제 18 회 한국정보처리학회 추계학술발표대회 논문집 제 9 권 제 2 호 (2002.11)

Java Card SIM API 의 Toolkit Registry 구현에 관한 연구

임현준, 김현아, 정재우, 김광훈
경기대학교 전자계산학과 워크플로우 연구실
e-mail : gwbasic@kyonggi.ac.kr

A Study on Toolkit Registry Implementation of the Java Card SIM API

Hyeon-Jun Lim, Hyun-Ah Kim, Jae-Woo Jung, Kwang-Hoon Kim
Dept. of Computer Science, Kyong-gi University

요 약

SIM 카드는 개인식별을 위해 GSM 단말기에 내장되는 스마트 카드의 한 종류이다. 그러나 기존의 SIM 카드는 어플리케이션을 업그레이드하거나 추가하기가 쉽지 않고 어플리케이션이 하드웨어에 종속적인 단점이 있어 SIM 카드의 활용범위를 좁히는 결과를 가져왔다. 본 논문에서는 이러한 단점을 해결할 수 있는 대안으로 제시되고 있는 Java Card 와 Java Card 기술을 알아보고 이러한 Java Card 기술을 SIM 카드에 적용하기 위하여 필요한 Java Card SIM API 에서의 Toolkit Registry 에 대한 구현 내용을 기술한다.

1. 서론

SIM(Subscriber Identity Module; 가입자 인식 모듈) 카드는 개인식별을 위해 GSM 단말기에 내장되는 스마트 카드의 한 종류이다. 이러한 SIM 카드는 스마트 카드를 이동전화서비스를 위한 모든 데이터 처리과정이 암호, 인증 등 강력한 보안기능을 바탕으로 이루어지도록 하면서도 개인이 휴대할 수 있는 일종의 정보 단말기로써 활용한 것으로써 국내에서 서비스중인 이동전화 방식인 코드분할다중접속(CDMA) 방식 단말기에 내장되는 UIM(User Identity Module)과 달리 범유럽 표준이동전화(GSM) 방식 단말기에 내장되는 탈착이 가능한 카드이다.

그러나 기존의 SIM 카드는 활용에 있어 스마트 카드의 다음과 같은 단점을 그대로 가지고 있다. 첫째, 스마트 카드 어플리케이션 개발에 많은 시간과 자금이 필요하고 일단 카드가 발급된 이후에는 스마트 카드 어플리케이션을 업그레이드하거나 추가하기가 쉽지 않다. 둘째, 어플리케이션이 특정 하드웨어 및 COS (Card Operating System)에 종속적인 결과를 가져오고 새로운 기능이나 어플리케이션을 추가하고자 경우 기존의 COS 나 어플리케이션을 수정해야 하는 하는 경우가 발생한다. 이러한 문제점은 스마트 카드 및 SIM 카드의 활용 범위를 좁히는 결과를 가져왔다.

본 논문에서는 이러한 스마트 카드와 SIM 카드의 단점을 해결할 수 있는 대안으로 제시되고 있는 Java Card 와 Java Card 기술을 알아보고 이러한 Java Card 기술을 SIM 카드에 적용하기 위하여 필요한 Java Card SIM API 에서의 Toolkit Registry 에 대한 구현 내용을 기술한다.

본 논문의 구성은 다음과 같이 하였다. 2 장에서는 Java 와 Java Card 기술을 설명하였고, 3 장에서는 Java Card SIM API 를, 4 장에서는 Toolkit Registry 구현 내용을 설명하였으며, 마지막으로 5 장에서는 결론 및 향후 연구 과제를 제시하였다.

2. Java Card 와 Java Card 기술

스마트 카드와 SIM 카드의 이러한 문제점을 해결할 수 있는 대안으로 제시되고 있는 것이 Java Card 이다. Java Card 는 Java 언어로 작성된 어플리케이션을

* 본 논문은 한국전자통신연구원 위탁과제(과제번호:0701-2002-0016)의 일부 결과임

실행시킬 수 있는 스마트 카드의 한 종류로써 이 Java Card 에는 Java Card 기술이 사용되는데 Java Card 기술은 Java 로 쓰여진 프로그램이 스마트 카드나 혹은 그 밖에 제한적인 자원을 가진 장치에서의 동작을 가능하게 하는 기술이다. 이러한 Java Card 기술을 적용했을 경우의 장점은 다음과 같다

- 하드웨어 플랫폼에 독립적 : 어떤 하드웨어를 사용하더라도 자바 가상 머신, 즉 Java Virtual Machine 을 가지고 있다면 자바 카드 어플리케이션의 구동이 가능하다.
- 보안성 : Java Card 기술은 자바 언어 자체의 보안 특성의 많은 부분을 수용하고 있으며, 자바 카드 어플리케이션들이 각각 자바 카드 애플릿(어플리케이션) 방화벽에 의해 서로 보호된다.
- 개발 시간 및 비용을 절감 : 스마트 카드 어플리케이션을 개발하는데 필요한 공통적인 API 를 제공함으로써 보다 효율적인 개발환경을 제공한다.
- 다중 어플리케이션 : 한 장의 카드에 복수의 어플리케이션을 탑재하는 다중 어플리케이션이 가능하다.

3. Java Card SIM API

이러한 Java Card 기술을 적용한 SIM 에서 사용하는 어플리케이션을 쉽게 개발할 수 있도록 제공하는 응용 프로그램 인터페이스, 즉 API 를 Java Card SIM API 라고 한다. 이것은 기존의 Java Card 기술에서 제공하는 API 에 SIM 에서 사용하는 어플리케이션의 특성을 고려한 별도의 API 가 추가되는 것이며, 이것에 대한 내용은 표준기관인 Third Generation Partnership Project(3GPP)의 3GPP TS 03.19 문서[8]에 명시되어 있다.

3-1. GSM Java Card 의 시스템 구조

다음 그림 1 은 3GPP TS 03.19 문서[8]에 있는 기존의 Java Card 시스템에 Java Card SIM API 가 추가된 GSM Java Card 의 시스템 구조를 보여주고 있는 그림이다.

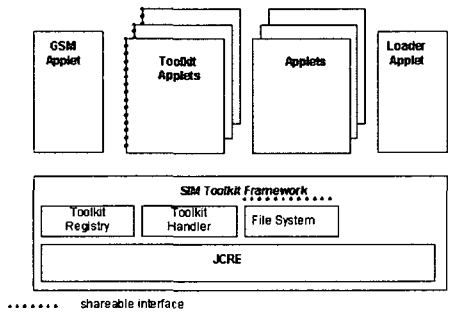

그림 1. GSM Java Card 시스템 구조

이러한 GSM Java Card 시스템 구조에서 각 부분에 대한 설명은 다음과 같다.

- SIM Toolkit Framework : GSM Java Card 이며 JCRE, Toolkit Registry, Toolkit Handler, File System 으로 구성되어 있다.
- JCRE : Java Card 2.1 Runtime Environment Specification[5]에 명세되어 있으며 어떤 특정한 applet 을 선택할 수 있고 그러한 applet 의 APDU 처리를 그러한 applet 에 전송할 수 있다.
- Toolkit RegistryToolkit : applet 의 모든 등록 정보와 JCRE 에 대한 이러한 등록 정보의 연결을 다룬다.
- Toolkit Handler : system handler 와 toolkit applet suspension 과 같은 toolkit protocol 을 다룬다.
- File System : 카드 발행사 파일 시스템을 포함하고 있으며, 파일 접근 제어와 applet 파일 context 를 다룬다. 이것은 shareable interface sim.access.SIMView 를 구현하고 있는 JCRE owned object 이다.
- Applets : javacard.framework.applet 을 상속받고 Java Card 2.1 Runtime Environment Specification[5] 에서 정의되어진 것처럼 process, select, deselect, install 의 entry point 들을 제공하는 applet 이다.
- Toolkit applets : javacard.framework.applet 을 상속받아서 applet 과 똑같은 entry point 들을 제공하고 이러한 applet 들이 그들의 processToolkit 메소드의 호출에 의해서 트리거될 수 있도록 shareable interface sim.toolkit.ToolkitInterface 를 구현한 applet 이다. 이러한 applet 의 AID 는 TS 101 220[7]에 정의되어 있다.
- GSM applet : Java Card 2.1 Runtime Environment Specification[5]에서 정의되어진 것처럼 default applet 이며, 이것은 예를 들면 다른 applet 이 SELECT AID APDU 를 통하여 선택되어질 때 이러한 applet 의 deselect 메소드가 호출되어지는 것 같은 일반적인 applet 으로써 작동한다. 이러한 applet 의 AID 는 TS 101 220[7]에 정의되어 있고 TS 11.11[10] APDU 들, CHV1/2, GSM 인증 알고리즘과 TS 11.11[10]에 의한 가입자 파일 접근 제어를 다룬다.
- Loader applet : applet loading specification TS 03.48[9]에서 명세 되어진 것처럼 applet 의 설치와 제거를 다루는 applet 이다.
- Shareable interface : 이것은 Java Card 2.1 specification[4]에 정의되어있다.

이러한 GSM Java Card 시스템 구조에서 구현과 관계된 부분은 *Toolkit Registry, Toolkit Handler, File System* 이다.

3-2. GSM Java Card 의 시스템 구조

3GPP TS 03.19 문서[8]의 부록 A 에는 Java Card SIM API 에 대한 구현부가 없는 Java 소스 파일들이 제공되고 있다.

다음 그림 2 는 Java Card SIM API 의 클래스, 인터페이스의 계층 구조를 보여주고 있는 그림이다.

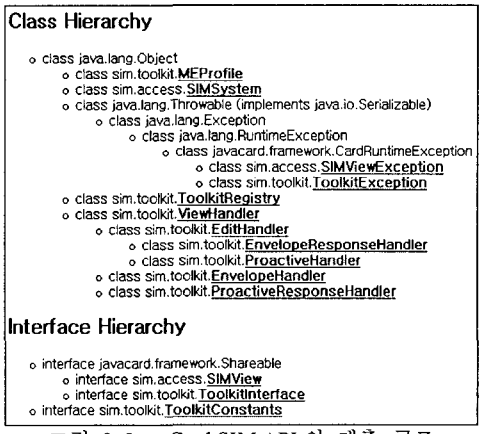

그림 2. Java Card SIM API 의 계층 구조

이러한 Java Card SIM API 의 각 클래스, 인터페이스는 다음과 같다.

- MEProfile 클래스 : handset profile 을 조회하기 위한 메소드를 포함하는 클래스이다.
- SIMSystem 클래스 : GSM File system 의 논리적 view 를 얻기 위한 방법을 제공하는 클래스이다.
- SIMViewException 클래스 : 에러의 경우에 SIMView 인터페이스의 메소드에 의해서 예외가 던져질 수 있는 특수한 예외들을 넣은 클래스이다.
- ToolkitException 클래스 : Throwable 클래스를 확장하고 이 패키지의 클래스가 문제가 있는 경우에 특수한 예외들을 던질 수 있도록 하는 클래스이다.
- ToolkitRegistry 클래스 : 모든 애플릿의 life time 동안 어떠한 Toolkit applet 이라도 install 단계 동안에 Toolkit applet 의 구성(configuration)을 등록하고 Toolkit applet life time 동안 내내 Toolkit applet 의 구성(configuration)을 될 수 있는 한 바꿀 수 있게 하는 클래스이다.
- ViewHandler 클래스 : Terminal Response data field 또는 a BER-TLV element (Envelope data field 또는 Proactive command) 에서와 같은 Simple TLV List 를 다루기 위한 기본적인 서비스를 제공하고 기본적인 메소드를 포함하는 클래스이다.
- EditHandler 클래스 : simple TLV 요소들의 리스트의 구성에 대하여 기본적인 클래스이다.
- EnvelopeResponseHandlerEnvelope 클래스 : response data field 를 다루기 위한 기본적인 메소드를 포함하고 있는 클래스이다. 이 클래스는 Temporary JCRE Entry Point Object 이다.
- ProactiveHandler 클래스 : Proactive command 의 정의에 대한 기본적인 클래스이다. 이 클래스는 Temporary JCRE Entry Point Object 이다.
- EnvelopeHandlerEnvelope 클래스 : data field 를 다루기 위한 기본적인 메소드를 포함하고 있는 클래스이다. 이 클래스는 Temporary JCRE Entry Point Object 이다.
- ProactiveResponseHandler 클래스 : ProactiveResponseHandler 클래스는 Terminal Response data field 를 다루기 위한 기본적인 메소드를 포함하는 클래스이다. 이 클래스는 Temporary JCRE Entry Point Object 이다.
- SIMView 인터페이스 : SIMView 인터페이스는 GSM System Service 와 어떠한 SIM Toolkit applet 또는 다른 applet 간에 인터페이스이다. 모든 메소드들은 GSM 11.11 Specification[10]을 기반으로 하고 있다. 이 인터페이스는 JCRE owned object 로 구현되어야 한다.
- ToolkitInterface 인터페이스 : ToolkitInterface 인터페이스는 Toolkit applet 이 등록 정보에 따라 Toolkit Handler 에 의해서 트리거 될 수 있기 위해서 Java Card 의 javacard.framework.Applet 클래스를 확장한 Toolkit applet 에 의해서 구현되어야 한다. Toolkit applet 은 Toolkit applet 이 이벤트를 통지받을 수 있도록 shared 메소드 processToolkit 을 구현해야만 한다.
- ToolkitConstants 인터페이스 : Toolkit applet 과 관련된 상수들을 모아놓은 인터페이스이다.

이러한 Java Card SIM API 의 클래스, 인터페이스에서 Toolkit Registry, Toolkit Handler, File System 의 구현과 관계된 클래스, 인터페이스는 각각 다음 표 1 과 같다.

구성요소	구현과 관계된 클래스,인터페이스
Toolkit Registry	ToolkitRegistry 클래스
Toolkit Handler	MEProfile 클래스 ViewHandler 클래스 EditHandler 클래스 EnvelopeResponseHandlerEnvelope 클래스 ProactiveHandler 클래스 EnvelopeHandlerEnvelope 클래스 ProactiveResponseHandler 클래스
File System	SIMSystem 클래스 SIMView 인터페이스

표 1. 구현과 관계된 클래스, 인터페이스

4. Toolkit Registry 의 구현

Toolkit Registry 는 Toolkit Applet 을 어떤 이벤트나 메뉴 선택에 대하여 등록하거나 등록을 수정, 또는 등록을 제거할 수 있게 하는 역할을 함으로써 어떤 이벤트나 메뉴가 선택되었을 시에 등록된 Toolkit Applet 의 processToolkit 메소드가 호출되게 하여 Toolkit Applet 이 어떤 이벤트나 메뉴 선택에 대하여 적절한 처리를 할 수 있게 한다. 이러한 Java Card SIM API 에

서 Toolkit Registry 의 구현은 ToolkitRegistry 클래스를 구현하는데 있는데 다음 그림 3 은 Toolkit Registry 의 구조를 보여주고 있다.

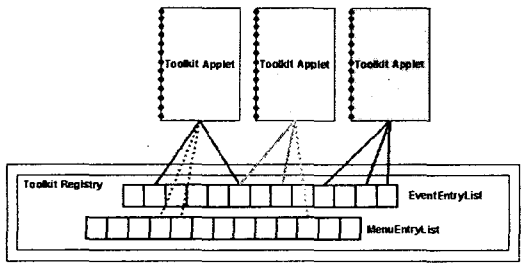

그림 3. Toolkit Registry 의 구조

그림 3 과 같이 Toolkit Registry 는 이벤트와 관련된 EventEntryList 와 메뉴와 관련된 MenuEntryList 를 가지고 있어서 이벤트와 메뉴에 대하여 등록, 등록 수정 또는 등록 제거될 수 있도록 한다. 수행절차는 다음과 같다.

1. Toolkit Applet 이 Toolkit Registry 에 등록되어지기 위하여 Toolkit Registry 의 getEntry() 메소드를 호출

2. Toolkit Registry 에서 JCSystem.getPreviousContextAID() 메소드를 호출하여 호출한 Toolkit Applet 의 고유한 AID(Application IDentifier)를 알아냄.

3. 알아낸 AID 를 가지고 각 지정된 이벤트나 메뉴 리스트에 등록하거나 수정함으로써 해당하는 Toolkit Applet 들을 등록, 등록 수정, 등록 제거

이렇게 Toolkit Registry 에 등록된 Toolkit Applet 들은 이벤트나 메뉴선택에 의하여 command 가 발생하게 되면 다음 그림 4 의 흐름에 따라 command 가 처리된다.

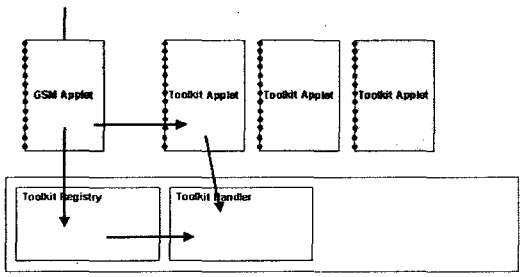

그림 4. command 의 처리 흐름

그림 4 에서의 command 가 처리되는 수행절차는 다음과 같다.

1. command 가 GSM Applet 으로 전달

2. GSM Applet 은 해당 이벤트나 메뉴에 대하여 등록된 Toolkit Applet 들을 찾기 위해서 Toolkit Registry 에 등록된 AID 를 사용하여 JCSystem.getAppletShareableInterfaceObejct() 메소드를 호출

3. JCRE 에서는 JCRE 에 등록된 Toolkit Applet 들의 getShareableInterfaceObject()를 호출하여 해당 Toolkit Applet 을 찾음.

4. JCSystem.getAppletShareableInterfaceObejct() 메소드로부터 넘겨받은 Toolkit Applet 의 참조를 통하여 GSM Apple 은 해당 Toolkit Applet 의 processToolkit() 메소드를 호출

5. 2 에서 4 의 절차를 이벤트나 메뉴에 등록된 Toolkit Applet 들에 반복한다. 경우에 따라서는 이벤트나 메뉴의 종류에 따라서 처음 등록된 Toolkit Applet 만 수행되는 경우 등이 있다.

5. 결론 및 향후 발전 과제

본 논문에서는 기존 SIM 카드의 단점을 해결할 수 있는 대안으로 제시되고 있는 Java Card 기술을 알아보고 이러한 Java Card 기술을 SIM 카드에 적용하기 위하여 필요한 Java Card SIM API 에서의 Toolkit Registry 에 대한 구현 내용을 기술하였다. 향후 발전 과제로는 나머지 요소들인 Toolkit Handler, File System 들을 구현하고 이러한 Java Card SIM API 를 실제 SIM 카드에 적용할 수 있도록 해야 한다.

참고문헌

[1] 김성웅, Smart Card with JAVA Technology, 2001.6 지급결제와 정보기술(Payment Systems & IT), 금융결제원
[2] Zhiqun Chen, Java Card Technology for Smart Cards
[3] http://java.sun.com/products/javacard
[4] Java Card 2.1 Specification, Sun Microsystems, Inc.
[5] Java Card 2.1 Runtime Environment Specification, Sun Microsystems, Inc.
[6] Java Card 2.1 Virtual Machine Specification
[7] ETSI TS 101 220 - "Integrated Circuit Cards (ICC); ETSI numbering system for telecommunication ; Application providers (AID)".
[8] 3GPP TS 03.19, 3rd Generation Partnership Project ; Technical Specification Group Terminals ; Subscriber Identity Module Application Programming Interface (SIM API) for Java Card™ ; Stage 2 (Release 1999)
[9] 3GPP TS 03.48, Security mechanisms for SIM application toolkit ; Stage 2
[10] 3GPP TS 11.11 Specification of the Subscriber Identity Module Mobile Equipment (SIM - ME) interface
[11] 3GPP TS 11.14 Specification of the SIM Application Toolkit for the Subscriber Identity Module Mobile Equipment (SIM - ME) interface