

주 제

무선 인터넷 단말 플랫폼의 구조와 역할 및 현황

SK Telecom 이은복, 임종태, 김종배, 김후종

차 례

I. 서론

II. T-PAK 플랫폼의 개요

III. 각 세부 모듈 기능

IV. 모듈 업그레이드 기능

V. 결론

요 약

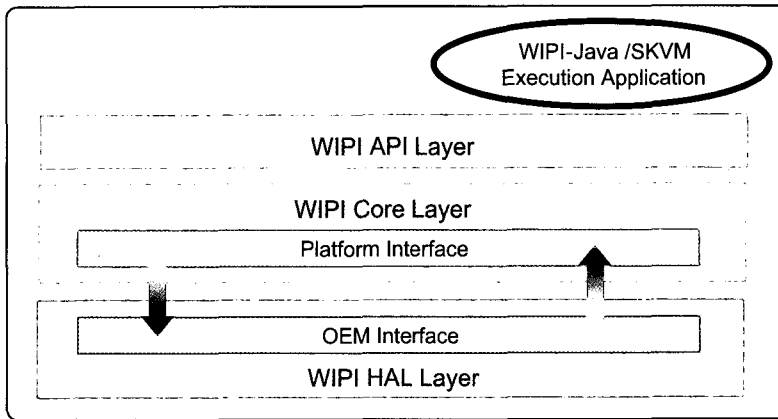
무선 인터넷 서비스가 다양화됨에 따라 핸드폰에 올라가는 시스템 소프트웨어들은 매우 복잡해지고 이에 따라 출시되는 시간도 오래 걸리게 되었다. 경쟁적인 휴대폰 시장에서는 새로운 서비스가 올라간 새 휴대폰을 빨리 시장에 출시하는 것이 중요하다. 이에 따라 휴대폰 개발사와 통신사는 어플리케이션과 운영 체제 사이의 통일된 실행 환경으로 모바일 핸드폰 플랫폼을 채택하려 한다.

본문에서는 한국의 최초 무선 인터넷 서비스 회사인 SK 텔레콤이 개발한 핸드폰 플랫폼인 T-PAK에 대해서 설명하도록 한다. T-PAK은 여러 어플리케이션과 서비스를 지원하기 위해 C와 C++, 그리고 J2ME MIDP까지 지원하는 일관된 API 집합들을 제공하고 있다. 이들 API 집합들은 전화 기능과 단말 리소스 관리 기능, 그리고 유연한 UI 기능을 포

함하고 있으며 이들을 이용하여 T-PAK이 핸드폰의 모든 기능을 가지게 하는 것이 가능했다. 또한 T-PAK에서는 이들 API 집합들을 구분된 DLL 형태로 가지고 있어서 내부에서 지원하는 동적 모듈 지원 기능을 이용하여 전파를 통하거나 케이블을 통해서 각자 확장하거나 업데이트 하는 것이 가능하다.

I. 서론

핸드폰이 처음 세상에 나왔을 때 기본 기능이었던 통화와 단문 메시지(SMS) 기능에 주력했던 것에 비해 시간이 지남에 따라 각 통신사들의 차별화된 부가 서비스가 제공되면서 많은 기능들이 추가되어 왔다. 그러나 이러한 다양화된 서비스의 기능 추가는 핸드폰 소프트웨어의 복잡성을 높였으며 이러한 복잡도가 핸드폰의 출시를 늦어지게 하는 요인으로 작용하



(그림 1) 기존의 WIPI 플랫폼 구조

기도 했다. 따라서 이러한 다양한 서비스들을 빠른 시간 안에 수용하기 위해서 이동 통신사들 마다 플랫폼의 개념을 도입하게 되었고 이 플랫폼을 이용하여 서비스의 개발을 편리하게 하고 단말에 적용하는 시간을 줄이며 서비스를 빠른 시간 안에 제공할 수 있었다.

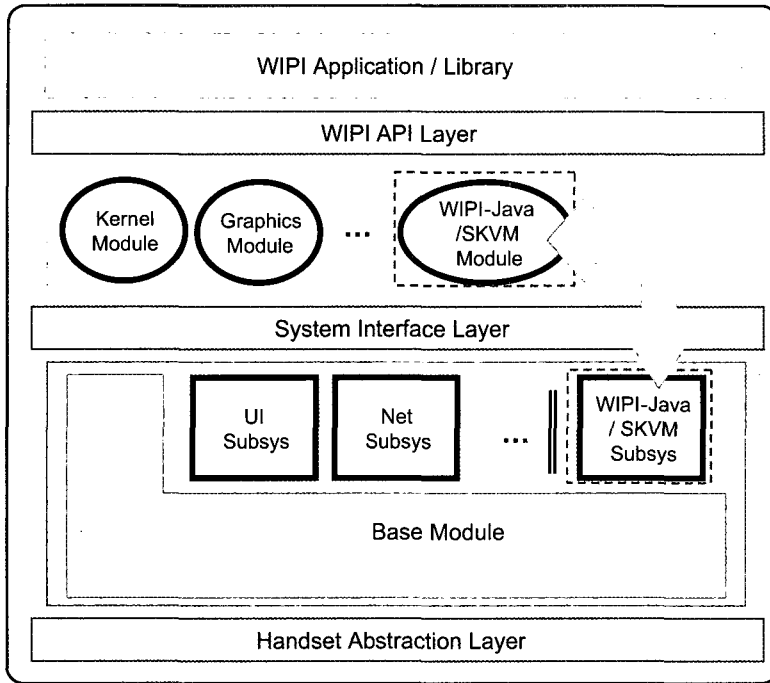
처음 플랫폼의 개념이 도입되던 때에는 각 이동 통신사 별로 독자적인 플랫폼을 가지고 서비스를 시작했으나 시장의 요구에 따라 이동 통신사 사이의 호환성이 요청됨으로 정부 주도로 Wireless Internet Platform for Interoperability (WIPI) 라 불리는 표준화 플랫폼을 개발하게 되었다.[1] WIPI 플랫폼에서는 다양한 핸드폰의 기능들을 접근할 수 있는 API 들을 제공함으로써 기존의 어플리케이션들이 게임과 같은 제한된 기능만을 제공했던 것에 비해 전화 기능이나 주소록 기능 등 어플리케이션이 커버할 수 있는 영역을 넓혀나갔다.

초기에 개발된 WIPI 플랫폼의 경우는 이러한 다양한 API 집합들을 하나의 단위 안에서 처리하려고 했기 때문에 API 의 확장이나 업데이트를 하기 위해서는 전체 WIPI 플랫폼을 교체해야 하는 문제가 있었다. SK 텔레콤에서 개발한 T-PAK 에서는 이러한 문제점을 해결하기 위해서 WIPI API 집합들을

기능에 따라 모듈화를 시켜서 서로 의존성이 없는 경우 별도로 업데이트나 확장이 가능하도록 만들었다. 본문의 나머지는 다음과 같이 구성되어 있다. II장에서는 전체적인 T-PAK 플랫폼의 개요를 살펴보고 III장에서는 각 API 집합들이 어떻게 구성되어 있는지 살펴보고 이들의 기능을 살펴보고자 한다. IV장에서는 모듈별 동적 업데이트 기능에 대해서 살펴보고 V장에서 정리하도록 한다.

II. T-PAK 플랫폼의 개요

T-PAK 은 SKT에서 자사의 여러 가지 서비스들을 효율적으로 제공하기 위해서 기존에 단말 제조사들이 차지했던 OEM 영역을 플랫폼 내부로 끌어들이 플랫폼만 포팅이 되면 서비스 제공이 자연스럽게 이루어지도록 구성된 통합 플랫폼이다. 또한 T-PAK 에서는 내부적으로 User Interface (UI)를 제공하기 위한 framework를 제공함으로써 고객에게는 서비스의 차별화를 통해 만족도를 증가시키고, 솔루션 개발사에게는 개발 편의성과 효율성을 제공함으로써 서비스의 개발 경쟁력을 강화할 수 있도록 하고, 단말



(그림 2) 모듈화된 WIPI 플랫폼

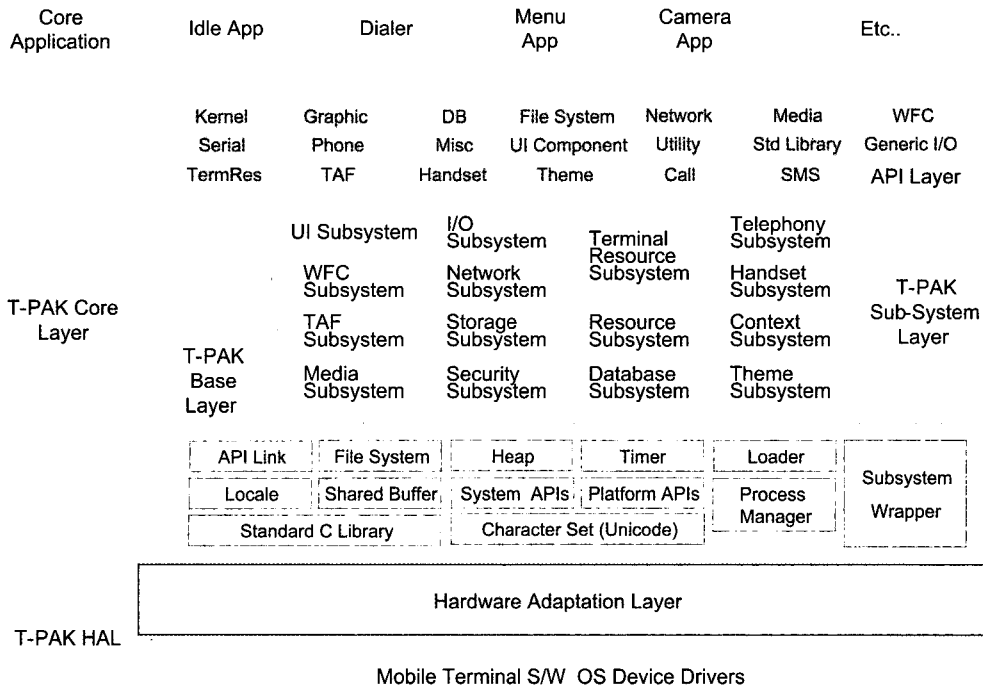
제조사에게는 단말 개발 비용을 획기적으로 절감할 수 있게 함으로써 단말의 제품 경쟁력을 강화할 수 있도록 한다.

T-PAK 의 기본이 되는 플랫폼은 기존의 WIPI 플랫폼을 보완 확장시킨 플랫폼이다. 기존의 WIPI 플랫폼과 가장 크게 달라진 점은 바로 각 API 집합들을 모듈화하여 확장성을 높인 점이다. 기존의 WIPI 플랫폼은 (그림 1)과 같은 구조를 가지고 있다. 전체적인 플랫폼은 하나의 형태로 존재하며 모든 API 들은 서로 상호 의존적인 관계를 가지고 있어서 부분을 구분하기 힘든 구조를 가지고 있었다. 이와 같은 구조에서는 플랫폼의 어떤 API 내부의 버그가 나중에 발견되어 이를 수정하려고 하거나 추가된 디바이스를 지원하려고 한다면 전체 플랫폼을 다 교체해 해야 한다.

이와 같은 기존의 WIPI 플랫폼의 단점을 보완하

기 위해서 서로 의존할 필요가 없는 부분을 찾아내고 그 부분들을 모듈화하여 별도로 업그레이드나 교체가 가능하도록 만드는 작업이 필요하다. 이렇게 바뀌어진 플랫폼의 모듈은 (그림 2)와 같다. 기존의 WIPI 플랫폼이 WIPI HAL Layer를 통해서 OEM 단과 인터페이스를 가지고 있는 반면에 모듈화된 WIPI 플랫폼의 경우에는 중간에 Sub-System Layer를 추가하여 하위로는 Handset Abstraction Layer 로 상위로는 System Interface Layer 로 상호 모듈사이에 연결을 하고 있다. 중간에 들어가는 Sub-System Layer 는 각 모듈들과 어쩔 수 없이 모두 상호 의존성을 가지는 Base 모듈과 나머지 Sub-System 모듈들로 구성되어 있다.

이와 같은 구조가 가지는 장점은 크게 두가지로 나누어 볼 수 있다. 첫 번째 장점은 플랫폼 자체가 핸드폰 환경이 아닌 다른 환경 즉 예를 들면 WinCE 나



(그림 3) T-PAK Architecture

Embedded Linux 같은 환경으로 포팅되는 경우 일부 Sub-System 의 모듈들은 하부 운영 체제가 지원하는 기능을 활용하여 구현할 수 있게 된다. Sub-System Layer 가 하부 운영 체제의 도움을 받도록 구현이 변경된다 하더라도 WIPI API Layer 의 구현은 System Interface Layer를 통해서 이루어지기 때문에 기존 WIPI API 가 변경되지 않으며 따라서 이 API 를 사용하는 어플리케이션들을 그대로 활용할 수 있다는 장점이 있다. 다른 하나의 장점은 Sub-System Layer 의 각 모듈들은 Base 모듈을 제외하면 상호 의존성이 없기 때문에 모듈 별로 교체가 가능하기 때문에 이후에 어떤 버그가 발견되거나 기능을 추가할 일이 발생하더라도 전체 시스템을 교체하지 않고 해당되는 모듈만 교체하면 된다는 장점이 있다.

III. 각 세부 모듈 기능

여기에서는 T-PAK 의 전체 구조 중에서 Sub-System Layer 에 해당하는 각 구성 요소들을 좀 더 자세히 살펴보도록 한다. (그림 3)은 T-PAK의 전체 구조를 보이고 있다. 여기서 유념할 점은 이 구조는 (그림 2)에서 설명한 모듈화된 WIPI 플랫폼에 기반하였다는 점이다. T-PAK 에서 Sub-System Layer 에서는 각 WIPI API 집합들을 구현하는 단위 모듈들로 구성되어 있으며 이들 모듈들은 서로 분리가 가능한 논리적 단위로 독립적인 개발이 가능하다. 이렇게 독립적인 개발이 가능하기 때문에 각 모듈은 서로 다른 개발사가 개발할 수도 있으며 제조사는 필요한 모듈을 선택적으로 수용할 수도 있다. Sub-

System Layer 를 구성하는 모듈들은 크게 다음의 두 가지로 나뉘어 진다.

- Base 모듈 : WIPI 플랫폼에서 가장 기본이 되는 필수 모듈을 추출하여 하나로 구성된 모듈이다. 이 모듈에서 정의하고 있는 것들은 WIPI 플랫폼과는 무관한 다른 시스템으로 연결될 수 있다. 즉 만약에 T-PAK 이 WIPI HAL 이 아닌 WinCE 나 Embedded Linux 와 같은 일반적인 운영 체제 위에서 구현된다고 할 때 이곳에서 정의하고 있는 API 들을 하위 운영 체제가 제공하는 API를 이용해서 구현이 가능하다. 이 Base 모듈에서 제공하는 API 들에는 메모리 관리자, 파일 시스템, 타이머, 프로세스 관리자, 로더, 로케일 관리자, 표준 C 라이브러리, 유니 코드 지원 함수, 시스템 API 등이 포함되어 있다.
- Sub-System 모듈 : Base 모듈 위에서 구현되며 각 해당되는 부분의 WIPI API를 구현하고 있다. 총 9가지 Sub-System 으로 구성되어 있으며 이들은 각각 I/O, Media, UI, Storage, Terminal Resource, Telephony, Network, Security, Handset 으로 이루어져 있다.

전체적으로 보았을 때 1개의 Base 모듈 과 다수의 세부 Sub-System 모듈들로 구성되며 각 모듈은 DSL (Dynamic System Library) 형태로 구현되어 있다. 각 Sub-System 모듈 들은 별개의 모듈들로 구분되어 있기 때문에 동적으로 추가나 업그레이드가 자유롭다.

1. Base 모듈

Base 모듈은 필수 모듈로서 다른 모듈과 종속 관계가 너무 심해서 분리하기 힘든 부분을 포함하고 있다. 이 모듈이 가지고 있는 기능은 다음과 같다.

(1) 플랫폼 관리

플랫폼 관리 부분에서는 플랫폼 자체를 초기화 하고 종료하는 API 들을 포함하고 있다. 또한 외부에서 이벤트도 여기서 제공하는 API를 통해서 받도록 하고 있다. 다른 API 들은 플랫폼 내부에서 불리는 API 들이지만 여기에 정의된 API 들은 플랫폼 외부에서 불러주는 API 이다.

(2) 프로세스 관리

T-PAK 에서는 어플리케이션이 실행되면 그에 매핑 되는 하나의 프로세스가 생성된다. 이들 프로세스는 Base 모듈의 프로세스 관리 API 에 의해서 관리되며 기본적인 기능으로는 프로세스를 생성, 종료 하는 기능과 상태 변경, 프로세스 정보 관리 등을 담당하고 있다.

(3) 이벤트 전달

이벤트 전달 API 들은 외부에서 플랫폼으로 전달 된 이벤트를 이를 처리할 프로세스에 전달하는 기능을 담당하고 있다. 외부에서 볼 때는 플랫폼에 이벤트를 전달하는 것이지만 내부적으로는 이벤트의 종류에 따라서는 특정 프로세스가 처리해야 하는 이벤트가 있기도 하고 현재 화면에 보이는 프로세스가 처리해야 하는 이벤트가 있기도 하기 때문에 적절한 이벤트 분배 과정을 거쳐서 해당 프로세스에게 이벤트가 전달된다.

(4) 동기화 처리

플랫폼이 실행됨에 따라 외부 운영 체제에서 플랫폼의 API 들을 여러 태스크에서 동시에 호출될 가능성이 있다. 이러한 경우 플랫폼이 자체적으로 내부 데이터의 보존을 위해서 동기화를 시켜줄 필요가 있는데, 동시에 접근 가능한 부분에 대해서는 상호 배제가 가능하도록 API를 제공한다. 가장 대표적인 경우가

외부 이벤트를 전달할 때 여러 곳에서 이벤트 큐를 제어하지 못하도록 동기화 API를 통해서 보호하고 있다.

(5) 메모리 관리

플랫폼에 제공되는 메모리는 전체 큰 덩어리로 제공되지만 실제로 내부적으로는 이 덩어리를 나누어서 효율적으로 사용해야 한다. 따라서 다른 Sub-System에서 메모리를 할당할 수 있도록 메모리 관리자가 구현이 되어 있다. 여기에서 제공하는 기능은 메모리의 할당, 해제 기능을 제공한다.

(6) 로더 기능

어떤 라이브러리나 어플리케이션을 사용 가능한 형태로 만들기 위해서 플랫폼 내부로 로딩을 해야 한다. 이렇게 어플리케이션이나 라이브러리를 로딩하고 관리해주는 API를 제공한다.

(7) 파일 시스템 관리

라이브러리나 어플리케이션은 모듈 형태로 존재하며 실제로는 파일 형태로 시스템에 저장되어 있다. 따라서 이들 모듈을 읽어서 사용 가능한 형태로 만들기 위해서는 파일에 접근할 수 있는 API가 필요하다. 따라서 Base 모듈 내부에서 파일 시스템에 관련된 API를 포함하고 있으며 여기서 제공되는 기능은 파일을 열고 닫고 읽고 쓰는 기능들을 제공한다.

(8) 타이머 기능

외부에서 플랫폼에 제공하는 타이머는 WIPI Handset Abstraction Layer를 통해서 하나 밖에 제공되지 않는다. 따라서 이를 플랫폼 내부에서 나누어 사용할 수 있도록 Base 모듈에서 기능을 제공한다.

(9) 기타 유틸리티 기능

Base 모듈 내부에서 사용하는 기타 기능들을 제공

한다. 여기 해당하는 것으로는 프로세스간 공유 메모리 관리 기능과 디버깅을 위한 로그 출력 기능, 현재 시간을 알려주는 기능, 로컬 코드와 유니 코드 간의 코드 변환 기능 등을 포함한다.

2. Storage 모듈

Storage 모듈은 어플리케이션의 정보를 단말기의 Persistent 영역에 저장하는 역할을 담당한다. 상위 어플리케이션에서는 파일 시스템에 직접 접근할 수 있는 API를 제공하고 있지 않으며 프로그램이 종료된 이후에도 저장하고 싶은 데이터는 이 Storage 모듈에서 제공하는 DataBase 와 Registry API를 통해서 저장해야 한다. 다른 Sub-System 모듈들도 마찬가지지만 여기서 제공하는 API는 어떻게 구현하는가에 따라 핸드폰 별로 다른 공간에 저장할 수도 있다. 즉, 구현에 따라서는 핸드폰 내부의 파일 시스템에 저장할 수도 있으며 어떤 경우에는 SIM 카드와 같은 외부 저장 장치에 저장할 수도 있을 것이다.

(1) Registry

Registry는 어플리케이션에 관련된 정보를 플랫폼에 저장해 놓는 역할을 한다. 쉽게 생각하면 윈도우즈에서 제공하는 registry와 같은 기능을 제공한다고 보면 된다. 여기서 많은 정보를 저장할 수 있는 기능 보다는 간단하게 key 와 value 의 pair로 존재하는 정보들을 저장할 수 있는 기능을 제공한다.

(2) DataBase

DataBase (DB)는 Registry 보다는 좀 더 많은 양의 정보를 저장하는데 사용된다. DB 정보에서는 복수개의 필드가 조합되어 하나의 레코드를 구성하도록 되어 있다. DB 생성시에 필드의 종류를 지정하고 조합하는 방법을 정의할 수 있다. 또한 여러 프로

그램이 하나의 DB 에 동시에 접근하는 경우를 대비하여 동기화를 제공하고 있으며 갑자기 전원이 나가는 경우에도 데이터를 잃어버리지 않도록 설계되어 있다.

3. UI 모듈

UI 모듈에서는 주로 LCD 화면을 제어하는 기능을 담당하고 있다. 가장 하위 단계의 기능인 프레임 버퍼에 관련된 작업부터 시작하여 기본적인 그리기 기능과 상위 단계 에서 요구하는 Component 기능까지 담당하고 있다.

(1) 프레임 버퍼 관련

먼저 가장 간단한 기능인 프레임 버퍼를 제어할 수 있는 기능을 제공한다. 프레임 버퍼는 LCD 화면에 보이는 raw 데이터를 메모리 형태로 제공한다. 여기에 실제 그리기를 한 후에 flush를 하면 raw 데이터가 LCD 에 보이게 된다.

(2) 그리기 기능

하위 단계에서 그리기 기능을 제공한다. 제공하는 기능은 점을 찍는 기능부터 시작해서 선 그리기, 원 그리기 및 arc 나 polygon 등까지 간단한 도형을 그릴 수 있는 기능을 제공한다.

(3) 이미지 처리

이미지를 생성하고 이를 화면에 보여줄 수 있는 기능을 제공한다. 화면에 보이고 있는 것을 이미지로 만들 수도 있고 T-PAK 플랫폼이 해석할 수 있는 포맷으로 된 데이터에서 이미지를 만들어 낼 수도 있다. T-PAK에서 지원되는 이미지 포맷은 wbmp, jpg, bmp, png, abmp, sis 등의 포맷을 지원한다.

(4) 폰트 지원

글자를 화면에 그리기 위한 폰트 데이터 등을 접근할 수 있는 API를 제공한다.

(5) 입력기 지원

핸드폰의 입력 방식을 반영할 수 있도록 automata를 제공한다. 입력한 키 값에 따라 어떤 글자로 조합되는지 알 수 있다.

(6) UI Component

간단한 레벨의 UI Component를 제공한다. 여기에는 메시지 박스와 프로그레스 바, 안테나 영역 등이 포함된다.

4. I/O 모듈

I/O 모듈에서는 Serial 포트 등을 통한 하드웨어와 통신을 담당하고 있다. 여기에서 다루고 있는 API 는 일반적인 통신을 다룰 수 있는 Generic I/O, Serial 포트를 통한 통신, 그밖에 GPS 나 LED 등의 하드웨어도 이 모듈의 API를 통해서 통제가 가능하다.

(1) Generic I/O

Generic I/O 는 일반적인 입출력 장치를 제어하기 위해서 만들어진 API 이다. I/O 모듈에서는 이러한 장치들을 관리하고 입력 출력 API를 매핑할 수 있도록 API를 제공하고 있다.

(2) Serial

Serial 은 단말기의 직렬 포트를 이용해서 통신을 할 수 있는 API를 제공한다. 직렬 포트 자체를 컨트롤 할 수 있는 API 들과 데이터를 주고 받을 수 있는 API를 제공하고 있다.

(3) GPS

GPS 모듈을 제어하고 위치 정보를 가져오는 API를 제공하고 있다.

5. Network 모듈

Network 모듈에서는 인터넷을 통한 통신에 대한 API를 다룬다. 이 모듈에서는 하부 네트워크를 통제하는 API와 Socket 통신에 관련된 API를 포함하여 그보다 상위 레벨에 해당하는 프로토콜인 HTTP와 SIP 등을 지원하는 API도 가지고 있으며 Provisioning 기능을 수행하는 API도 포함한다.

(1) PPP 연결 제어

단말기가 네트워크에 연결되기 위해서 가장 하부 단에서 지원하는 프로토콜이 PPP이다. 이는 CDMA 통신을 이용해서 네트워크를 사용하는 것을 가능하게 해주는데, 이러한 PPP를 연결하고 끊고 제어하는 API를 제공한다.

(2) Socket 관련 API

T-PAK에서 지원하는 Socket API는 크게 TCP와 UDP 두가지로 나누어 진다. 각 프로토콜을 이용해서 연결을 하거나 데이터를 주고 받을 수 있는 API를 제공한다.

(3) HTTP 프로토콜

상위 네트워크 프로토콜인 HTTP 프로토콜을 제공한다. HTTP 서버에 접속을 하거나 서버에서 콘텐츠를 받아오는 기능 등을 수행할 수 있다.

(4) SIP 프로토콜

Session Initiation Protocol에 관련된 API를 제공한다. 이 API를 이용해서 SIP 다이얼로그 세션을

연다던가 메시지를 전송하는 것들이 가능하다.

6. Telephony 모듈

Telephony 모듈에서는 핸드폰 단말의 전화 기능을 담당하고 있다. 이곳에서는 음성 통화 기능과 단문 전송(SMS) 기능을 통제할 수 있는 API를 포함하고 있다.

(1) 음성 통화 기능

전화 걸기 기능, 전화 통화 종료 기능, 전화 수신 수락 기능, 전화 수신 거절 기능, 통화시 다른 전화 수신 및 발신 기능, 전화 수신 시 정보 획득 기능 등을 제어할 수 있도록 API를 제공한다.

(2) SMS 전송 기능

SMS의 전송 기능이나 전송 후 처리 결과를 얻을 수 있는 API들을 제공한다.

7. Security 모듈

이 모듈에서는 보안 통신 기능을 제공하고 있다. 이를 위해서 SSL 프로토콜을 지원하고 있으며 보안 통신을 위해 필요한 인증서 관리 기능도 제공하고 있다.

(1) SSL

보안 통신을 제공하는 SSL 프로토콜에 해당하는 API를 제공하고 있다.

(2) 인증서 관리

보안을 위한 인증서를 관리하고 인증서의 정보를 조회할 수 있는 API를 제공한다.

8. Media 모듈

Media 모듈에서는 음악이나 영상등 매체 재생이나 녹화 등의 기능을 담당하고 있다. 각 매체 처리기의 기기를 관리하거나 매체 처리기 장치를 제어하고 영상이나 음성 데이터를 재생하거나 녹음하는 기능들을 제공한다.

9. Terminal Resource 모듈

Terminal Resource 에서는 단말에서 지정하는 리소스를 관리하는 기능을 가지고 있다. 단말이 지정하는 리소스에는 전화번호부에 들어있는 정보들과 폰트 데이터들, SMS 메시지, 벨소리, 사진 데이터, 통화 기록 등 핸드폰 단말에 저장되는 모든 정보를 나타낸다.

10. WFC 모듈

WFC 는 기존의 SKT에서 서비스 하고 있던 UI 라이브러리이다. 이들 UI 라이브러리를 제공함으로써 기존의 개발자들이 쉽게 T-PAK 플랫폼에 익숙해지도록 할 수 있으며 WFC 로 만들어졌던 어플리케이션들을 활용할 수 있다.

IV. 모듈 업그레이드 기능

T-PAK 이 가지는 주요 특징 중 하나는 각 모듈들이 상호 의존성이 없기 때문에 독립적으로 추가, 갱신, 삭제가 가능하다는 점이다. 이들 독립 모듈들은 Dynamic System Library (DSL) 포맷을 가지며 Base 모듈에서 이 포맷을 가지는 파일을 로드하여

활성화 할 수 있는 기능을 가지고 있다. T-PAK 플랫폼에서는 업그레이드가 필요하다고 판단이 되면 다운로드를 실행시켜서 유선이나 무선을 통해서 DSL 파일을 다운로드 하고 기존에 설치되어 있던 모듈을 교체할 수 있다.

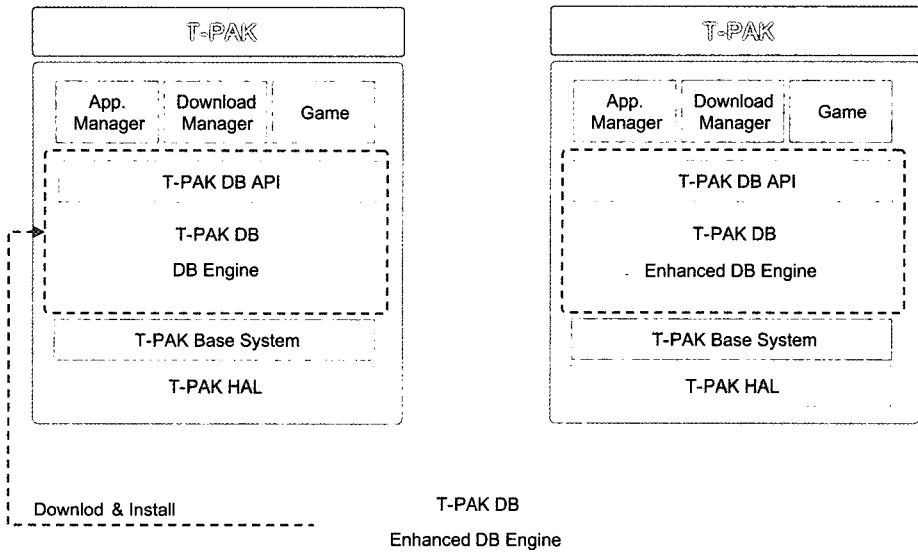
이러한 모듈 업그레이드 기능은 다음의 특징을 가진다.

- 각 모듈화된 DSL 의 동적인 확장이 가능하다.
 - 중심 엔진을 비롯한 모든 API 가 확장이 가능하다.
 - 중심 엔진의 확장을 통해서 새로운 서비스가 가능하다.
- 유선 / 무선 망을 통한 업데이트가 가능하다.
 - 유선망을 통해서 USB 케이블을 통해서 업데이트가 가능하다.
 - 무선망을 통해서 다운로드가 가능하다.
- 단말 출시 이후
 - 단말 버그나 오류로 인한 플랫폼의 교체가 가능하다.
 - 새로운 서비스를 바로 적용 가능하다.

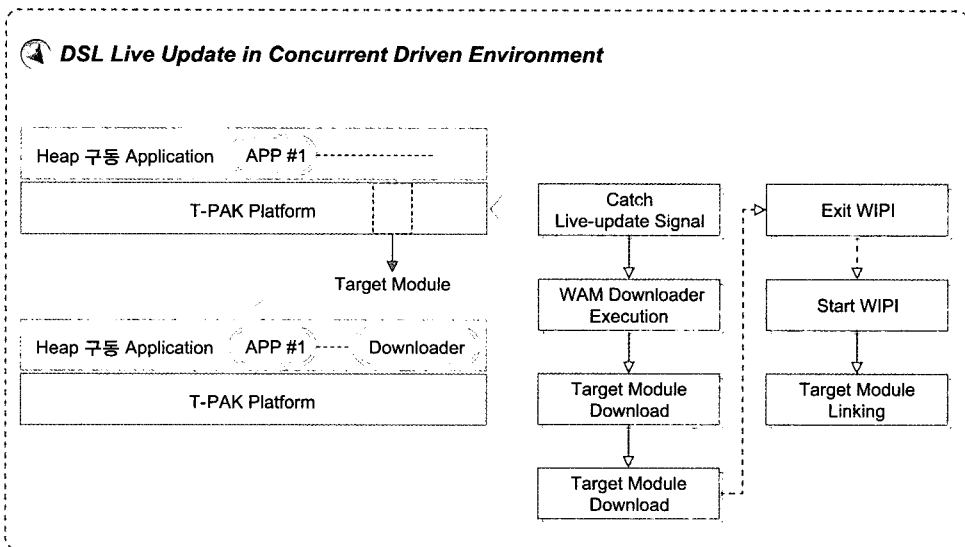
(그림 4)는 실제로 모듈 업그레이드를 적용하는 예를 보여준다. 왼쪽의 T-PAK 은 기존에 출시될 당시의 플랫폼의 모습을 나타낸다. 이후에 Sub-System 중에서 DB 에 관련된 엔진이 보다 나은 알고리즘이나 안정적으로 동작하는 알고리즘으로 새롭게 구현되었다고 가정한다.

이 경우 T-PAK 의 Sub-System 의 DB 엔진만 따로 다운로드 하여 인스톨을 하면 새롭게 바뀐 DB 엔진이 인스톨 되어 동작하는 모습을 볼 수 있다. 이때 DB 엔진 외의 다른 모듈들은 변하지 않고 그대로 남아있으면서 그대로 동작할 수 있다.

(그림 5)는 실제로 T-PAK 시스템에서 Live DSL Update 과정을 보여준다. T-PAK 플랫폼이



(그림 4) 모듈 업그레이드 예



(그림 5) Live DSL Update Flow

구동되어 있을 경우 외부로부터 Live-Update 가 필요하다는 신호를 받으면 그때부터 모듈의 다운로드가 동작하고 모듈을 다운로드 받아서 설치한 다음 플

랫폼이 다시 시작하면 변경된 모듈이 들어간 새로운 T-PAK 시스템이 구동되는 모습을 볼 수 있다.

V. 결 론

지금까지 SKT에서 전략적으로 개발하고 있는 T-PAK 플랫폼에 대해서 그 내부 구조와 동작 구조를 살펴보았다. 앞으로 출시되는 휴대 단말기들의 경우 플랫폼을 지원하지 않을 수는 없다. 플랫폼이 모듈화 되고 세분화 되지 않는다면 플랫폼에 문제가 생기거나 새로운 기능을 추가하는데 많은 어려움이 생길 수 밖에 없다. 따라서 플랫폼 자체도 세부적으로 모듈화가 되어 부분 업데이트가 가능하게끔 설계가 되고 개발이 되고 있다. 모듈화된 플랫폼 기반의 휴대 단말기의 개발은 앞으로 계속 진행되는 추세이며 많은 단말기가 이러한 접근 방식을 채택할 것이라 예상된다.

더구나 국내 이동 통신 사업의 경우, WIPI 탑재가 필수가 되고 있기 때문에, T-PAK과 같은 WIPI 확장 형태의 단말 플랫폼은 이동 통신 사업자와 핸드셋 제조사 모두에게 도움을 줄 수 있는 현실적인 솔루션이라고 볼 수 있다.

[참 고 문 헌]

- [1] WIPI 포럼, <http://www.wipi.or.kr/>



이은복

1994년 한양대학교 전자공학과 학사
 1994년 ~ 1995년 대우전자 vcr연구소 연구원
 1995년 ~ 2001년 디지털 사업본부 CDMA프로토콜 개발
 1995년 ~ 현재 SK Telecom
 2001년 ~ 현재 터미널개발 업무 수행



임종태

1986년 연세대학교 전자공학과 학사
 1988년 연세대학교 대학원 전자공학과 석사
 1993년 연세대학교 대학원 전자공학과 박사
 1993년 ~ 1998년 SK Telecom 중앙연구소 전파기술팀장
 1998년 ~ 2000년 정보통신연구진흥원 전파방송기술평가실장

술평가실장

2000년 ~ 2005년 SK Telecom Network 연구원, 엔지니어링 기술개발팀장, 차세대 기술개발팀장, Access망 개발팀장(상무)
 2003년 ~ 2004년 WCDMA USIM 기술개발관리단장
 2005년 ~ 2006년 NGMC Forum 서비스 및 마켓 분과위원장
 2005년 ~ 2006년 Platform 연구원장
 2005년 ~ 현재 텔레매틱스 표준화 포럼 의장
 2006년 ~ 현재 Access 기술연구원장



김종배

1998년 아주대학교 산업공학과 학사
 1998년 ~ 2002년 LG전자 CDMA 단말연구소 연구원
 2002년 ~ 현재 SK Telecom Access 기술연구원



김후중

1988년 서강대학교 전자공학과 학사
 1988년 LG전자 연구원
 1993년 서강대학교 대학원 전자공학과 석사
 1995년 ~ 현재 SK Telecom
 2002년 ~ 현재 SK Telecom 터미널 개발 팀장