에서 Broker 모듈을 통하여 사용자의 통합 애플리케이션까지 해당 푸시 알림 서비스 정보를 받게 되면, 통합 애플리케이션은 해당 MQTT 프로토콜의 Payload 부분을 파싱하여 정보를 추출한다. 만약 사용자가 푸시 알림에 대한 정보를 확인하게 되면, 사용자는 통합 애플리케이션을 통하여 게이트웨이에

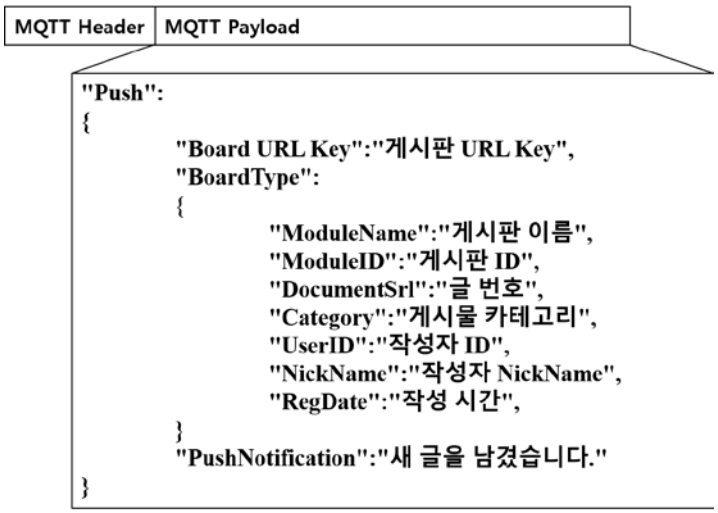

그림 7. 푸시 알림 프로토콜 설계

상세한 정보를 요청하게 된다. 게이트웨이는 사용자의 콘텐츠 로드에 대한 요청을 받게 되면 보다 더 상세한 정보를 얻기 위해 사용자의 개인화 된 설정을 참고하여 웹 게시판 서버에 콘텐츠 로드를 요청하게 된다. 다음 절은 통합 애플리케이션과 게이트웨이 그리고 웹 게시판 서버 간의 데이터를 요청/응답하는 콘텐츠 로드 방법에 대하여 설명한다.

3.4. 모바일 통합 웹 게시판 게이트웨이 콘텐츠 로드 방식

사용자가 푸시 알림을 받게 되면, 이러한 푸시 알림에 대하여 정확한 정보를 얻기 위해 통합 애플리케이션을 이용하여 게이트웨이에 콘텐츠에 대한 정보를 요청하게 된다. 게이트웨이는 사용자의 콘텐츠 로드 요청이 오게 되면, 사용자의 개인화된 설정에 따라 정보들을 각각의 게시판에 요청하고 응답 결과를 파싱하여 통합한 후 통합 애플리케이션을 통하여 푸시 알림에 대한 상세한 정보를 사용자에게 제공한다.

일단 사용자가 개인화 설정을 통하여 원하는 정보를 게이트웨이에 요청하게 되면, 게이트웨이의 Contents Adapter 는 요청한 사용자 설정 정보를 Data Manager 에 요청하게 된다. 사용자 설정 정보를 응답받게 되면, 해당 설정 정보를 통하여 Web Service Adapter 에 각각의 게시판마다 사용자가 원하는 데이터를 요청/응답받게 된다. 게이트웨이는 이러한 정보들을 Contents Adapter 의 각각 해당하는 게시판 파서를 통하여 데이터를 파싱하고 정보들을 추출하여, Contents Adapter 의 Integrated Contents Provider 에서 통합한다. 이렇게 통합된 정보를 JSON 을 활용하여 통합 애플리케이션에 콘텐츠를 전송하게 된다. 그림 8 은 모바일 통합 웹 게시판 게이트웨이의 콘텐츠 로드 방법에 대하여 시퀀스 다이어그램으로 나타낸 것이다.

3.4.1. 콘텐츠 로드 프로토콜 설계

우리가 설계한 콘텐츠 로드 프로토콜은 사용자가 푸시 알림을 받고, 새로운 게시판 정보를 상세히 얻기 위해 사용되는 프로토콜이다. 그림 9 는 JSON 형식으로 콘텐츠 로드 프로토콜을 설계한 결과이다.

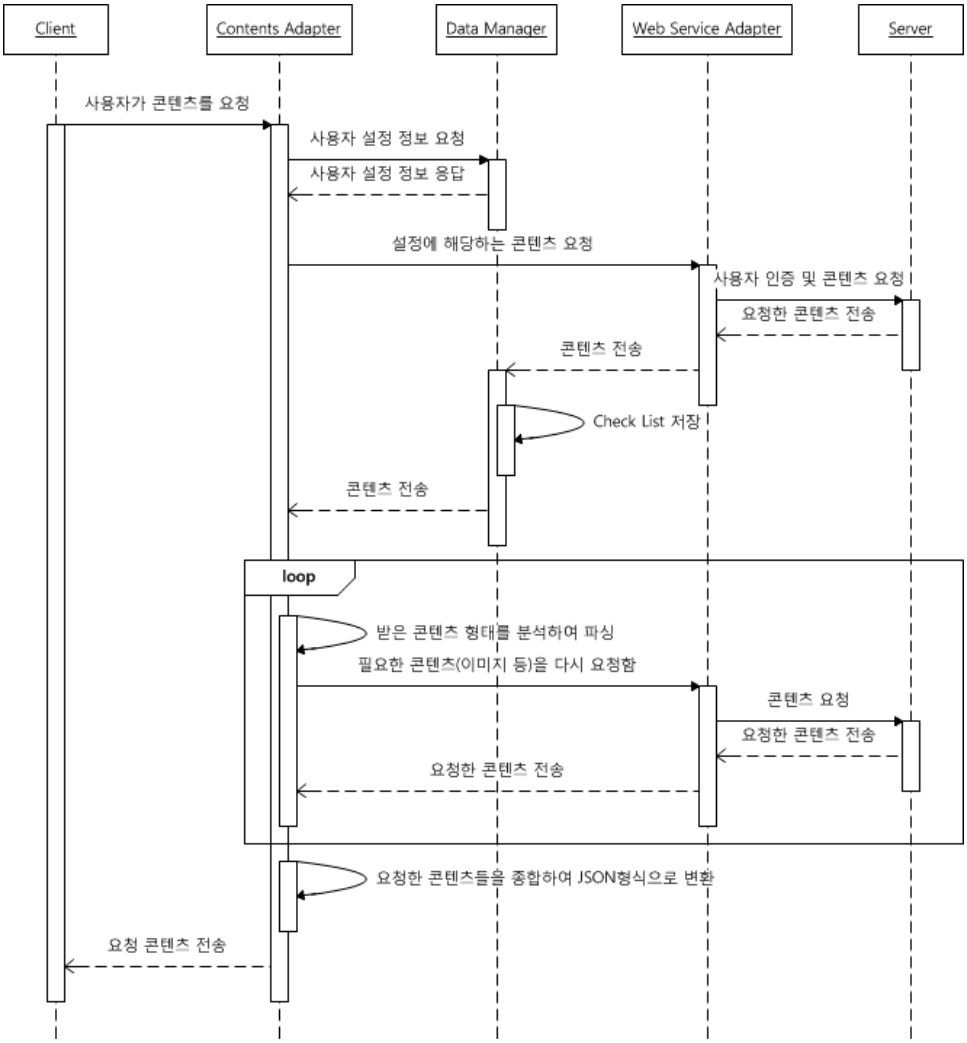

그림 8. 모바일 통합 웹 게시판 게이트웨이 콘텐츠 로드 시퀀스 다이어그램

```
"Request":
{
        "Integrated ID":"통합 ID",
        "Board URL Key":"게시판 URL Key",
        "BoardType":
        {
                "ModuleID":"게시판 ID",
                "DocumentSrl":"글 번호",
        }
}
```

```
"Response":
{
        "Board URL Key":"게시판 URL Key",
        "BoardType":
        {
                "ModuleName":"게시판 이름",
                "ModuleID":"게시판 ID",
                "DocumentSrl":"글 번호",
                "Category":"게시물 카테고리",
                "UserID":"작성자 ID",
                "NickName":"작성자 NickName",
                "RegDate":"작성 시간",
                "Title": " 게시판 제목",
                "Content":"게시판 본문 내용"
        }
}
```

그림 9. 콘텐츠 로드 프로토콜 설계

Contents Adapter 의 각각의 게시판 파서가 사용자가 원하는 정보를 파싱하여 추출하게 되면 이러한 정보들은 Contents Adapter 의 기능 중 하나인 Integrated Contents Provider 에서 정보를 통합하고 통합 애플리케이션에 정보를 제공하기 위하여 JSON 형식으로 변환한다.

사용자가 통합 애플리케이션을 통하여 통합 ID 와 게시판 URL key 그리고 Topictree 정보를 게이트웨이에 전달하여 정보를 요청하게 되면, 게이트웨이는 이러한 사용자 정보들을 이용하여 Data Manager 에 설정 정보를 받아오고, 설정 정보를 토대로 사용자의 개인화 설정에 따라 정보를 모아 전달하게 된다.

4. 모바일 통합 웹 게시판 게이트웨이 구현

이 장에서 우리는 통합 웹 게시판 게이트웨이의 푸시 알림 서비스와 콘텐츠 로드에 대한 구현 결과를 제시한다. 우리는 서로 다른 2 개의 XpressEngine 게시판 서버에 대하여 모바일 통합 웹 게시판 게이트웨이에 대한 구현을 테스트하였다.

4.1. 모바일 통합 웹 게시판 게이트웨이 실험 환경

그림 10 은 모바일 통합 웹 게시판 게이트웨이의 구현 결과를 테스트한 실험 환경이다. 게이트웨이와 2 개의 XpressEngine 게시판 서버 OS 는 Linux 환경의 CentOS 6.3 Final 을 사용하였으며, 통합 애플리케이션은 Android 4.0.3 환경에서 3G 망을 이용하여 실제 네트워크와 유사한 환경으로 구현 결과를 테스트하였다.

4.2. 모바일 통합 웹 게시판 게이트웨이 푸시 알림 구현 결과

우리는 이 절에서 모바일 통합 웹 게시판 게이트웨이 푸시 알림 서비스를 구현한 결과를 제시한다. 푸시 알림 서비스를 테스트하기 위하여 우리는 안드로이드 MQTT 라이브러리인 IBM 의 IA92 를 활용[16]하여 안드로이드 환경에서 푸시 알림 서비스를 구현하였다. 그림 11 은 게시판 글에 대한 댓글을 푸시 알림으로 사용자의 통합 애플리케이션에 정보를 전달한 결과이다.

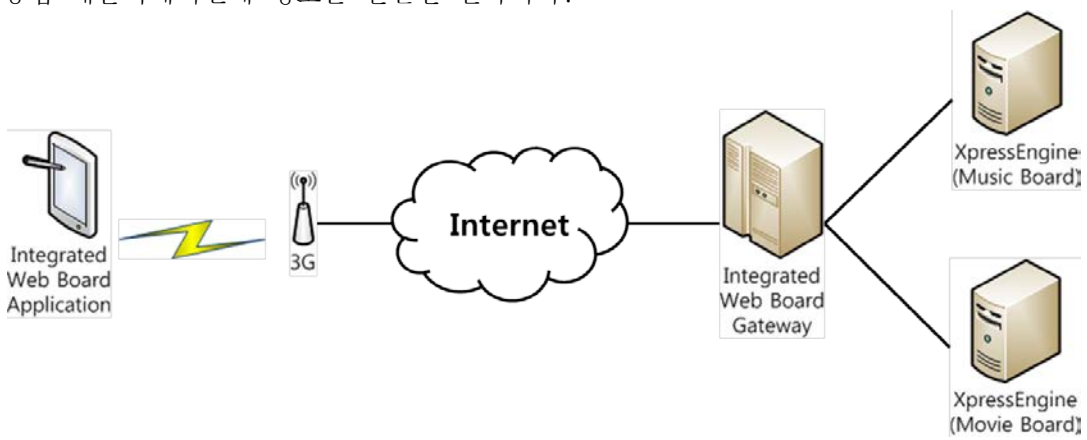

그림 10. 모바일 통합 웹 게시판 게이트웨이 구현 테스트 실험 환경

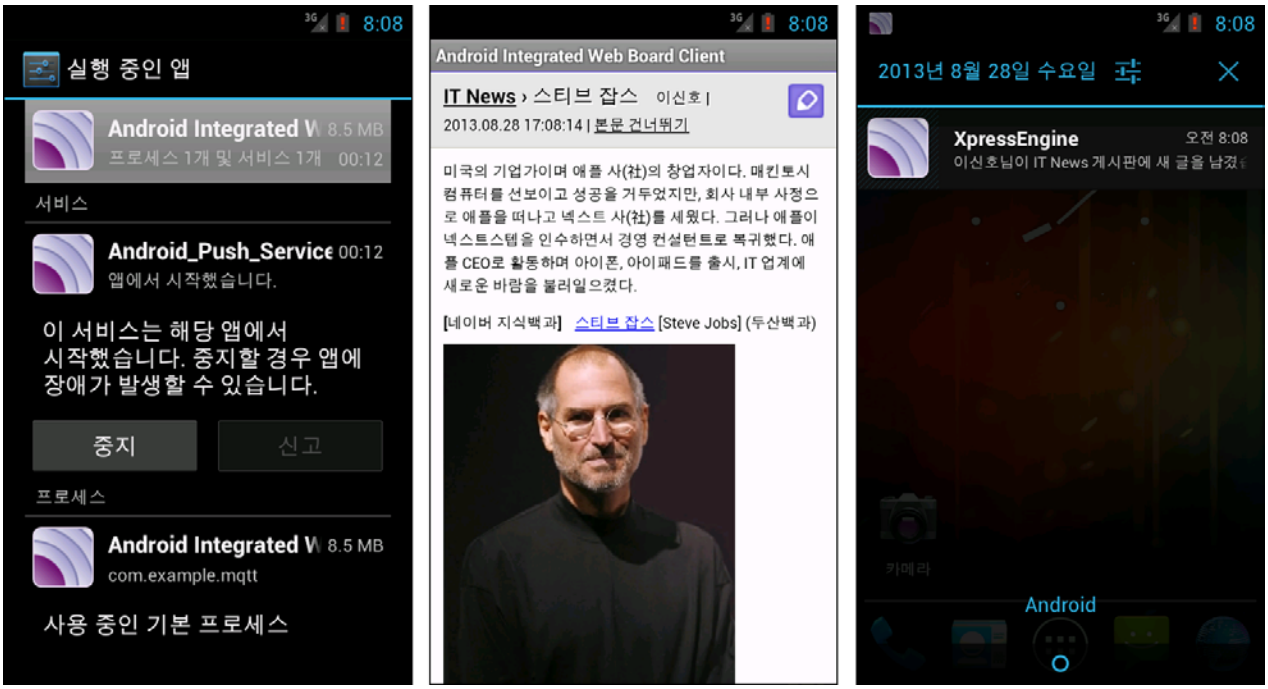

그림 11. 모바일 통합 웹 게시판 게이트웨이 푸시 알림 구현

우리는 안드로이드의 애플리케이션에서 서비스 형태로 개발하여, 백그라운드 작업을 통한 푸시 알림에 대한 정보를 실시간으로 받을 수 있도록 구현하였다. 그림처럼 사용자가 게시판에 작성한 글에서 다른 사용자의 댓글을 확인하여 실시간으로 안드로이드 앱에 푸시 알림을 해준 것을 확인할 수 있다.

4.3. 모바일 통합 웹 게시판 게이트웨이 콘텐츠 로드 구현 결과

우리는 이 절에서 사용자가 개인화 설정을 통하여 서로 다른 게시판들을 통합하여 정보를 받을 수 있도록 통신하는 콘텐츠 로드를 구현한 결과를 제시한다. 그림 12 는 모바일 통합 웹 게시판 게이트웨이의 콘텐츠 로드 구현 결과이다.

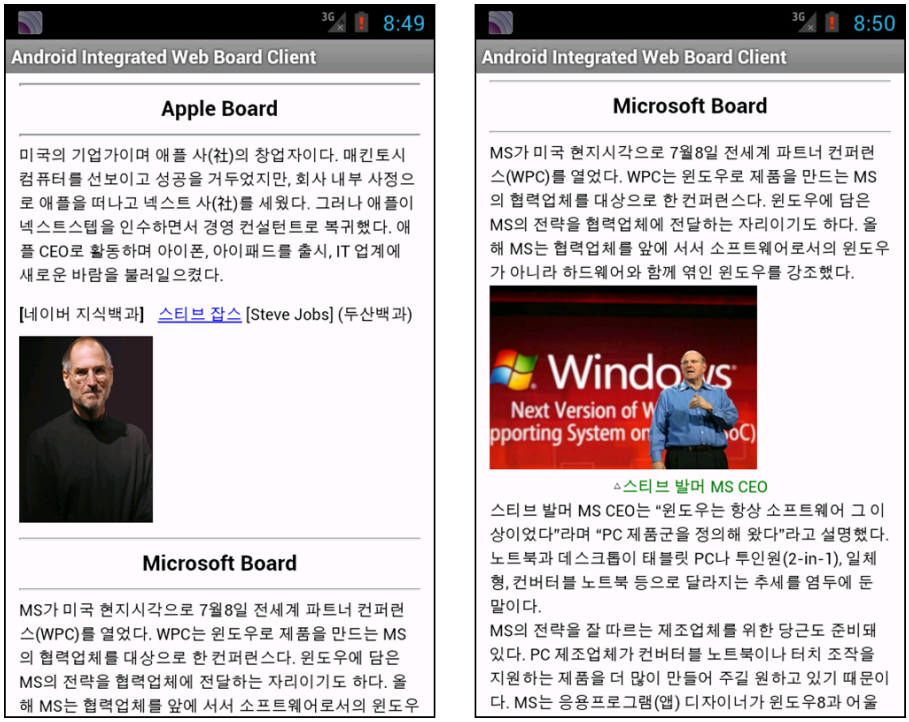

그림 12. 모바일 통합 웹 게시판 게이트웨이 콘텐츠 로드 구현

XpressEngine 은 JSON 을 이용하여 게시판에 콘텐츠에 대한 요청이 가능하다. 따라서 우리가 제시한 Contents Adapter 모듈의 어댑터 형식인 XpressEngine 게시판 파서는 JSON 을 기반으로 하여 개발하였다. 서로 다른 2 개의 XpressEngine 게시판을 게이트웨이를 통하여 정보를 통합하고, 그 결과를 통합 애플리케이션으로 정보를 제공해줌으로써 사용자는 상세한 정보를 한 번에 확인할 수 있다.

푸시 알림을 받게 되면 사용자는 상세한 정보를 얻기 위해 게이트웨이에 사용자가 개인화 설정을 한 게시판들 정보를 종합하여 전달하게 된다. 각각의 다른 게시판들의 내용을 통합하여 하나의 애플리케이션에서 정보를 제공한다. 이러한 방법으로 우리는 여러 사이트를 방문하지 않고도 한번에 종합적인 정보들을 실시간으로 전달받아 확인할 수 있다.

5. 결론 및 향후 연구

우리는 빠르고 쉽게 다양한 사이트에서 원하는 정보를 얻기 위한 방법으로 모바일 통합 웹 게시판 게이트웨이에 대한 설계 결과를 제시하였다. 제시한 게이트웨이 설계 결과를 바탕으로 간단한 예시를 통해 구현한 결과를 제시하였으며, 모바일 환경에서의 장점인 푸시 알림을 통하여 원하는 정보를 실시간으로 빠르게 정보를 얻을 수 있는 것을 확인하였다.

아직 각각의 여러 가지 게시판 형태에 대하여 다양한 파서를 제공할 수 있도록 하는 파서 설정 기능은 현재 개발 중이다. 우리는 향후 연구로 Contents Adapter 에서 각각의 게시판에 대한 어댑터 형식의 파서 설정에 대한 구체적인 방법과 개인정보를 이용한 인증방식의 보안 부분을 설계 및 구현할 계획이다. 또한, 웹 캐시 서버를 이용하여 트래픽을 절감할 방법에 대한 연구와 향후 연구들을 통합하여 게이트웨이에 대한 성능 분석을 진행할 것이다.

6. 참고 문헌

[1] 이지스, “대한민국 누리꾼 현황 보고서”, http://egis.onoffkorea.com/.

[2] KISA ISIS, “2012년 인터넷이용실태조사 최종보고서”, http://isis.nida.or.kr/.

[3] H. P. Huy, T. Kawamura, and T. Hasegawa, “Web Service Gateway – a step forward to e-business,” Proceedings of the IEEE International Conference on Web Services (ICWS), pp. 648-655, California, USA, Jul. 2004.

[4] IBM, “The MQTT protocol,” http://www.mqtt.org/.

[5] P. Th. Eugster, P. A. Felber, R. Guerraoui, and A. Kermarrec. “The many faces of publish/subscribe,” Journal ACM Computing Surveys (CSUR), vol. 35, Issue 2, pp. 114-131, Jun. 2003.

[6] P. Julio, “MQTT Performance Analysis with OMNeT++,” M.S. thesis, IBM Zurich Research Laboratory, Institut Eurécom, Sep. 2005.

[7] M. Prihodko, “Energy Consumption in Location Sharing Protocols for Android Applications,” Linkopings University, Oct. 2012.

[8] L. Zhang, “Building Facebook Messenger,” http://www.facebook.com/notes/facebook-engineering-/building-facebook-messenger/10150259350998920/.

[9] S. Nicholas, “Power Profiling: HTTPS Long Polling vs. MQTT with SSL, on Android,” http://stephendnicholas.com/archives/1217/.

[10] M. Collina, G. E. Corazza, A. Vanelli-Coralli, “Introducing the QEST broker: Scaling the IoT by bridging MQTT and REST,” Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on, pp.36-41, Sydney, Australia, Sep. 2012.

[11] M. Ma, Y. Huang, C. H. Chu, P. Wang, “User-Driven Cloud Transportation System for Smart Driving,” Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on, pp 658-665, Taipei, Taiwan, Dec. 2012.

[12] D. Crockford, “Introducing Json,” http://www.json.org/.

[13] N. Nurseitov, M. Paulson, R. Reynolds, and C. Izurieta “Comparison of JSON and XML data interchange formats: A case study,” the ISCA 22nd International Conference on Computers and Their Applications in Industry and Engineering (CAINE 2009), pp. 157-162, San Francisco, California, USA, Nov. 2009.

[14] C. Rodrigues, J. Afonso and P. Tomé, “Mobile Application Webservice Performance Analysis: Restful Services with JSON and XML,” ENTERprise Information Systems Communications in Computer and Information Science Volume 220, pp. 162-169, Oct, 2011.

[15] Mosquitto, “Mosquitto,” http://www.mosquitto.org/.

[16] IBM, “IA92: WBI Brokers – Java implementation of WebSphere MQ Telemetry transport,” http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24006006/.

이 신 호

2012년 2월 계명대학교 컴퓨터공학과 학사

2012년 3월 ~ 현재 계명대학교컴퓨터공학과 석사과정

<관심분야> SNS, Mobile 네트워크 관리

권 동 우

2010년 2월 계명대학교 컴퓨터공학과 학사

2010~2012년 계명대학교 컴퓨터공학과 석사

<관심분야> 인터넷 침입 예측, Firewall, 네트워크 관리

김 현 우

2010년 8월 계명대학교 컴퓨터공학과 학사

2010~2012년 계명대학교 컴퓨터공학과 석사

2012년 9월 ~ 현재 계명대학교컴퓨터공학과 박사과정

<관심분야> Firewall, 네트워크 관리

주 홍 택

1989년 한국과학기술원 전자계산학과 학사

1989 ~ 1991년 포항공과대학교컴퓨터공학과 석사

1991 ~ 1997년 대우통신 종합연구소 선임연구원

1997 ~ 2002년 포항공과대학교컴퓨터공학과 박사

2002년 ~ 현재 계명대학교 컴퓨터공학과 부교수

<관심분야> 네트워크 및 시스템 관리, Embedded 웹 서버를 이용한 네트워크 요소관리, SyncML, 인터넷 침입 예측, 네트워크 보안