

---

# 안드로이드상의 날씨 시각정보화 시스템 개발

황성문\* · 이효성\* · 박승현\* · 김주영\* · 김태석\*

\*동의대학교 컴퓨터소프트웨어공학과

## Development of a weather information visualization system on Android

Sung-mun Hwang\* · Hyo-sung Lee\* · Seung-hyun Park\* · Ju-young Kim\* · Tae-suk Kim\*

\*Dep. of Computer Software Engineering in Dong-eui University

E-mail : ffsm12@nate.com

### 요 약

본 연구에서는 안드로이드상의 날씨정보에 대해서 사용자가 보다 효과적으로 인식가능하도록 시각적인 표현을 제공하는 날씨정보시각화 시스템을 개발한다. 날씨정보 시각화는 날씨를 표현하는 기온, 바람, 비, 황사, 천둥번개, 일조량 등 다양한 요소들을 일차원적인 숫자표현이 아닌 2,3차원적인 시각적 효과를 느낄수 있는 미디어로 표현함으로 인해 사용자들이 쉽게 인지할 수 있는 정보표현법이다. 여기서 나타내고자 하는 날씨정보는 일반적인 날씨표현과 동일한 현재날씨와 주간날씨 그리고 특정 일자의 날씨등을 시각화하는 것을 대상으로하며 날씨에 관한 정보제공은 구글API를 이용하여 개발하고 안드로이드를 어떻게 분석하여 활용할 수 있는지에 관해서 검토하기로 한다.

### ABSTRACT

This study has been developed in order to provide a visualization of weather information system that users are capable of effective understanding on Android. The visualization of weather information is not expressed by number. It is the easiest way to express information with two or three dimensional of media based on the temperature, wind, rain, yellow dust, thunderstorm and amount of sunshine.

What here shows, is by targeting current weather, weekly-weather and particular day's weather which is exactly same as general weather expression and informed by Google API. Now, let's examine how to analyze Android in order to manage it. That users are capable of effective understanding of weather information on Android.

### 키워드

Android, 날씨, 시각화시스템, 스마트폰

### 1. 서 론

예전에 핸드폰이라고 하면 주로 전화를 목적으로 사용됐었다. 그러나 우리나라에서는 작년 12월 아이폰의 출시 이후로 스마트폰의 관심이 점점 높아지는 가운데 각 핸드폰제품회사에서는 자기들의 회사이름을 내건 여러 가지의 스마트폰들이

마구 쏟아져 나오고 있다. 회사들마다 스마트폰에 들어가는 O/S만해도 윈도우모바일(Windows Mobile), 안드로이드(Android), 아이폰OS(I-Phone OS), 심비안(Symbian) 등이 있다. 수많은 스마트폰O/S 중 안드로이드는 출시가 된지 2년정도 밖에 되지않아 아이폰OS나 윈도우모바일에 비해서 어플리케이션이 많지 않은 단점이 있다. 본 연구

에서는 이러한 단점을 조금이나마 보완하기 위해 실생활에도 유용한 날씨 시각정보화 시스템을 개발한다. 날씨 시각정보화는 구글API를 이용하고, 날씨정보를 1차원적인 숫자표현이 아닌 2,3차원적인 시각적 효과를 느낄 수 있도록 한다.

## II. 안드로이드상의 날씨 시각정보화

본 장에서는 안드로이드 어플리케이션구조, 시각정보화 시스템을 구동하게 될 기기, 시스템관련 기술을 좀 더 자세히 분석하였다.

### II-1. 안드로이드 어플리케이션구조

안드로이드 어플리케이션은 Activity, Intent Receiver, Service, Content Provider 이 4가지로 구성되어 있다. 모든 어플리케이션이 이 4가지를 다 필요로 하지 않으며, 이들의 조합으로 이루어진다. Activity는 어플리케이션에서 하나의 화면을 지칭한다. 각 Activity는 Activity base class를 상속하여 구현한 하나의 클래스이며, 사용자에게 View와 event 응답으로 이루어진 인터페이스를 화면상에 보여줄 것이다. 예를들어 텍스트 메시징 어플리케이션은 첫 화면에서 contact의 목록을 보여줄 것이고, 두번째 화면에서는 선택한 contact에게 메시지를 쓸 수 있도록 하고, 나머지 화면에서는 보낸 메시지를 확인하거나 환경설정을 바꾸는 화면이 될 것이다. 이러한 화면이 Activity가 되며, 다른 화면으로의 이동은 새로운 Activity를 시작하는 것과 같다. 어떤 상황에서는 Activity가 이전의 Activity로 값을 보내줄 수도 있다. 예를 들면 사용자가 사진을 선택했을 경우를 이전의 Activity에 선택한 사진을 보내주는 것과 같은 것이다.

화면이 오픈되면, 이전 화면은 멈추게 되며, history stack에 저장된다. 사용자는 history내에 있는 이전 화면으로 돌아갈 수 있다. 화면들은 history stack내에 저장될 필요가 없는 경우 삭제될 수 있으며, 안드로이드는 이 history stack을 home screen으로부터 실행시킨 각 어플리케이션마다 유지하게 된다.

IntentReceiver는 작성한 어플리케이션 코드내에서 핸드폰에 통화가 걸려오거나, 데이터 네트워크 접속이 활성화되는 것과 같은 외부 이벤트를 처리하는데 사용된다. IntentReceiver는 UI를 그려주는 것은 아니고, NotificationManager를 이용하여 사용자에게 어떤 일이 발생했다는 것을 알려준다. IntentReceiver도 AndroidManifest.xml 파일에 등록되지만, Context.registerReceiver()를 이용하여 코드상에서 등록해줄 수도 있다. 어플리케이션이 호출되어야하는 자신의 IntentReceiver에 의해 실행할 수 없을 때는 IntentReceiver가 트리거 되면 필요에 따라 시스템이 어플리케이션을 실행한다. 또한 어플리케이션은 Context.bradcastIntent()를

호출하여 자신의 Intent를 다른 어플리케이션에 broadcast할 수 있다.

Service는 UI와 상관없이 아주 오랫동안 존재하며 실행되는 코드이다. 예를 들면 재생목록에서 노래를 재생하는 미디어 플레이어 같은 것이다. 미디어 플레이어 어플리케이션은 사용자가 곡을 선택하고 재생을 시작하게 하는 하나 이상의 Activity를 가지고 있지만, 스스로 음악 재생을 하는 것은 Activity에 의해 실행되는 것은 아니다. 사용자는 새로운 화면으로 이동하고 나서도 음악을 계속 재생하기를 기대하기 때문이다. 이런 상황에서 미디어 플레이어의 Activity는 Context.startService()를 이용하면 Service로 실행된다. 시스템은 음악 재생 서비스를 멈출때까지 계속 재생할 것이다. Context.bindService() 메소드는 서비스에 연결하거나 아직 시작하지 않은 Service를 시작한다. Service에 연결이 되면 Service에 접근가능한 인터페이스를 통해 멈춤, 다시 재생등을 할 수 있도록 접근이 가능하다.

Content Provider는 어플리케이션은 자신의 데이터를 SQLite 데이터베이스 또는 다른 방법으로 파일에 저장할 수 있다. ContentProvider는 어플리케이션 데이터가 다른 어플리케이션과 공유할 필요가 있을 때 아주 유용하다. 이 클래스는 다른 어플리케이션이 데이터를 저장하거나, 가져오는 것과 같은 작업을 할 수 있도록 해준다.

어플리케이션에서 어떤 컴포넌트들을 사용할지 결정하였다면 이 컴포넌트들의 목록을 AndroidManifest.xml 파일에 기록해야 한다. 이 xml 파일은 어느 어플리케이션에서 이 컴포넌트들을 선언했는지, 그들의 기능과 요구사항은 무엇인지를 기록하는 파일이다.[1]

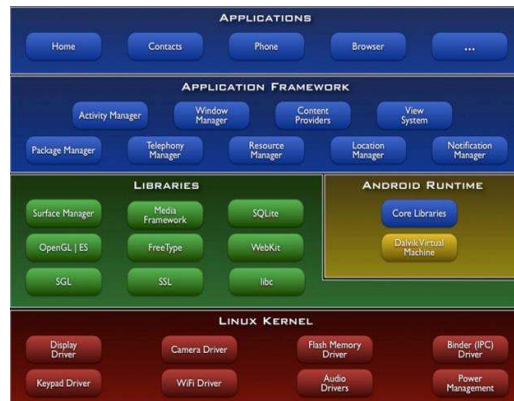


그림 1. 안드로이드 플랫폼 구조

위에서 설명한 어플리케이션구조는 그림1의 첫 번째 Application Layer에 해당한다.

### II-2. 시스템 및 관련 기술

이 시각정보화 시스템은 우리나라에서 최초로 안드로이드O/S를 탑재하고 2010년 2월 5일 출시한 그림2의 모토로이에서 구동될 예정이다. 모토

로이의 특징으로는 3.7인치 대형 디스플레이와 무선랜지원, 800만화소 카메라가 있다. 그리고 안드로이드O/S를 탑재하였기 때문에 구글 모바일 서비스를 사용할 수 있다.



그림 2. 모토로이[2]

그리고 구글API의 날씨주소는 "<http://www.google.com/ig/api?hl=ko&weather=도시명>"인데 인코딩형식을 utf-8로 바꿔줘야 한글로 된 결과를 받아올 수 있다. 그리고 api값을 사용하기 위해서는 파서를 이용해야하는데 파서의 종류로는 SAX, DOM, Pull Parser가 있다. 아래 표1은 각 파서들의 특징 및 장단점이다.

DOM Parser	특징 : Element를 트리구조로 사용 장점 : 한번 파싱하게되면 아무 때나 값을 얻을 수 있다 단점 : 메모리의 소모가 많음
SAX Parser	특징 : 이벤트 기반의 파서 장점 : 라인별로 파싱하기 때문에 일정한 메모리가 소요됨 단점 : 파싱시 그냥 지나갔던 Element의 정보를 얻으려면 다시 파싱해야함
Pull Parser	특징 : 이벤트 기반의 파서이지만 SAX와 달리 문서에 대한 모든 파싱하지않아도 특정 부분까지의 파싱내용을 활용할 수 있음 장점 : 원하는 부분까지만 파싱가능 단점 : SAX의 단점 + SAX보다 약간느림

표1. 파서들의 특징 및 장 단점[3]

시각정보화 시스템에서는 크기가 큰 XML파일을 이용하지 않기 때문에 3가지의 파서들중 SAX Parser를 이용하였다. 결과가 그림3의 1차원적인 파싱결과이고 다음 장에서 각각의 기온, 강우량, 눈, 바람을 2차원적인 시각적 효과로 설명 할 것이다.

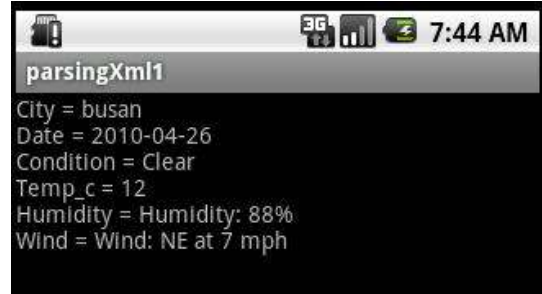


그림 3. 1차원적 결과

### III. 안드로이드상의 날씨 시각정보화 시스템 표현요소

날씨 시각정보화 시스템은 기온, 강우량, 눈, 바람, 구름을 일차원적인 숫자표현이 아닌 2차원적인 시각적 효과를 느낄수 있는 미디어로 표현함으로써 사용자들이 쉽게 인지할 수 있을 것으로 예상된다.

#### III-1. 기온



그림4. 더운 날의 시각적 효과

계절별 기온에 따라서 화면에 나타나는 캐릭터의 옷, 배경, 나무가 변화하도록 나타내었다. 더울 때는 나무도 푸르고 바닥에 잔디도 푸르게 했고, 캐릭터의 복장은 기온에 따라 5가지의 복장으로 나뉘었다. 그림4의 화면은 기온이 29도 일때의 옷차림이다.

#### III-2. 강우량



그림5. 비 내리는 날의 시각적 효과

강우량은 비가 내리는 비의 양에 따라서 화면에 내리는 비의 양을 조절하였는데, 비가 작게 내릴 경우 빗줄기가 조금 내리고 비가 많이 내릴 경우 빗줄기를 많이 내리게 하였다. 그림5의 화면은 강우량이 60mm 일때의 화면이다.

### III-3. 눈



그림6. 눈 내리는 날의 시각적 효과

눈 내리는 날에는 배경이 눈이 쌓인 모습으로 바뀌고 눈도 비와 마찬가지로 적설량에 따라서 눈의 양의 조절하여서 나타내었다. 그림6의 화면은 적설량이 6cm 일때의 화면이다.

### III-4. 바람



그림7. 강풍에 따른 시각적 효과

강풍은 바람의 세기를 얼마나 잘 표현하느냐가 중요하다. 그래서 배경에 있는 풀과 나무를 바람 세기에 따라서 바람이 세게 불어올 때는 세게 흔들리도록 표현했고 바람이 약하게 불어올때는 살랑살랑 거리도록 표현했다, 비나 눈이 올 경우에는 비와 눈 또한 바람세기에 따라서 날리도록 표현했다. 그림7은 바람의 세기가 40km/h 일때의

시각적 효과이다.

### III-5. 구름



그림8. 구름양에 따른 시각적 효과

구름양에 따른 시각적 효과는 맑음, 대체로 맑음, 흐림, 대체로 흐림, 구름 조금, 구름 많음으로 나누었는데 하늘색의 변화로써 시각적 효과를 나타내었다. 그림4의 하늘은 맑음 일때의 시각적 효과이고, 그림7과 그림8은 각각 구름많음과 흐림 상태의 시각적 효과이다.

## IV . 결 론

날씨 시각정보화 시스템은 날씨정보 시각화는 날씨를 표현하는 기온, 바람, 비, 황사, 천둥번개, 일조량 등 다양한 요소들을 일차원적인 숫자표현이 아닌 2,3차원적인 시각적 효과를 느낄수 있는 미디어로 표현함으로 인해 사용자들이 쉽게 인지할 수 있도록 제안했고, 이를 위해서 파싱된 결과값을 가지고 일차원의 결과가 아닌 이차원적인 이미지를 이용하여 보다 시각적인 효과를 높일 수 있었다.

## 참고문헌

- [1] 안드로이드사이드, 안드로이드 어플리케이션 구조, "http://androidside.com"
- [2] 모토로이, "http://www.motoroi.co.kr/"
- [3] SHANE CONDER, "Android wireless application development", 11. 19. 2009