2001년 한국정보처리학회 추계 학술발표논문집 제8권 제2호

COS API 품질 평가를 위한 테스트 데이터 생성에 관한 연구

염희균, 김상영, 황선명
대전대학교 컴퓨터공학과

e-mail: yhg1124@zeus.taejon.ac.kr

A Study on the Test Data Generation for the COS API Quality Evaluation

Hee-Gyun Yeom, Sang-Young Kim, Sun-Myung Hwang
Dept of Computer Engineering, Daejeon University

요약

최근 다양한 어플리케이션과 전용 임베디드 시스템의 개발 필요성이 증가하고 있으며 이는 Smart Community를 추구하는 디지털 지식사회의 중요한 이슈가 되고 있다. 현재 상용화되고 있는 대부분의 실시간 임베디드 시스템들을 다양한 측면에서 자사의 제품 기준에 따르는 기술적 특징과 성능 지표를 제공하고 있으며 이를 통한 성능 품질 등의 판단 근거로 제시되고 있다. 그러나 각 자료들이 공인 기관에서 제공되지 않아 자료의 신뢰도가 낮으며 따라서 성능측정 및 품질척도로 비교하는데는 어려움이 있다 따라서 본 연구는 이러한 문제점을 착안하여 자바 카드 API의 품질 평가를 위해서 사전에 테스트를 거쳐 좀더 신뢰성과 안전성이 높은 자바 카드 애플릿을 개발하고자 한다. 이에 가장 중요한 부분 중에 하나인 에러체크를 통해 임의적으로 에러 처리 시나리오를 작성해서 그 시나리오를 거친 테스트 데이터를 생성한 후, 그 데이터를 가지고 API의 품질 평가를 하고자 하는데 목적이 있다. 이에 본 논문에서는 테스트 데이터 생성 시나리오를 제안하였다. 단지 제안만 해 놓았기 때문에 향후 이 시나리오를 거친 테스트 데이터 생성을 연구하고자 한다. 또한, Verification을 위해 수학에 기초를 둔 정형 기법(Formal Method)을 이용하여 좀더 깊이 있고, 정확한 검증을 수행 하고자 한다.

1. 서론

최근 다양한 어플리케이션과 전용 임베디드 시스템의 개발 필요성이 증가하고 있으며 이는 Smart Community를 추구하는 디지털 지식사회의 중요한 이슈가 되고 있다. 따라서 우리 나라도 Smart Community에 능동적인 대처 방안과 기술의 경쟁력 확보가 요구되고 있다 현재 상용화되고 있는 대부분의 실시간 임베디드 시스템들을 다양한 측면에서 자사의 제품 기준에 따르는 기술적 특징과 성능 지표를 제공하고 있으며 이를 통한 성능 품질 등의 판단 근거로 제시되고 있다. 그러나 각 자료들이 공인 기관에서 제공되지 않아 자료의 신뢰도가 낮으며 따라서 성능측정 및 품질척도로 비교하는데는 어려움이 있다. 현재 개발된 품질 및 성능평가 시스템으로는 한국 래쇼날 소프트웨어의 액티비티와 산출물의 통합관리(UCM)를 통합 품질 관리 시스템과 핸디소프트의 KM 솔루션, 기능성, 사용성 중심의 웹 어플리케이션 시험평가를 하는 제니우스 기술의 e-TEST Suite, 컴퓨웨어 단위테스트 솔루션을 개발한 DevPartner Studio, Mercury Interactive사의 테스트 자동화 솔루션 등과 현재 썬 마이크로시스템사의 JavaTest을 통한 Smart IC Card를 읽는 COS용 Java API의 테스트 도구인 JavaTest 등이 있다[1]. 본 연구에서는 JavaTest를 통하여 개발되었거나 개발되어야 할 자바 카드 애플릿에 대한 사전 수행과 테스트를 통하여 이들의 테스트 데이터를 생성하여 품질, 성능을 평가 하고자 한다. 본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로 자바 카드와 자바 카드 API, 자바 애플릿의 동작에 대해 살펴보며, 3장에서는 공개된 자바 카드 애플릿을 가지고 객체지향 모델링 언어인 UML을 이용하여 Method Trace를 통해 API 정적 분석을 한 후, 4장에서는 자바 카드 API의 일반적인 품질 평가를 위한 테스트 데이터 생성을 제안하고, 마지막으로 5장에서 결론 및 향후 연구 과제로 끝을 맺고자 한다.

2. 관련 연구

2.1 자바 카드

자바 카드란 COS(Card Operating System) 위에 JCVM(Java Card Virtual Machine)이 랩핑(Wrapping)되어 있는 구조의 스마트 카드를 말한다. 자바 카드 API는 자바 카드 상에서 Java를 이용한 소프트웨어 개발에 필요한 API들을 정의한 것이다[2].

자바 카드 API는 전자상거래, 네트워크 접근, 인증을 위한 차세대 네트워크 기술을 제시하였다. Bull, Gemplus, Schulmberger 등 전 세계 스마트 카드 제조 회사의 90%이상이 자바 카드의 개발을 위해 라이센스를 이미 받은 상태이다[4].

2.2 자바 카드 API

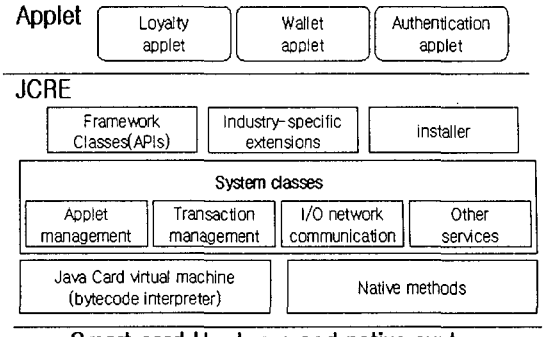
자바 카드는 카드 애플릿 개발자들을 위한 도구로 자바 카드는 하드웨어 플랫폼과 독립적인 하나의 언어로 지정되어 있고, 자바 API는 카드 어플리케이션 코드를 지정한다. ISO/IEC 7816-4에 기반을 둔 독립 플랫폼 레이어는 자바 어플리케이션 사이에 있고, PC/SC, 원래의 코드, 자바 드라이버 등으로의 인터페이스를 제공하고 있다. 초기 버전의 자바 카드는 Sun Microsystems의 테스트를 완벽하게 통과하지 못했고, SIM 카드 벤더간 상호운용성을 허용하지 못했으며, 특히 개방형 플랫폼을 제공하지 않았었다[7]. 또한, 스마트 카드의 응용으로 자바로 프로그래밍 되어있다. 애플릿이 실행되기 위해서는 JCRE가 사용된다.

자바 카드 API의 종류를 표로 정리하면 다음과 같다[3].

Core Package	Extension Package
Java.lang.*	Javacardx.crypto.*
Javacard.framework.*	
Javacard.sequrity	

[표 1] 자바 카드 API 종류

다음은 자바 카드 시스템 구성에서 애플릿의 동작을 보여주는 그림이다[3].



[그림 1] JC System Component

모든 애플릿은 Javacard.framework.Applet 클래스의 서브클래스가 만들어짐으로써 실행된다. 여기서 테스트하는 것이 바로 애플릿이 JCRE로 실행되기 전에 애플릿의 테스트 데이터를 만들어 그것들의 품질을 평가하여 좀더 신뢰성과 안전성이 높은 애플릿을 구현 하고자 한다.

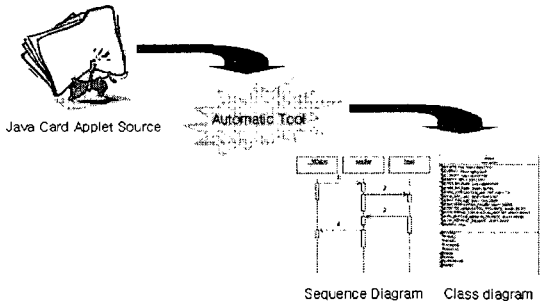
Javacard.framework.Applet 클래스의 메소드를 살펴보면 다음과 같다[3][7].

Public static void	Install (byte[] bArray, short bOffset, byte bLength) The JCRE calls this static method to create an instance of the Applet subclass
Protected final void	Register() This method is used by the applet to register this applet instance with the JCRE and to assign the default AID in the CAP file to the applet instance
Protected final void	Register (byte[] bArray, short vOffset, byte bLength) This method is used by the applet to register this applet instance with the JCRE and to assign to the applet instance the AID *specified in the array bArray*.
Public boolean	Select () The JCRE calls this method to inform the applet that it has been selected
Public abstract void	Process (APDU apdu) This JCRE calls this method to instruct the applet to process an incoming APDU command.
Public void	Deselect () The calls this method to inform the currently selected applet that another (or the some) applet will be selected.

[표 2] Javacard.framework.Applet 클래스 메소드

3. 자바 카드 애플릿의 정적 분석

현재 cJDK2.1.1에 포함하고 있는 Wallet 애플릿을 UML 다이아 그램 중, Class 다이아 그램, Sequence 다이아 그램과 그것들의 Method Trace를 통해 애플릿의 동작을 정적으로 살펴볼 수 있다. 이런 과정은 그림 2 에서 간단히 보여준다.



[그림 2] 자바 카드 애플릿 정적 분석

3.1 Method trace

자동화 도구를 통해서 Method trace를 수행한 결과 다음과 같은 평가를 할 수 있다.

-Number of lines of code(LOC)
-Number of lines of comments
-Ratio of the amount of comments to the amount of program code
-Structure of the program code
-Number of functions
-Nesting depth
-"dead" code

3.2 Verification

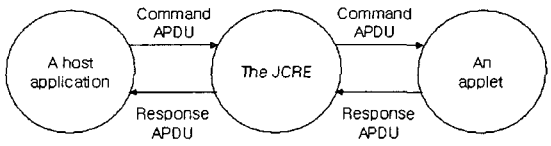
설계, 이후 소스코드와 명세간의 일관성 검증을 통해 설계단계에서의 오류를 추출해내어 구현단계 이후에 오류 수정을 하는 것 보다 훨씬 비용을 줄일 수 있고, 안전하고 신뢰성 높은 애플릿을 완성할 수 있다. 이런 검증도구로 현재 나와있는 도구로는 B toolkit이 있다. 이 도구는 증명하는 엔진으로, 정형화 검증을 자동으로 수행하게 된다.

4. 자바 카드 API 테스트 데이터 생성

4.1 에러 체크의 필요성

스마트 카드 응용프로그램 및 애플릿 개발 시 특별히 중요하고, 발견되지 않은 오류로 인해 카드에 장애가 발생하게 되면, 카드에 저장된 데이터가 결정적으로 상실할 수 있게 된다. 따라서 애플릿에 대한 평가를 위해서는 일반화된 Verification이 필요하다. 자바 카드 API를 처리하게 함으로서 좀 더 안정적이고 신뢰성 있는 프로그램 작성이 가능하다.

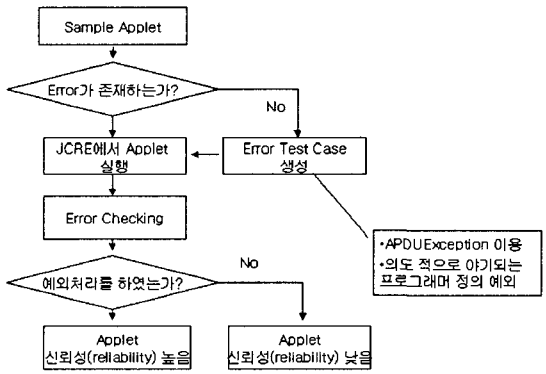
4.2 자바 카드 애플릿의 에러 체크 범위



[그림 3] 자바 카드 애플릿의 에러 체크 범위

애플릿이 스마트 카드에서 동작하게 되면 APDU 명령을 통해 외부에 접근(Host)을 하면, 명령을 다시 애플릿에게 보내진다. 이런 규칙은 ISO 7816-4에 애플릿과 호스트 어플리케이션은 반듯이 APDU 명령의 각 필드의 의미 있는 값들에 일치하여야 한다는 표준을 따르고 있다[3][6].

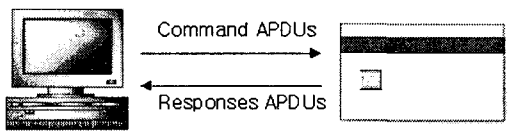
4.3 자바 카드 애플릿의 에러 체크 단계 [그림 4]



본 논문에서 제안하는 부분이 바로 에러가 존재하지 않을 때 제안하는 시나리오 데로 적용하여 에러를 체크하겠다는 것이다. 즉 에러 테스트 케이스를 거쳐 의도적으로 야기되는 프로그래머 정의 예외 처리를 하는 것이다.

4.4 테스트 데이터 생성

1. 시나리오



[그림 5] 테스트 데이터 생성 과정

기본적으로 APDU Protocol은 ISO 7816-4에 정의해 놓았다. 아래 표는 Command APDU 구조와 Response APDU 구조를 다음 표 3과 4에 나타내었다.

Mandatory header				Optional body		
CLA	INS	P1	P2	Lc	Data field	Le

[표 3] Command APDU 구조

표 3은 APDU 프로토콜은 호스트에서 카드 쪽으로 보내는 command의 APDU의 구조를 보여주었고, 표 4에서는 반대로 카드에서 호스트 쪽으로 보내는 요청 APDU 구조를 표로 보여주고 있다.

Optionalbody	Mandatory Trailer	
Data field	SW1	SW2

[표 4] Response APDU 구조

시나리오1 테스트 데이터 생성 방법은 위의 표 3, 4에서 정의해놓은 구조 이외의 데이터를 입력 함으로써 나타나는 값과 기대 결과치를 갖고 평가하고자 한다. 이러한 데이터 생성 방법은 표 5로 정리해 놓았다.

CASE		기대 결과
CASE1	APDU header 바이트가 올바른지 확인	Ncd, nrd
CASE2	Le 필드가 데이터 필드 길이와 맞는지 확인	Ncd, srd
CASE3	Lc 필드가 데이터 필드 길이와 맞는지 확인	Rcd, nrd
CASE4	Le, Lc 필드가 데이터 필드 길이와 맞는지 확인	Rcd, srd

[표 5] 테스트 데이터 생성 방법

위 표 5에 나타난 기대 결과와 테스트된 데이터 결과 값을 갖고 기대 되어진 결과와 실제 결과 값이 같게 측정될 경우 그 애플릿의 에러 처리가 Pass, Fail 이냐를 평가하여 향후 신뢰성이라는 품질항목을 평가 할 수 있을 것이라 생각된다.

2. 시나리오

· 일반적인 카드 ExceptionsHandler를 이용 [표 6]

CASE	기대 결과
CardException	· Checked · Unchecked
RuntimeException	· Checked · Unchecked

표 6은 표 5와 같이 본 연구에서 예외처리의 방법으로 보여준 표로서 일반적으로 지원되는 카드 예외처리를 갖고 이 예외처리를 거쳐 기대되는 결과로는 단순히 예외를 처리 Check, Uncheck으로 구별하여 위에서 소개한 방법으로 평가 할 수 있다.

5. 결론 및 향후 연구 방향

본 연구에서는 자바 카드 API의 품질 평가를 위한 테스트 데이터 생성 방법을 2가지의 시나리오로 제시하였다. 이 시나리오를 기반으로 테스트 데이터를 생성하여 생성된 데이터를 가지고 자바 카드 API의 신뢰성을 평가할 수 있는 지침을 만들 수 있다. 또한 향후 연구 방향은 실제 앞으로 사용되어지거나, 사용되고있는 자바 카드 애플릿을 가지고 이러한 테스트 데이터를 가지고 신뢰성을 평가하고자 하고. 수학적인 기반을 두고 있는 정형 기법(Formal Method)을 이용하여 애플릿 소스가 아닌, 요구 사항 명세를 입력으로 하여 처음 요구 사항 명세 데로 설계되고, 구현되었는지를 좀더 수학적으로 증명하고자 한다.

참고 문헌

[1] 문상재 외," 차세대 IC 카드를 사용한 정보보호 신기술 시스템 개발", 정보통신부 1997. pp.17.

[2] 김연선, 이창욱, "자바 카드 애플릿 설계 및 검증에 관한 연구" 「한국통신정보보호학회 종합학술발표회 논문집」 2000. Vol.10 No.1 pp.805.

[3] Chen, Zhiqun. " Java Card Technology for Smart Cards", ADDISON-WESLEY Company, 2000 pp.9.

[4] Jack W. Davidson and Cristopher W. Fraser, Automatic Generation of Peephole Optimizations, Proceedings of the ACM SIGPLAN '84 Symposium ojn the Compiler Construction SIGPLAN Notices, Vol.19, No.6, pp.111-116, 1984

[5] A. Menezes, P.van Oorschot, S.Vanstone, "Handbook of Applied Cryptography", CRC Press, 1996. pp.614

[6] Sun Microsystems, Inc. Java Card 2.1 Virtual Machine, Run Time Environment, and Application Programming Interface Specification, Public Review ed.,February 1999. Information available at http://java.sun.com/products/javacard2.1.html

[7]http:java.sun.com/products/javacard/datasheet.html