

안드로이드 양방향 모바일 프레젠테이션 어플리케이션 개발

곽주형⁰, 박세영, 임채성, 한익주
한국산업기술대학교 컴퓨터공학과

e-mail : {elrun2⁰, amekist,hiphoppc}@naver.com, ijhan@kpu.ac.kr

Android Interactive Mobile Presentation

Kwark Ju-Hyung⁰, Park Se-Young, Lim Chae-sung, Han ik-joo
Dept. of Computer Engineering, Korea Polytechnic University

요 약

최근 스마트 기기의 보급화로 인해 많은 종류의 스마트기기용 어플리케이션이 개발되고 있다. 본 논문에서 다루는 안드로이드 OS(Android Operation System)를 사용한 양방향 모바일 프레젠테이션(Presentation)이란 기존 일방적 전달 방식의 프레젠테이션에서 모바일 플랫폼을 활용하여 청취자와 프레젠테이션 주최자간의 동일 자료를 통해, 제한적인 공간에서의 실시간적인 쌍방향 프레젠테이션 환경을 제공 해줄 수 있는 어플리케이션이다. 기존의 일방적인 방향으로 인해 생길 수 있는 청중들과의 커뮤니케이션의 개선과 발표 자료에 대한 가독성 저하의 문제를 해결하여 보다 발표자로 하여금 프레젠테이션의 효율을 높일 수 있도록 하였다.

1. 서 론

iOS, Android, Windows Mobile 등 2011년 현재 수많은 모바일 OS(Operation System)이 존재하며 많은 사람들이 스마트폰을 구매하여 이러한 운영체제들을 사용하고 있다. 그 때문에 우리 생활 속에서 스마트 기기는 보다 편리한 생활을 가능하도록 도와주고 있으며 그에 따른 각 OS에 맞는 어플리케이션 개발이 되고 있다.

사회적으로 스마트기기가 끼치는 영향을 굉장하며 스마트 기기라는 새로운 시장이 생성되어 점점 그 규모가 커져가고만 있다. [1]

그에 따라 많은 수의 사용자의 욕구를 충족시켜 줄 수 있는 참신한 아이디어의 어플리케이션 개발이 요구 되고 있다.

본 논문에서 다루고 있는 어플리케이션은 비슷한 기능을 보유한 어플리케이션들이 실제로 각 운영체제들이 운영하고 있는 마켓에서 제공되고 있다. 하지만 각 어플리케이션들이 제공하는 기능들은 일반적인 리모트 (Remote) 기능을 사용 하거나 컴퓨터 자체를 원격제어 하는 방식을 제공할 뿐 어떤 특정 기능에 특화 되어 있지 못하며 특화 되어 있다하여도 그 기능자체가 미비한 수준에 그친 경우가 많다. 어플리케이션 자체의 품질자체가 떨어지는 경우가 많으며 조작하는 인터페이스 또한 굉장히 불편한 경우가 많다.

실제로 간단한 리모트 어플리케이션은 프로그램 자체가 가벼워 네트워크 환경에 큰 영향을 받지 않으며 인터페이스 또한 단순하여 사용자에게는 편리 할 수 있다. 하지만 그 이외의 기능에 대한 추가 가능성을 기대하기가 힘들다.

반대로 원격제어 어플리케이션일 경우 컴퓨터 전체를 전부 제어하는 경우가 많이 때문에 속도에서 생각보다 느린 경우가 많으며 특히 인터페이스 굉장히 복잡하여 손바닥 크기에 스마트 기기 경우 사용이 어려운 점을 가지고 있다. 이런 경우 기능은

다양하나 프로그램이 무거워질 우려가 있으며 간단한 제어만이 필요할 경우에도 필요 이상의 자원을 요구 할 수 있는 상황이 생길 수 있다.

본 논문에서는 발표자의 화면과 청중들의 스마트기기의 화면을 동기화 시켜 같은 화면을 볼 수 있게 제공하며 간단한 리모트 기능 구현을 통해 발표자의 프레젠테이션을 돕고 부가적인 권한제어와 간단한 질의응답 기능 구현을 통해 보다 효율적인 프레젠테이션을 진행할 수 있도록 보조할 수 있는 무겁지 않게 필요한 기능들을 제공 할 수 있는 어플리케이션 개발을 목표로 한다.

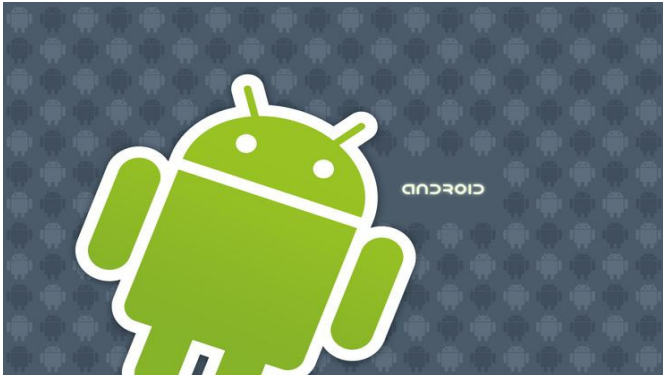
2. 관련 연구

2.1 TCP/IP 통신 (TCP/IP Communication)

컴퓨터 통신 분야에 네트워크 기술이 사용된 초창기에는 여러 종류의 네트워크가 산재 되어 사용 되었고, 네트워크의 종류마다 프로토콜(protocol, 통신규칙)도 서로 달랐다. 그래서 어떤 네트워크를 사용하느냐에 따라 통신 프로그램의 작성 방법도 달라야 했다. 하지만, 인터넷이 널리 사용되기 시작한 아래로 TCP/IP 프로토콜이 주류를 이루기 시작하였고, 그렇기 때문에 요즘은 네트워크 통신의 대개의 경우가 TCP/IP프로토콜로 통신을 한다.

TCP/IP 프로토콜은 TCP프로토콜과 IP 프로토콜을 함께 부르는 이름이다. 이 중 IP 프로토콜은 컴퓨터의 유일한 주소를 붙이는 규칙을 정의하는 프로토콜이다. IP프로토콜을 통해 정해진 주소를 이용하여 통신을 하게 된다. 이 규격화된 방식을 통해 안드로이드 기반의 기기와 기존의 PC사이의 통신이 가능하게 된다. [2]

2.2 안드로이드 OS (Android Operation System)



(그림 1) 안드로이드 OS

구글에서 개발한 ‘안드로이드(Android)’라는 모바일 단말기에 탑재되는 개방형 확장 플랫폼으로, 모바일 환경에서 개발자들이 자유로이 어플리케이션을 개발하여 탑재할 수 있도록 하는 통합적인 SW 스택이다. 안드로이드는 OS와 API, 미들웨어, 사용자 인터페이스, 브라우저에 이르는 모바일 기기에 들어가는 전반적인 SW환경을 포괄하는 것이다.

안드로이드의 가장 큰 장점은 오픈소스로 공개된 무료 플랫폼이라는 것이다. 폐쇄적인 경쟁 운영체제인 애플사의 iOS 와 가장 큰 다른 점이라고 할 수 있다. 그 때문에 개발자들로 하여금 다양한 개발환경을 제공하는 것이 가능해 졌으며 각 기업들이 안드로이드를 선호하는 이유가 되었다.

안드로이드는 기본 자바 언어를 사용하며 기존의 자바가 사용하는 JVM (Java Virtual Machine) 이 아닌 달빅이라는 구글이 개발한 독자적인 가상머신을 사용하여 차별화를 두었고 무엇보다도 자바언어를 활용한 만큼 기존 자바 개발자들에게 새로운 도전과제를 제시하는 데 성공 했다. 이젠 스마트 세상에 필요한 모든 기능을 제공하는 안드로이드는 개발자들에게 새로운 방향을 제시하고 있다. [3]

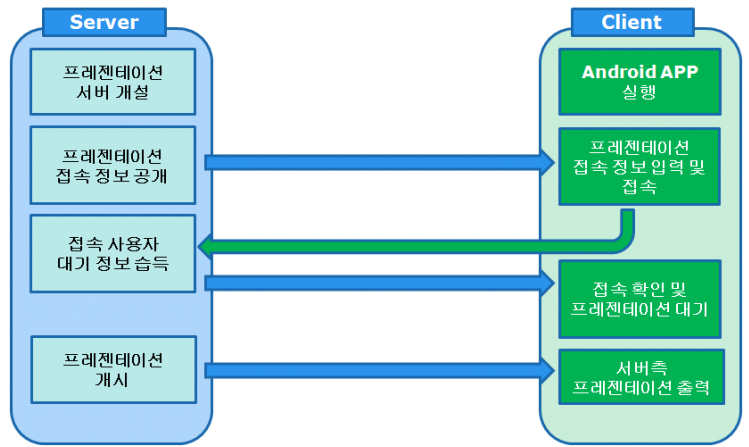


(그림 2) 안드로이드 아키텍처

2. 개발환경

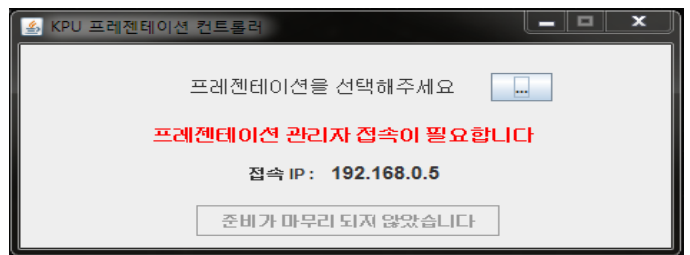
본 논문에서 제시한 소프트웨어는 Eclipse을 이용하여 전체 프레임과 네트워크를 구현 하였으며, 안드로이드의 내부 클래스 구현은 Android SDK 추가 이용하여 구현 하였다.[4]

그림 3.은 구현된 시스템 구성도를 도시화 하였다. 개발된 프로그램은 통합 개발 환경으로 구성하였다.



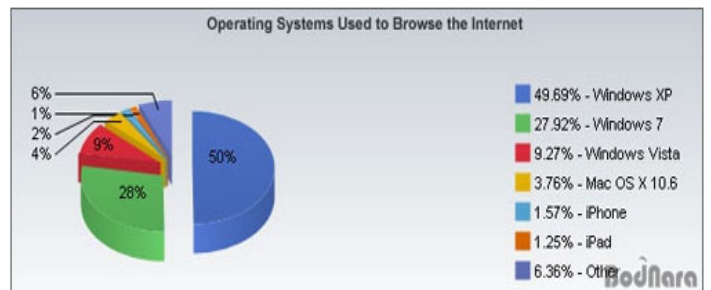
(그림 3) 전체 시스템 구성도

3. Java server using PC



(그림4-1) PC에서 실행된 Server

본 논문에서 사용한 소프트웨어의 서버 프로그램은 OracleTM 의 Java 프로그래밍 언어로 개발되었다. 이는 서버 프로그램이 구동되는 서버 시스템의 운영체제, 응용 프로그램, 의존 라이브러리에 관련된 서버 프로그램의 구동 문제에서 보다 자유롭게 어떤 환경의 서버 시스템에서나 구동되어 서버의 역할을 수행할 수 있도록 의도적 개발된 것이다.



(그림 4-2) 국내 OS 종류별 점유율[8]

그림 4-2와 같이 국내 OS 점유율은 Windows XP를 포함해, Windows 7, Linux, Mac OS 등의 다양한 분포를 보이며[8], 이는 결국 서버 프로그램이 가동될 많은 시스템이 다양한 운영체제에서 구동 가능해야한다는 결론을 얻게 된다. 따라서 개발

언어 선택에 있어서 C와 C++ 같은 Native Language와 WINAPI, MFC 등과 같은 특정 OS 의존적인 라이브러리는 사용할 수 없으며, Mac OS와 같은 운영체제 또한 지원의 가능성을 생각하여 .NET의 CLR 언어까지 제외를 하게 되면, 다수의 운영체제에서 안정적으로 돌아갈 수 있는 환경은 필연적으로 JAVA 언어 밖에 남지 않게 된다. 결과적으로 본 논문에서 사용한 서버 프로그램은 JAVA 언어로 개발되었기 때문에 다수의 OS 환경에서도 독립적으로 안정적인 구동을 보장한다는 결론을 내릴 수 있다.

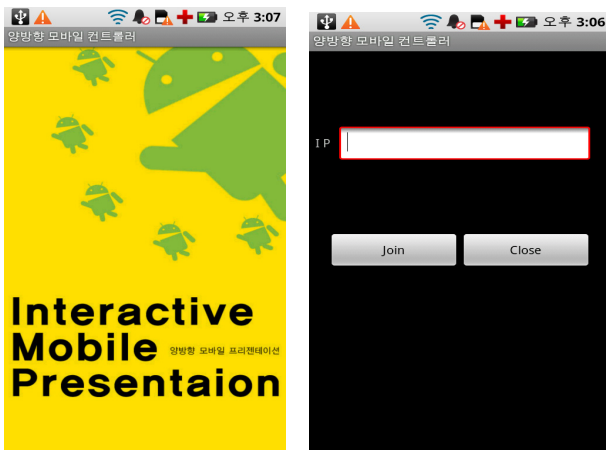


(그림 5) 안드로이드 화면 동기화

4. Android App

본 논문에서 사용한 안드로이드 OS의 버전은 이클레어 (2.1), 프로요(2.2),진저브레드 (2.3) 총 3가지의 버전에서 테스트를 하였다. 안드로이드의 문제점 중 하나인 각 버전별로의 호환성이 좋지 않기 때문에 각 버전별로테스트가 필요하다.[5]

구성되는 프레임 자체가 많지 않기 때문에 간단한 LinearLayout을 사용하여 구성하였으며 간단한 Touch Event 처리를 통해 화면전환과 내부 연산 및 네트워크 연동이 가능하다. 아래의 그림 4는 이러한 기법들이 사용된 모습을 보여주고 있다.[4]



(그림 4) 안드로이드 실행화면

4.1 Android network communication

기본적인 TCP/IP 방식을 사용하며 이 방식일 경우 생기는 안정성과 손실에 의한 정보 손상을 막는 효과가 있다.

그림 5를 살펴보면 실제 서버의 화면을 동기화 시킨 모습을 확인 할 수 있다. 이 과정은 서버에서 전송되는 이미지 패킷 (Image Packet)들을 바이트 단위로 처리하여 일정 공간의 버퍼에 저장하는 버퍼링방식을 사용하여 버퍼에 담겨있는 패킷들을 다시 안드로이드에서 제공하는 API를 통해 이미지로 재조립하여 화면에 출력하는 것이 가능하다.

안드로이드 상에서 그림5 에 보이는 각 버튼이벤트에 따라 서버 측으로 정보를 송신 할 경우도 정해진 패킷 설계 따라 송신하여 그에 따른 서비스를 요청하게 된다.[4]

4.2 Android UI & Thread

안드로이드 상에서 인터페이스를 구현하기 위해서는 먼저 UI 쓰레드를 사용해야만 한다. 일반적인 OS와는 달리 안드로이드는 운영체제상에서 어플리케이션이 동작하다 충돌하여 문제가 생기는 경우가 많으며, 그 중 개발 시 UI 쓰레드 충돌문제가 많이 생긴다. 기존의 화면에 대한 점유권을 가지고 있는 쓰레드를 다른 곳에서 무턱대고 점유하여 변경하여 할 경우 운영체제는 어플리케이션 보호를 위해 그 시도를 막게 되고 그 순간 어플리케이션에 대한 충돌이 일어나게 된다.

이런 경우를 막기 위하여 타이밍 혹은 핸들러를 이용한 제어 방법이 가능하다. 실제로 타이밍을 사용하는 것보다는 이벤트 핸들러를 통하여 점유권을 강제로 빼앗아 필요한 쓰레드가 사용할 수 있도록 하는 방법이 더 효과적이다. [4][5]

5. Network Packet Design

<표 1> 네트워크 제어패킷의 재인

네트워크 패킷 구성				
컨트롤 패킷의 정의 (총 21바이트)				
패킷 종류	서비스 사이즈	서비스에 사용되는 제어 패킷 내용		
1Byte	4byte	? byte	...	Null
서버와 클라이언트는 21바이트로 정의되어 있는 패킷을 프레젠테이션 제어 목적으로 사용하며, 패킷에 대한 분류는 최상위 1바이트를 통해 구분하며, 제어 패킷이 전송되는 포트는 20000이며, 포트 20001과 20002번은 이미지와 같은 가변 데이터 송수신에 사용함				

패킷 종류 구성			
패킷 번호	용도	패킷 번호	용도
1	유저 정보	7	유저 번호 할당
2	펜 드로우 정보	10	컬처 이미지
3	X	11	펜 드로우 종료
4	모바일기기 사이즈 정보	12-13	화면 이동
5	X	15	질의 송신 요청
6	질의 수령	18	프레젠테이션 권한 대외요청

본 논문에서 사용되는 네트워크 프로토콜은 TCP/IP 프로토콜이며, 서버와 클라이언트간의 상호 교환되는 패킷은 제어 패킷과 데이터 패킷의 두 종류로 나뉘게 된다. 제어 패킷은 통상 시에 클라이언트의 프레젠테이션 제어 정보가 즉시 서버로 송신되어 서버가 응답을 하게 되는 제어 명령이며 해당 제어 명령을 기본적으로 서버의 명령이 수행되도록 하되, 전체 21바이트의 최초 1바이트는 패킷의 종류가 후위 4바이트는 서비스 종류에 따른 데이터의 사이즈, 그 후위에는 서비스의 내용에 관한 제어 정보가 삽입되는 구조로 이루어지게 된다. 제어정보에 따라 데이터 전송이 이루어지는 소켓이 별도로 열리게 되며,

해당 소켓으로 이미지 데이터와 같은 대용량 데이터가 전송되게 된다. 제어 패킷에는 따로 체크섬과 같은 불완전 패킷에 대한 예외처리가 이루어지지 않는데, 이는 체크섬 대신 패킷의 종류와 전체 21바이트와 최후의 Null이라는 고정된 톨의 존재로 체크 섬의 역할을 대신하고 있기 때문이다. 가령, 정상적인 패킷의 경우 항상 패킷의 종류는 기존의 명시된 값이 나와야 하고 최후 21번째 1바이트는 반드시 Null이어야 하는 규칙에 만족하지 못한 패킷은 모두 버려지게 되어 손실 및 불완전 패킷을 수신하여 서버에 의도하지 않은 응답을 일으키는 현상을 방지한다. 이러한 방식은 고정된 크기와 구조를 갖는 방식에 한해 별도의 연산과정을 수행하는 체크섬 방식에 비해 오버헤드가 적다.

Complete guide , 한빛 미디어 , 김상형 저

[5]Andriod site

<http://www.android.com/>

[6] 안드로이드 베이직&클래식 안드로이드 입문서 3rd Edition

[7] Modern C++ Design, 안드레 일렉산드레스쿠 저, 이기형 역, 광용재 감수, 인포북

[8] 국내 하드웨어 정보지 : 보드나라

<http://www.bodnara.co.kr>

6. Singleton Pattern

내부적으로 데이터를 처리하는 방식은 싱글톤 패턴(Singleton Pattern)이라는 디자인 패턴을 사용하였다.

싱글톤 패턴은 Java를 사용하면 가장 많이 사용 하는 패턴이다. 이 패턴은 정적인 클래스를 만들어 호출시 마다 새로운 인스턴스가 생성되어 여러 개의 인스턴스가 생성되는 것이 아니라 처음 호출시 생성된 인스턴스가 사라지지 않고 유지가 되어 변수 및 메서드들의 데이터 값을 계속 유지시켜주어 그 값들을 활용 할 수 있게 해주는 패턴이다. [7]

Java로 만들어진 서버 프로그램에서도 이 패턴을 사용하여 네트워크 연결 및 송수신과 프레젠테이션을 사용하기위한 정보처리를 보다 효율적으로 사용, 관리하는 것이 가능하다. 서버에서 사용한 것처럼 안드로이드도 기본 사용언어가 java 이기 때문에 이 패턴의 적용이 가능하다.

7. 결론 및 고찰

본 논문에서는 프레젠테이션의 효율을 높이기 위한 어플리케이션을 구현하였다 이 과정에서 안드로이드와 기존의 PC와의 정보를 네트워크를 통해 서로 통신이 가능 하게 하여 양방향간의 소통이 가능하게 하였고, 안드로이드라는 스마트 운영체제를 사용하여 보다 스마트 기기 사용자들로 하여금 편리성을 제공할 수 있도록 쉽게 사용이 가능한 인터페이스를 만들어 편리성을 부각 시켰다. 이러한 어플리케이션을 통해 프레젠테이션을 진행하며 보다 청중들에게 보다 높은 가독성과 집중력을 제공하는데 도움을 줄수 있을 것이라 생각된다. 또한 간단한 채팅 기능과 권한 이입의 기능을 추가하여 보다 효율적인 프레젠테이션 환경을 조성할 수 있을 것이다.

참고문헌

[1] 김영기,“스마트폰 기기의 특징과 기술동향“,전자정보센터(EIC) 산업동향분석, 2010.12.9

[2] 너를 자극하는 Java 프로그래밍, 한빛미디어, 김윤명 저

[3] 구글폰의 실체와 대응방향 , 정책연구센터 , 정재호, 2007.11.30

[4] 안드로이드 프로그래밍정복 Android Programming