
스마트폰 상에서 프로젝트 관리를 위한 소셜 네트워킹 서비스 기반의 일정 통지 및 이슈 추적 시스템

강대기* · 장원태*

Project Schedule Notification and Issue Tracking System Based on Social Networking Service on a Smartphone

Dae-Ki Kang* · Won-Tae Chang*

요 약

본 논문에서는 스마트폰 상에서 프로젝트 관리를 위한 소셜 네트워킹 서비스 기반의 일정 통지 및 이슈 추적 시스템을 새롭게 제안하고자 한다. 제안된 시스템은 서버 시스템과 클라이언트 시스템으로 나뉜다. 서버 시스템은 기존의 소셜 네트워킹 서비스를 적극적으로 활용하여 참여자에게 프로젝트 작업의 데드라인에 대한 통지와 이슈 추적을 가능하게 한다. 스마트폰 상에서 개발된 클라이언트 시스템은 프로젝트 참여자에게 간트 차트를 디스플레이 해 주고, 자신의 일정을 입력 내지 수정할 수 있도록 한다. 제안된 시스템은 기존의 스마트폰이 가지고 있는 이동성과 소셜 네트워킹 서비스가 가지는 연결성을 적극적으로 활용하여, 프로젝트 개발에서의 일정 통지 및 이슈 추적에 이용하며, 이를 통해 소셜 네트워킹 서비스의 새로운 응용 가능성을 제시한다.

ABSTRACT

In this paper, we propose a novel project schedule notification and issue tracking system based on a social networking service for project management on a smartphone. The proposed system has a server subsystem and a client subsystem. The server is in charge of enabling a deadline notification and an issue tracking of the project to project participants by exploiting a legacy social networking service. The client running on a smartphone displays timelines of the project schedule using Gantt chart and let the project participant edit their schedule. The proposed system combines the mobility of smartphones and the connectivity of social networking services and apply them to schedule notification and issue tracking, which demonstrates a novel usage of social networking services.

키워드

안드로이드, 스마트폰, 소셜 네트워킹 서비스, 트위터, 일정 통지, 이슈 추적 시스템

Key word

Android, Smartphone, Social Networking Service, Twitter, Schedule Notification, Issue Tracking System

I. 서 론

최근 들어, 안드로이드 [1] 및 아이폰과 같은 스마트폰 (Smartphone) [2] 시스템이 차세대의 컴퓨팅 환경에 계속 퍼져나가고 있으며, 가까운 장래에는 PC의 개수보다 휴대폰, 특히 스마트폰의 개수가 더 많아질 것이라는 예측이 지배적이다. 또한 다른 한편으로는, 트위터[3]나 미투데이와 같은 소셜 네트워킹 서비스[4]의 사용자의 수가 기하급수적으로 늘어나면서 활발해지는 추세이다. 그러나, 이러한 스마트폰과 소셜 네트워킹 서비스를 적극적으로 활용하는 응용 예는 현재 초기 성장 단계로 계속 소개되고 있으나, 아직 많지 않은 대신 많은 가능성을 가지고 있는 상황이다.

본 연구에서 소개할 응용은 프로젝트 관리를 위한 일정 통지 및 이슈 추적 시스템이다. 이러한 응용을 개발하고자 하는 동기를 설명하자면, 소프트웨어 개발 소프트웨어 프로젝트를 수행함에 있어 개발 일정을 설정하는 것은 가장 어려운 문제 중 하나이다. 대부분의 개발자들은 정확한 개발 일정을 설정하지 않으려고 하는데, 그 이유는 그것이 정말로 어렵기 때문이고 또한 잘 지켜지지 않기 때문이다. 예를 들어 증거 기반 스케줄링 (Evidence Based Scheduling) 을 주장한 Joel Spolsky[5]는 정확한 개발 일정 산정을 위한 다음의 4가지 규칙들을 제시한 바 있다.

1. 실제 코딩을 하는 프로그래머만이 예상되는 개발 일정을 정할 수 있도록 한다.
2. 버그가 발견되면, 그 버그를 고치는데 사용하는 시간을 일정에 반영하도록 한다.
3. 관리자가 개발 일정을 임의로 줄이지 않게 한다.
4. 개발 일정은 나무 블록을 담은 상자와 같아서, 나무 블록이 많으면 나무 블록의 개수를 줄이던지 상자를 늘려야 한다.

이러한 대체적인 규칙들에도 불구하고, 일부 소프트웨어 프로젝트에서는 전체 개발 일정을 기록한 파일이 개발 초기 단계에 만들어지고 프로젝트 개발자들 모두에게 열람된 후에 개발 시스템의 어느 디렉토리가 저장되지만, 전체 개발 기간 동안 모두에게 잊혀지고 마는 경우가 있다. 만일 소프트웨어 개발자들이 개발 일정과

개발 과정에서의 이슈들에 대해 쉽게 접근하고 참조할 수 있다면 소프트웨어 개발 과정의 일정 관리 문제에 다 소나마 기여할 수 있을 것이다.

본 논문에서는 특히 소프트웨어 개발자들이 모바일 환경을 심분 활용할 수 있도록 하기 위해 스마트폰 상에서 프로젝트 관리[6]를 위한 소셜 네트워킹 서비스 (Social Networking Service; SNS) 기반의 일정 통지 및 이슈 추적 시스템을 제안하고자 한다. 제안된 시스템은 서버 시스템과 클라이언트 시스템으로 나뉜다. 서버 시스템은 기존의 소셜 네트워킹 서비스를 적극적으로 활용하여 참여자에게 프로젝트 작업의 데드라인에 대한 통지와 이슈 추적을 가능하게 한다. 스마트폰 상에서 개발된 클라이언트 시스템은 프로젝트 참여자에게 간트 차트 (Gantt Chart) [7]를 디스플레이해 주고, 자신의 일정을 입력 내지 수정할 수 있도록 한다. 제안된 시스템은 기존의 스마트폰이 가지고 있는 이동성과 소셜 네트워킹 서비스가 가지는 연결성을 적극적으로 활용하여, 프로젝트 개발에서의 일정 통지 및 이슈 추적에 이용하며, 이를 통해 소셜 네트워킹 서비스의 새로운 응용 가능성을 제시한다. 제안된 시스템은 전세계적으로 공개된 SNS 서비스와 오픈 소스를 적극적으로 채택하여 효율적으로 구현되어, 보안성이 크게 중요하지 않은 소규모 경량 프로젝트나 오픈 소스 프로젝트 개발에 활용 가능성이 높으리라 여겨진다.

본 논문의 구성은 다음과 같다. 2장에서는 선행 연구들에 대한 소개를 한다. 3장에서는 본 논문에서 제안한 시스템을 설명한다. 마지막으로, 4장에서는 결론을 내리고 전체를 요약하도록 한다.

II. 선행 연구 (Previous Work)

2.1 안드로이드 (Android)

안드로이드 (Android)란 모바일 시스템을 위한 공개 운영체제 및 미들웨어와 응용 프로그램들의 집합으로 구글(Google)에서 만들었다. 구글에서는 안드로이드에서 응용 프로그램을 개발하기 위한 Software Development Kit (SDK) 를 제공하고 있으며, 이 SDK는 사용자가 응용 프로그램을 자바 언어 (Java Language)

로 개발할 수 있게 한다. 그림 1은 에뮬레이터에서 수행되는 안드로이드 운영체제에서 대표적인 일정 관리 프로그램인 구글 캘린더(Google Calendar)를 수행한 모습을 나타내고 있다. 그러나 구글 캘린더와 같은 일정 관리 프로그램은 일반 사용자들을 위한 일정 관리에만 유용할 뿐, 소프트웨어 개발과 같은 프로젝트 일정 관리에는 적합하지 않은 문제점이 있다.



그림 1. Android 기반의 Google Calendar
Fig. 1 Google Calendar on Android

2.2 트위터(Twitter)

트위터 (Twitter)는 소셜 네트워킹 (Social Networking Service; SNS) 겸 마이크로 블로깅 (micro-blogging) 서비스 중 대표적인 서비스로, 현재 가장 많은 사용자들 거느리고 있다. 사용자는 트위터 웹 사이트, SMS, 또는 외부 응용 프로그램을 통해 업데이트라는 이름으로 최대 140자 길이의 메시지를 보낼 수 있다. 또한 다른 사용자들이 보내는 업데이트들을 트위터 웹 사이트, SMS, 또는 외부 응용 프로그램을 통해 받을 수 있다. 특정 사용자의 메시지를 받아보기 위해서는 우선 그 특정 사용자를 팔로우(follow)해야 하고, 그 다음에 그 사용자가 그 팔로우 요구를 기본 옵션이나 사후 승인 방식으로 허용해야 한다. 유명 연예인의 경우, 수천명 또는 수만명이 넘는 팔로워(follower)가 있을 수 있다. 그림 2는 트위터 소셜 네트워킹 서비스의 한 예이다. 자신과 자신이 팔로우하는 사용자들의 업데이트들로 구성된 타임라인(timeline)을 일목요연하게 볼 수 있을 것을 알 수 있다.

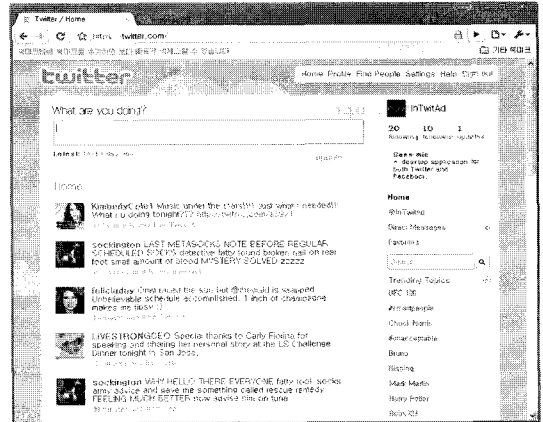


그림 2. 트위터 소셜 네트워킹 서비스
Fig. 2 Twitter Social Networking Service

트위터에서는 사용자가 자신의 휴대 전화기 또는 무선 단말기를 등록하면, 자신과 자신이 팔로우하는 사용자들의 업데이트를 SMS 문자로 받아볼 수 있다.

최근 몇 년간 미국에서의 트위터의 열풍은 놀라울 정도였으며, 대기업들도 트위터의 상업적 가능성에 주목하고 있다. 예를 들어, Dell, Pepsi, 그리고 Ford 등등의 수많은 여러 대기업이 트위터를 통해 판촉활동에 나서고 있다. 트위터와 같이 SNS와 마이크로 블로깅이 연결된 서비스로는, 현재 미국에서는 12 개가 넘는 비슷한 서비스가 운영되고 있으며, 국내에서는 미투데이(me2DAY)가 운영 중이다.

2.3 예약 트위터 서비스들

(Scheduled Twitter Services)

앞에서 언급한 바와 같이 트위터는 많은 사용자들 거느리고 있으며, 개인은 물론 기업용 어플리케이션으로도 다방면에 응용되고 있다.

그 중에서 스케줄 관리를 위한 트위터의 응용으로는 트웨이터 (Twaitter; <http://www.twaitter.com/>)가 있다. 스케줄 관리를 위한 트위터 응용 프로그램은 업데이트를 하자마자 바로 사용자에게 전송하는 것이 아니라, 사용자가 정한 시간에 사용자에게 전송하며, 특정 기간 동안 특정 시간에 반복적으로 전송하게 설정하는 것도 가능한 서비스이다.

그림 3은 트웨이터의 실행 화면이다. 실행 화면을 설명하자면, 트위터 사용자가 자신이 지금 뭘 하는지에 대

해 Send Now 버튼을 눌러 업데이트를 하면 네트워크와 서버의 속도에 따라 실시간으로 내용이 포스트되는 반면, Twait 버튼을 누르면 사용자는 정해진 시간에 특정 메시지를 업데이트를 하도록 미리 예약을 하거나, 매일, 매주, 매달, 또는 매년 중 정해진 시간에 특정 메시지를 업데이트하도록 설정할 수 있는 것이다.

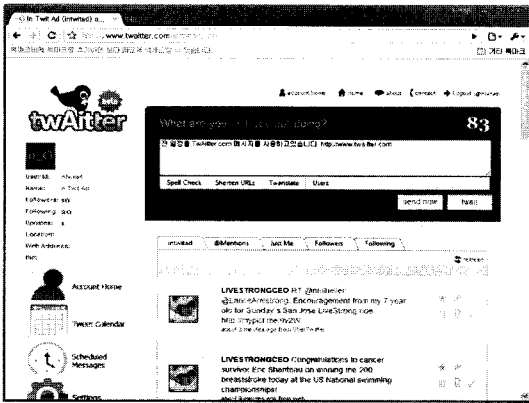


그림 3. 트웨이터: 소셜 네트워킹 기반 스케줄 관리 서비스

Fig. 3. Twitter: Social Networking based Schedule Management Service

비슷한 서비스로 퓨처트윗(FutureTweets; <http://futuretweets.com/>), 트윗리스펀스(TwitResponse; <http://twitresponse.com/>), 트윗메신저(TwitMessenger; <http://twitmessenger.com/>), 투우퍼(Twuffer; <http://twuffer.com/>), 트윗레이터(Tweet Later; <http://www.tweetlater.com/>) 등이 있다.

그러나, 이러한 시스템들은 기본적으로 일반 사용자가 범용적인 용도로 스케줄을 사용하도록 되어 있어서, 소프트웨어 개발자가 프로젝트의 수행 일정 및 이슈 추적에 사용하기에는 기능적으로 부족하며 모바일 시스템 개발 환경에도 적합하지 않다는 문제점이 있다.

2.4 간트 차트 (Gantt Chart)

간트 차트는 프로젝트의 일정을 표현하기 위한 도구로, 주로 바 그래프 (bar graph) 형태로 프로젝트의 일정을 나타낸다. 표 1은 소프트웨어 개발 작업의 한 예를 나타낸 것이다. 표 1과 같은 작업 스케줄은 간트 차트 그림 4와 같이 나타낼 수 있다. 첫 번째 작업은 2009

년 5월 1일부터 2009년 5월 21일까지로 작업 기간은 20일이다. 표 1의 예는 이러한 작업들이 9개가 있는 경우가 되겠다.

표 1. 소프트웨어 개발 스케줄의 예
Table 1. One Example of Software Development Schedule

작업	시작일	기간(일)	종료일
1	2009/05/01	20	2009/05/21
2	2009/05/11	5	2009/05/16
3	2009/05/21	30	2009/06/20
4	2009/05/31	11	2009/06/11
5	2009/06/10	23	2009/07/03
6	2009/06/20	25	2009/07/15
7	2009/06/30	48	2009/08/17
8	2009/07/10	20	2009/07/30
9	2009/07/20	5	2009/07/25

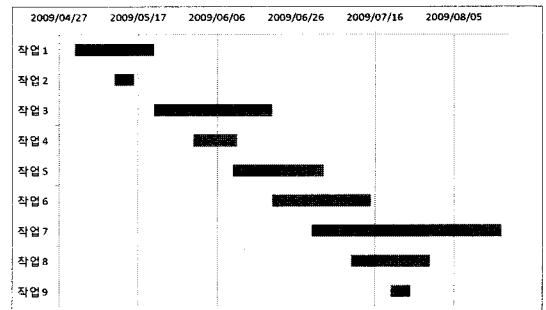


그림 4. <표 1>에 대한 간트 차트
Fig. 4. Gantt Chart for Table 1

간트 차트를 통해 각 작업 별로 업무의 시작과 끝을 알 수 있으며, 다른 작업과 얼마나 겹치는가도 쉽게 파악할 수 있다. 이러한 간트 차트는 또한 각 작업 간의 관계를 나타내도록 확장될 수도 있다.

스마트폰 상에서 간트 차트를 통해 다양한 프로젝트를 다루는 소프트웨어로는 Windows Mobile의 포켓 간트 차트 (Pocket Gantt Chart) 등의 프로그램이 있다. 이러한 프로그램들은 개발 일정을 위한 데이터베이스를 데스크탑 서버 시스템과 동기화하는 전통적인 방식으로 작동한다. 그림 5는 포켓 간트 차트 프로그램의 실행 화면이다.

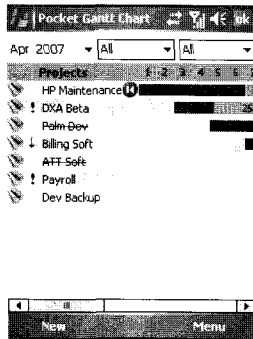


그림 5. 포켓 간트 차트
Fig. 5. Pocket Gantt Chart

이러한 시스템들은 개발 일정을 모바일 환경에 맞추어 관리하는 효과는 있으나, 프로젝트 개발상의 이슈 추적 및 개발자들 간의 의사소통의 기능은 없는 단점이 있다. 본 논문에서 제안하는 시스템은 이러한 개발자들 간의 소통의 문제를 소셜 네트워킹 서비스를 활용하여 해결하였다.

2.5 이슈 추적 (Issue Tracking)

이슈 추적 시스템(issue tracking system)은 [8] 작게는 프로젝트 수행, 크게는 기업 운영에서 발생하는 문제들을 기록하고 그 문제들에 대한 설명, 논의, 관련 질문과 대답, 해결책 등을 리스트로 유지하고 기록하는 시스템이다. 특히 소프트웨어 개발 중에서는 주로 버그에 대한 설명, 논의, 해결책 등을 유지하고 기록하게 된다.

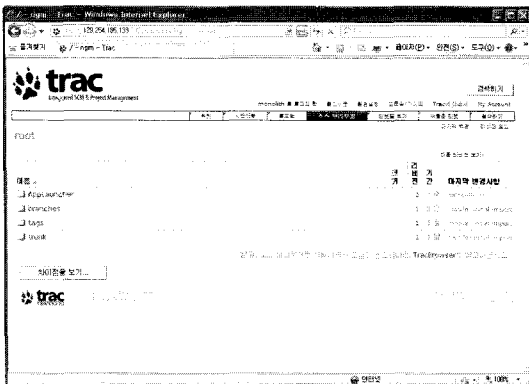


그림 6. 트랙(Trac): 프로젝트 관리 도구
Fig. 6. Trac: Project Management Tool

대표적인 이슈 추적 시스템으로는 그림 6에서 보는 트랙(Trac)이 있다. 트랙은 옛zeit 소프트웨어가 개발한 오픈소스 웹 기반 프로젝트 관리 겸 버그 추적 도구로, 버전 관리 소프트웨어의 웹 인터페이스 제공, 각종 개선점과 버그와 같은 프로젝트의 이슈 추적 (이슈를 트랙에선 티켓이라고 함), 그리고 위키를 통한 문서 관리 및 각 리소스 연동을 주 기능으로 한다.

그러나, 이러한 이슈 추적 시스템들은 데스크탑 컴퓨터와 웹 상에서 작업 스케줄을 관리하고 개발 상의 이슈를 추적하는 데에만 특화되어 있고, 모바일 환경에 적합하지 않다는 문제점이 있다. 예를 들어 Trac의 경우, 모바일 환경에서 Trac 데이터에 접근하기 위해서는 XML-RPC 를 사용한 응용 프로그램을 별도로 개발하여 사용해야 하는 데, 현재까지는 이러한 응용이 거의 없는 실정이다. 본 논문에서 제시하는 시스템은, 스마트폰이 가지고 있는 이동성과 소셜 네트워킹 서비스가 가지는 연결성을 적극적으로 활용하여 모바일 환경에 적합한 일정 통지 및 이슈 추적을 가능하게 한다.

III. 스마트 폰 상의 일정 통지 및 이슈 추적 시스템 (Schedule Notification and Issue Tracking System on Smartphone)

그림 7은 제안된 시스템의 구성도이다. 제안된 시스템은 크게 서버 시스템과 클라이언트 시스템으로 나뉜다. 서버 시스템은 기존의 소셜 네트워킹 서비스를 적극적으로 활용하여 프로젝트 개발자에게 프로젝트 작업의 데드라인에 대한 통지와 이슈 추적을 가능하게 한다. 스마트폰 상에서 개발된 클라이언트 시스템은 프로젝트 개발자에게 간트 차트를 디스플레이해 주어 프로젝트 개발 일정을 전체적으로 알아볼 수 있게 하고 자신의 일정을 입력 내지 수정할 수 있도록 한다. 프로젝트 개발자는 스마트폰 상에 디스플레이된 프로젝트 스케줄을 보고 스마트폰에서 수행되는 트위터를 통해 새로운 일정을 통지받고 이슈 추적을 수행한다. 개발자가 업데이트한 스케줄은 서버를 통해 스케줄 데이터베이스에 저장된다. 서버는 스케줄 중 개발자에게 통지해야 할 내용은 트위터를 통해 개발자의 스마트폰으로 알릴 수 있다.

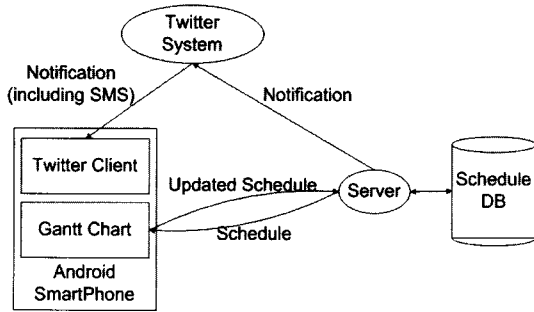


그림 7. 제안된 시스템의 전체 구성도
Fig. 7. Overall Diagram of the System

그림 7의 서브시스템들을 보면, 트위터 클라이언트는 기존의 안드로이드 상에서 작동하는 트위터 클라이언트들을 분석하여 설계하였다. 이를 위해 TwitterDroid¹⁾와 같은 프로그램들을 분석하였다. 최종적으로는 URI를 서버에 보내서 JavaScript Object Notation(JSON) 객체로 필요한 정보를 받아와서 해석하는 방법을 사용하였다. 트위터 클라이언트의 구현 의도는 업데이트 및 타임라인을 통해 새로운 일정을 통지받고 이슈를 추적하는 것이다. 또한 간트 차트를 그리는 Gantt Chart 렌더링 모듈은 안드로이드 SDK에서 지원하는 기본적인 2차원 그래픽 객체들을 이용하여 새로운 View를 설계하는 방법으로 개발하였다. 간트 차트의 기본적인 구현 의도는 전체적인 일정 정보를 스마트폰 상에서 조망하고 새로운 일정을 수정할 수 있게 하기 위한 것이다. 간트 차트를 그리기 위해 필요한 일정 데이터는 데스크탑 서버에서 eXtensible Markup Language - Simple Object Access Protocol(XML-SOAP)을 통해 가져오도록 하였으며, 트위터 클라이언트에서는 일정의 시작일 및 마감일과 현재 날짜들을 비교하여 트위터 상으로 일정을 공지하도록 하였다.

사용자가 안드로이드 스마트폰에서 본 시스템을 실행한 후 일정 관리 메뉴를 선택하면, 간트 차트를 렌더링해 주는 모듈이 Simple Object Access Protocol (SOAP)[9]을 사용하여 스케줄 데이터베이스가 있는 서버에 접속한 후 일정 정보를 가져와서, 안드로이드의 2차원 그래픽 서비스를 사용하여 그림 8과 같이 화면에 디스플레이 해준다. 일정 정보는 데이터베이스에서는 SOAP를 통

해 전달된 사용자의 ID를 검색 키로 검색되며, 표 1과 같은 작업 번호, 작업 이름, 작업 시작일, 작업 종료일 등의 정보로 구성되어 있다. 사용자가 수행할 프로젝트의 일정을 위한 데이터는 서버를 통해 SOAP 프로토콜을 통해 받아온다. SOAP은 HyperText Transfer Protocol(HTTP) 프로토콜을 통해 eXtensible Markup Language(XML) 형태의 텍스트를 데이터로 받는 프로토콜이다. 받아온 데이터는 XML 형태이므로 파서를 통해 파싱된다. 스케줄 데이터에 근거하여 클라이언트 프로그램을 간트 차트를 디스플레이하여 개발자가 프로젝트 수행 상황을 이해하는 데 도움을 준다.

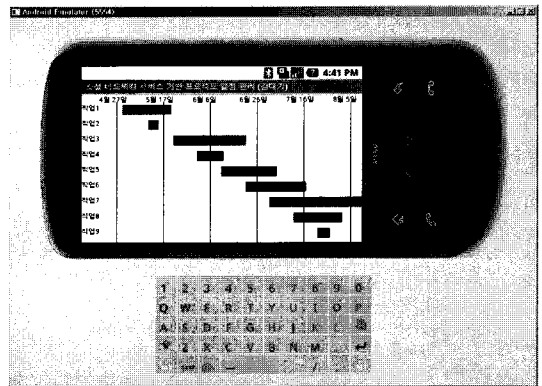


그림 8. 안드로이드에서 간트 차트 구성
Fig. 8. Gantt Chart on Android

사용자가 본 시스템에서 이슈 추적 메뉴를 선택하면, 트위터 클라이언트가 실행되어 Basic Authentication Scheme을 기반으로 공개 트위터 API를 사용하여 트위터 서버에 접속한 후, 타임라인을 읽어들이어서 그림 10과 같이 화면에 디스플레이해 준다. 이 타임라인은 전적으로 프로젝트에 관련된 일정 통지 및 이슈들로 구성되어 있다. 타임라인 및 업데이트 데이터를 요구하기 위한 공개 트위터 API는 HTTP 프로토콜에 맞춰진 URI의 형식으로 트위터 서버로 전달되며, 타임라인 데이터는 JavaScript Object Notation(JSON) 객체로 트위터 클라이언트 프로그램에 전달되며, 클라이언트 프로그램을 이 파싱하여 화면에 출력해야 할 정보를 추출하여 화면에 디스플레이한다. 이 때, 한글의 경우는 escape되어 전

1) <http://github.com/fbrunel/twitterdroid/>

달되므로 이를 unescape 해주어야 한다.

본 시스템의 특징은 그 실행 환경이 모바일 스마트폰이므로, 데스크탑 컴퓨터와 웹 상에서 작업 스케줄을 관리하도록 되어 있는 기존의 연구들과 차별됨을 알 수 있다. 또한 개발자들 간의 기본적인 통신을 소셜 네트워킹 서비스에 맡김으로써, 보안상 다소의 희생이 있을 수 있으나, 기본적인 개발 비용의 절감 및 신뢰성 있는 서비스를 수행할 수 있도록 해주는 것이다.

그림 8은 안드로이드 스마트폰 애플레이터에서 특정 개발자에 대해 간트 차트를 디스플레이한 예이다. 일반적으로 데스크탑 컴퓨터에서 보는 것과 동일하게 일정을 한 눈에 볼 수 있음을 알 수 있다. 이 클라이언트 프로그램은 구글 안드로이드 SDK 1.5를 사용하여 개발되었다. 안드로이드에서는 하나의 화면이 Activity 객체로 나타내지는 데, 그림 8과 같은 화면을 위해 안드로이드의 View 객체를 상속받아서 이러한 화면을 구현하기 위한 객체를 새로 만들었다. 앞에서 설명한 바와 같이 이 새로운 객체 내에서 서버로부터 가져온 일정 정보를 기반으로 간트 차트가 렌더된다.

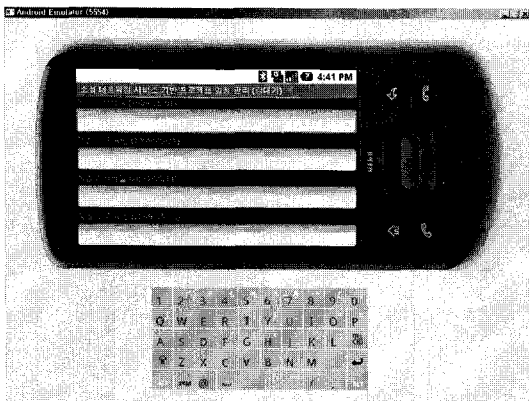


그림 9. 스마트폰에서 개발자가 일정을 입력하는 화면

Fig. 9. Developer's Schedule Input Screen on Smartphone

그림 9은 개발자가 스마트폰에서 자신의 일정을 입력하거나 수정하는 모습을 나타낸 그림이다. 개발자가 수정을 마치고 나서, 입력 버튼을 누르면 변경한 스케줄 내용은 SOAP 프로토콜을 통해 서버로 전달된다. 전달된 내용은 서버의 스케줄 데이터베이스에 저장되며, 나중

에 다시 간트 차트를 표시할 때 변경된 내용이 반영되어 차트가 디스플레이된다. 따라서 사용자는 이동 중에도 스마트폰을 통해 프로젝트의 일정을 한 눈에 조망하고 자신의 새로운 일정을 입력하거나 기존의 일정을 수정하거나 삭제할 수 있는 것이다.

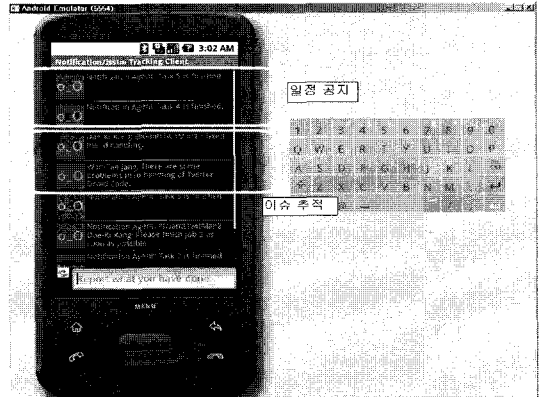


그림 10. 안드로이드의 클라이언트 상에 나타난 일정 공지와 이슈 추적의 예

Fig. 10. Example of Schedule Notification and Issue Tracking on Android Client

그림 10는 안드로이드 운영체제 상에서 본 연구에서 구현한 클라이언트 상에 나타난 일정 공지와 이슈 추적의 예이다. 본 일정 공지는 종료일이 다가오거나 지난 작업에 대해 시스템에서 개발자에게 속히 작업을 완료해 줄 것을 종용하는 업데이트이다. 이러한 공지는 웹으로도 볼 수 있으며, 앞에서 설명한 트위터의 기능 중 하인 무선 통지 기능을 통해 개발자가 무선 단말기기를 설정해 놓으면 SMS를 통한 문자 메시지로 개발자에게 통지된다.

또한 그림 10의 타임라인 (timeline)의 업데이트들을 보면 프로그램을 개발하면서, 버그에 대한 공지 및 설명, 논의, 해결책 등을 유지하고 기록하게 하기 위한 이슈 추적이 가능함을 알 수 있다.

이렇게 트위터와 같은 공개 SNS 서비스를 이용하여 일정 통지 및 이슈 추적을 구현한 의도는 공개 서비스를 사용함으로써 보안 문제는 다소 희생할 수 있으나 신뢰성 있는 지속적인 서비스 및 비용 절감의 효과를 얻을 수 있기 때문이다. 공개 SNS 서비스와 오픈 소스를 적극적으로 채택하여 일정 통지 서버 쪽의 별도의 구현이 없이

효율적으로 설계되어, 보안성이 크게 중요하지 않은 소규모 경량 프로젝트나 오픈 소스 프로젝트 개발에 활용 가능성이 높으리라 여겨진다.

제안된 시스템의 전체 설정에 대해 정리하면 다음과 같다. 서버와 클라이언트 프로그램은 자바 언어로 제작되었다. 기본적으로 서버는 PC 기반의 자바 어플리케이션이며 클라이언트는 구글 안드로이드 어플리케이션이다. 스마트폰의 GUI 프로그램 개발을 위해서 구글 안드로이드 SDK 1.5를 사용하였다. 사용자의 스마트폰에서 서버로의 통신은 Simple Object Access Protocol (SOAP)을 사용하였다. 서버의 데이터베이스는 MySQL Community Server 5.1을 사용하였다. 본 시스템에서 서버에서 트위터로, 그리고 트위터에서 클라이언트로의 트위터를 지원하는 부분은 기존의 TwitterDroid 를 분석한 후 새롭게 구현하여 사용하였다.

IV. 결론 (Conclusion)

4.1 요약 (Summary)

기존의 이슈 추적 시스템들은 데스크탑 컴퓨터와 웹상에서 작업 스케줄을 관리하고 개발 상의 이슈를 추적하는 데에만 특화되어 있고, 모바일 환경에 적합하지 않다. 이러한 문제점을 극복하기 위해, 본 논문에서는 스마트폰 상에서 프로젝트 관리를 위한 소셜 네트워킹 서비스 기반의 일정 통지 및 이슈 추적 시스템을 제안하고자 한다. 본 논문에서 제안하는 시스템은 모바일 환경에서 소셜 네트워킹 서비스를 적극적으로 활용하여 프로젝트의 스케줄 관리를 지원할 수 있도록 설계되었는데, 이러한 시스템은 본 논문에서 최초로 제안하는 독창적인 부분이라 할 수 있다.

제안된 시스템을 기술적으로 설명하자면, 서버 시스템과 클라이언트 시스템으로 나뉜다. 서버 시스템은 기존의 소셜 네트워킹 서비스를 적극적으로 활용하여 참여자에게 프로젝트 작업의 테드라인에 대한 통지와 이슈 추적을 가능하게 한다. 스마트폰 상에서 개발된 클라이언트 시스템은 프로젝트 참여자에게 간트 차트를 디스플레이해 주고, 자신의 일정을 입력 내지 수정할 수 있도록 한다.

제안된 시스템은 기존의 스마트폰이 가지고 있는 이동성과 소셜 네트워킹 서비스가 가지는 연결성을 적극적으로 활용하여, 프로젝트 개발에서의 일정 통지 및 이슈 추적에 이용하며, 이를 통해 소셜 네트워킹 서비스의 새로운 응용 가능성을 제시한다. 제안된 시스템은 프로젝트 개발에 있어 보안성이 크게 중요하지 않은 소규모 경량 프로젝트나 오픈 소스 프로젝트 개발 분야에서 저비용 일정 관리 및 통지 시스템으로 활용 가능성을 기대할 수 있다.

4.2 향후 연구 (Future Researches)

현재 트위터 관련 연구 및 개발은 말 그대로 열풍이라는 말이 무색할 정도로 활발하게 이루어지고 있다. 따라서 앞으로도 하루에도 수많은 트위터 관련 연구와 제품들이 쏟아질 것으로 예상된다. 본 연구에서 제안한 시스템은 그러한 연구들에 선도적인 제품이 될 것이며, 향후 다음 부분들에 대한 개선과 심화 연구를 계획하고 있다.

1. 지능형 증거 기반 일정 관리 (Intelligent Evidence Based Scheduling) - 프로젝트 개발 일정에서 예외 또는 개발 지연이 발생하는 경우, 프로젝트 책임자나 개발자가 제기하는 이슈를 소프트웨어가 자동으로 인식하여 일정을 조정하는 방안
2. 달력파 스케줄 입력 연동 - 안드로이드가 가지고 있는 달력 프로그램과 본 스케줄 시스템을 Intent Receiver 와 Broadcast Receiver 를 통해 확장하는 방안
3. 스마트폰의 통지 시스템과 연동 - 현재 대부분의 스마트폰은 자체적인 통지 시스템을 가지고 있는데, 이러한 통지 시스템과 연동하는 방안
4. 다양한 프로젝트 스케줄 차트 구현 - 간트 차트 외의 프로젝트 관리 및 시각화 방법들의 적용[9]
5. 트위터의 주제의 해명 기법과의 연결 - 소규모 경량 프로젝트를 위해 별도의 서버와 데이터베이스에 스케줄을 저장하는 것이 아니라 트위터 시스템에 전적으로 의존하게 하는 방안으로, 트위터에서 검색을 위한 해명(#project_X) 기법을 사용하여 스케줄 서버를 대신하는 방안

참고문헌

[1] J. Agüero, M. Rebollo, C. Carrascosa1 and V. Julián1, "Does Android Dream with Intelligent Agents?," International Symposium on Distributed Computing and Artificial Intelligence, pp. 194-204, 2008.

[2] R. Ballagas, J. Borchers, M. Rohs, and J. G. Sheridan, "The Smart Phone: A Ubiquitous Input Device," IEEE Pervasive Computing, vol. 5, no. 1, pp. 70-77, January-March, 2006.

[3] A. Java, X. Song, T. Finin and B. Tseng, "Why we twitter: understanding microblogging usage and communities," In Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis, pp. 56-65, 2007.

[4] Y. Ahn, S. Han, H. Kwak, S. Moon and H. Jeong, "Analysis of topological characteristics of huge online social networking services." In Proceedings of the 16th international Conference on World Wide Web, pp. 835-844, Banff, Alberta, Canada, May 08 - 12, 2007.

[5] J. Spolsky, More Joel on Software, APress, 2008.

[6] C. Chapman and S. Ward, Project risk management: processes, techniques and insights, Chichester, UK, John Wiley, 1996.

[7] W. Clark, F.W. Trubold, The Gantt chart: a working tool of management, The Ronald Press Company, 1922

[8] M. Fredericks and V. Basili, Using Defect Tracking and Analysis to Improve Software Quality, A DACS State-of-the-Art Report, November 1998..

[9] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Nielsen, S. Thatte, and D. Wine Simple Object Access Protocol (SOAP) 1.1, April 2007.

[10] H. Maylor, "Beyond the Gantt chart: - Project management moving on," European Management Journal, 19(1), pp. 92-100, February 2001.

저자소개

강대기(Dae-Ki Kang)



1992년 : 한양대학교 전자계산학과 (공학사)
 1994년 : 서강대학교 전자계산학과 (이학석사)

1994년~1999년 : 한국전자통신연구원 (연구원)
 2006년 : Iowa State University Dept. of Computer Science 졸업 (PhD in Computer Science)
 2007년 2월 ~ 2007년 8월 : 국가보안기술연구소 (선임연구원)
 2007년 9월 ~ 현재 : 동서대학교 컴퓨터정보공학부 조교수
 ※ 관심분야 : 기계학습, 관계학습, 통계적그래피컬모델, 스마트폰, 온톨로지학습, 침입탐지, 웹방화벽, 웹마이닝, 컴퓨터비전

장원태(Won-Tae Chang)



1989년 : 성균관대학교 전자공학과 (공학사)
 1996년 : 서울시립대학교 제어계측공학과 (공학석사)

1989년 ~ 2001 : Korea Telecom Authority International
 2002년 ~ 현재 : 동서대학교 컴퓨터정보공학부 교수
 ※ 관심분야 : 모바일S/W, RFID, Smart Phone, 컴퓨터 응용