情 報 メ デ ィ ア 36－7
電子化知的財産・社会基盤 7－7
（2000. 1. 29）

Java Card™ – An Access to the Internet World

Yibin XU* Shigeki YOKOI* Takami YASUDA**
*Graduate School of Human Informatics, Nagoya University
** School of Informatics and Science, Nagoya University

In this paper, we have modelized the smart card application management system (SCAMS) now in use, and proposed a new application management system for Internet applications – cardholder-managed SCAMS. For implementing the cardholder-managed SCAMS, we have designed a system scheme of Java card solution. The processes of application development, system construction, and access to the Internet service have been described.

1. Introduction

A smart card, a potable personal information carrier with processing power, has been expected to facilitate an access to the multiple Internet services through multiple devices. For example, the smart card may be used to store an accessory key to an Internet digital content[1]. A travel agency may issue a airline ticket and hotel coupon with a smart card through the Internet[2]. While, the rapid growth of Internet services requires smart card solution with openness, flexibility, and quick response to the changing needs day to day.

Java™ technology – a network oriented technology, with its capability of "Write Once, Run Anywhere", has been proved invaluable in opening new business opportunities. Today Java is everywhere, from the smallest device to a super computer and from mobile phone to satellite television decoder. The new member in Java family -- Java Card™, with its wide range of advantages such as application platform independent, and multi-application available, has been expected to resolve the main problems in today's smart cards: lack of application interoperability and functional flexibility.

Addressing the smart card's functional flexibility, we modelize the smart card application management systems now in use, and propose a new application management system. To implement the new application management system, we provide a plot of application system basing on Java Card. The paper[3] of Sakai about Java Card described the characteristics of the card, but there was no mention on the application system of Java Card. In this paper, for the first time, we present a system scheme of an end-to-end Java Card solution. The technical processes such as application development, system construction, and access to the Internet service are described.

2. Smart card application management systems

A smart card is a card that stores and processes information through the electronic circuits embedded in the plastic substrate of its body. Similar to a computer, a smart card contains a CPU, ROM, RAM, EEPROM, I/O port and possibly a coprocessor[4]. All applications of smart card are executed by a smart card solution, which in minimum consists of a card and a card terminal. A more complicated card solution includes networks and servers.

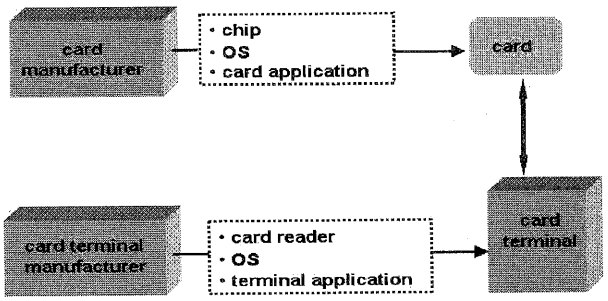

Figure 1 The model of card manufacturer-managed SCAMS.

We distinguish the applications running on the card, the card terminal and the server as card application, terminal application and server application, respectively.

The flexibility in smart card's function requires the applications on the card and terminal can be updated quickly and conveniently. The whole processes of application issuance (developing, installing, and updating) are managed by a smart card application management system (SCAMS). According to the manager of the SACMS, we conclude smart card solutions now in use into two models: manufacturers-managed SCAMS and card issuer-managed SCAMS. Here, the manufactures include the card manufacturer and the card terminal manufacturer; the card issuer refers to the organization (finical institute, enterprise, etc.) which manage the initialization and personalization of the smart card.

2.1 Manufacturers-managed SCAMS

The first generation smart card is specific function card. In a specific function card, applications and OS are microprocessor dependent. Applications and OS are in ROM., so the card's function is static. Because the card function is determined when the card is manufactured, and can not be changed after, the manufacturer is the manager of the SCAMS. The model of manufacturers-managed SCAMS is shown in figure 1. Consequently, the card terminal is also specially designed and produced for this type of cards. In such a system, there is no flexibility in function of the card.

2.2 Card issuer-managed SCAMS

The second generation smart card is multi-function card. For a multi-function card, the applications and data are stored in EEPROM, so can be added or deleted after the card is produced. There are several companies that develop and market multi-function card operating systems, for example, MultiFunction Card (IBM), Gemplus multi-application microprocessor card (Gemplus), SmarTB family (Bull) and Multipurpose Microprocessor Cards (De Ra lue).

A Java Card is a smart card that can execute programs written in JavaCard™ APIs. The Java Card contains a Java Card Runtime Environments (JCREs), which include a Java Card Virtual Machine and JavaCard APIs[5]. The Java Card applications, which is called card applets, are stored in EEPROM, and can be installed or updated after the card is issued.

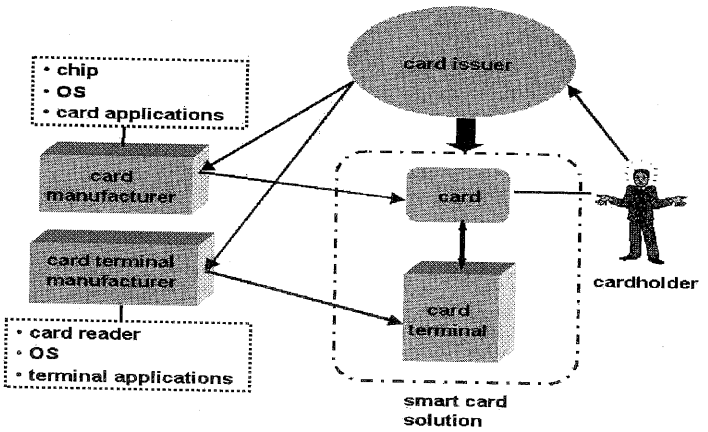

Figure 2 The model of card issuer-managed SCAMS.

The multi-function card and Java Card have given card issuers a chance to arrange content of a card after the card is produced, even after the card is issued. The model of card issuer-managed system is shown in figure 2.

In this system, the card issuer plays a central role. The card manufacturer, card terminal manufacturer and application developer produce the card, terminal and applications under the request of the card issuer. According to the needs of a cardholder, the card issuer arranges content of a card and issues the card. After the card is issued, the card issuer manages the card content in the whole life of the card (update or install new applications). And the card issuer is also responsible to establish card terminals for using their card. Comparing with the manufacturers-managed SCAMS, this system gives smart cards much more flexibility in function. Via card issuers, the function of the card can be changed without making a new card.

But there are still some problems with the card issuer-managed SCAMS: The first, the choices of card function are limited --the applications that can be installed on one card are determined by the card issuer. The second, the card terminal function is statistic. The third, different issuers' cards and card terminals are incompliant, so the cardholder has to find an appropriate terminal to use his card..

2.3 Cardholder-managed SCAMS

2.3.1 Conception of the Cardholder-managed SCAMS

Hereby we propose a conception of the cardholder-managed card system, which enables full flexibility in function of cards and card terminals. The model of the system is shown in figure 3. In this system, the applications for the card and the terminal are put on a web server. A cardholder downloads and installs the card application that he is interested in to the card by himself. The terminal application can be dynamically downloaded and run by an Internet connected terminal with a Java Virtual Machine (JVM), like a web browser.

2.3.2 The advantages of the new SCAMS

Comparing with the two card application management systems now in use, the cardholder-managed SCAMS has great flexibility and interoperability:

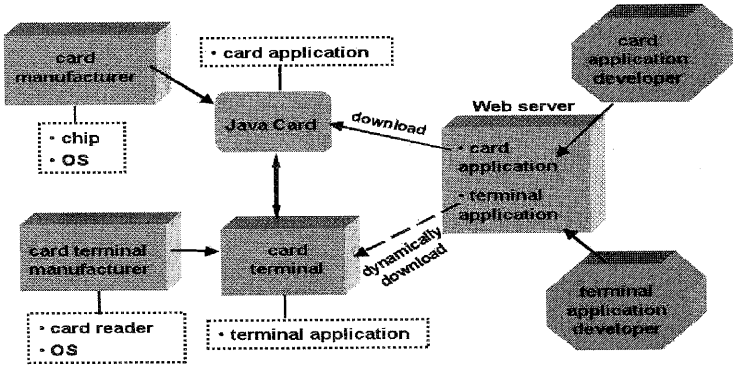

Figure 3 The model of cardholder-managed SCAMS.

(1) Card function flexibility – The cardholder can install any application to the card by himself. A new card service may be started with just putting a new program on the web server.

(2) Card terminal function flexibility –There is no limitation on the function of the terminal. It is not necessary to make any modification to the terminals when a new service starts.

(3) The cardholder can access the card service at any time from any locations by any Internet accessible card terminal.

(4) Compliant with the present SCAMSs – The card can carry three types of applications at the same time: embedded in factory, installed by a card issuer and downloaded from the Internet. The card terminal can run the applications installed or downloaded from the Internet.

3. Java Card solution for cardholder-managed SCAMS

In this section, we propose a scheme of Java Card solution to implement the cardholder-managed SCAMS. It is important to understand that in our system the introduction of Java does not only concern cards but all the elements of the scheme: the card, card terminal and server.

3.1 System scheme

As shown in figure 4, at the side of server, an Internet service provider put card applets and terminal applications written in Java on their Web server. A servlet is responsible for information transformation between the server and client computers. At the back of the web server, there is an application server, which is in charge of security control, administration, communication with the database, etc. At the side of client, the cardholder needs a Java Card and a computer with a card reader peripheral. In his computer, the following softwares are installed: Web browser, Java plug-in, tools for card applet installation and management, and a system that supports the terminal application to communicate with the card reader and card.

3.2 Application identification

Each smart card application is identified with an unique Application Identifier (AID) specified by ISO 7816[6]. AID is used when terminal application selects the corresponding card application to work with.

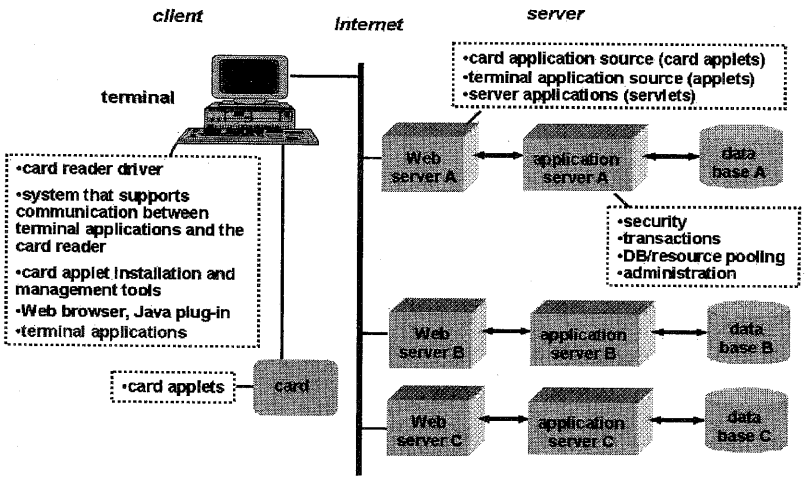

Figure 4 The scheme of a Java Card solution to implement the cardholder-managed SCAMS.

While, for the application downloaded from the Internet, the present AID system meets two main problems: the first, the card application has too much opportunity to repeat in AID. The second, it is necessary to indicate the location of the terminal application, because the terminal application does not resident in the terminal, it must be found through the Internet every time when it is executed.

Our suggestion to resolve this problem is to identify the card application with an expanded AID, which is consists of two parts: namespace part and local AID. The namespace part may be the URL of the Web page from where the card application is downloaded, or a specifically defined namespace, for example the namespace defined by a company for all of their smart card applications. Further more, we suggest that the terminal applications should also be identified in a similar way.

4. Java Card application server

4.1 Application development

Application development for the Java Card solution includes card application development, terminal application development and server application development. Since the server application does not directly related with the card, it is not included in this paper.

4.1.1 Card Applet development

To develop a card applet, a JavaCard Development Kit (the most recently released version is Sun's JavaCard 2.1), a JavaCard Converter, and an Integrated Development Environment (IDE) (for example, Java Development Kit), are necessary. The card applet development process[7] is shown in Figure 5. A developer writes one or more Java classes with JavaCard APIs, and compiles the source code with a Java compiler, producing one or more *class* files. The applet is run, tested and debugged on a workstation or personal computer using simulation tools to emulate the device environment. Then when the applet is ready to be downloaded to a card, the *class* files comprising the applet are converted to a *CAP* (converted applet) file using a JavaCard Converter, then be put on the web server.

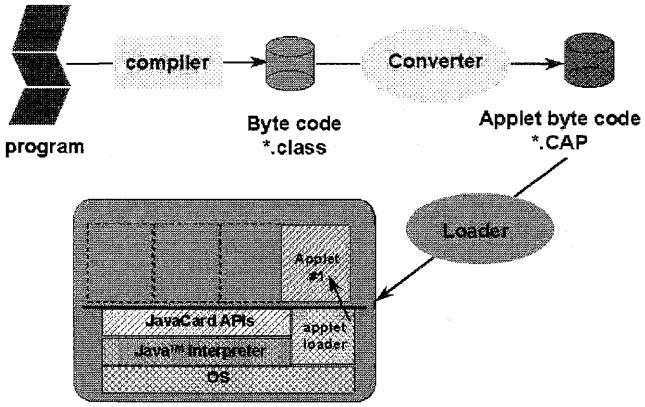

Figure 5. The card applet developing and installing process.

Many Java Card manufacturers provide application development kit for their card. For example Bull's OdesseyLab, Gemplus's GemXpresso RAD and Schlumbeger's Cyberflex Development Kit. It is convenient to use the development kit for testing the card applet on card.

4.1.2 Terminal application development

To implement the card terminal application, we need a system, which sits between the terminal application and the card, can handle and communicate with the card reader and the card. Typically, there are three such systems, PC/SC[8], OpenCard Framework (OCF)[9] and Visa Open Platform[10], which work on different terminal platform and in different environments. As the terminal application considered in this paper is written in Java and works on a Web browser, the OpenCard Framework is used in our system.

OCF communicates the card reader and the card through its *CardTerminal* layer and *CardService* layer[9]. The *CardTerminal* layer contains classes and interfaces that allow the terminal application developer to access card

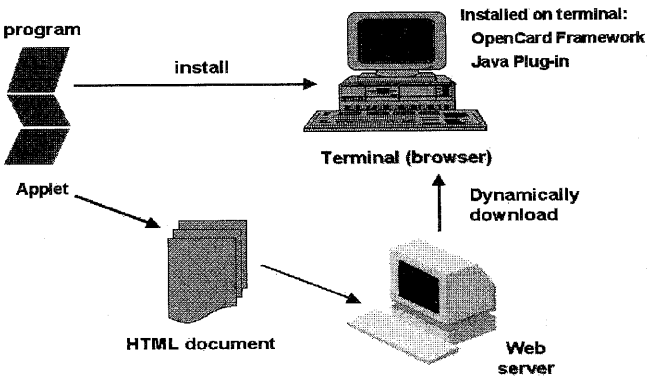

Figure 6. Terminal application development and installation process with OCF.

terminals and their slots. The *CardService* layer defines the abstract **CardService** class. Smart-card manufacturers and issuers have to provide **CardService** classes encapsulating the behavior of their smart cards and a **CardServiceFactory** class.

The application is developed and tested in a similar way of other Java applications. It is digitally signed and written in an HTML document with special tags required for Java Plug-in. Then the HTML document is put on a Web server (see figure 6).

4.2 Connection and download the Java Card application

The cardholder may find the card application from the Web page. If he is interested in it, he may simply download it from the Web page. Because the Java Card is not able to load class dynamically, the card applet must be installed to the card with the installation tool in his computer.

At first, the *CAP* file is downloaded to the cardholder's computer. Then the installation tool loads the *CAP* file and transmits it to the Java Card. An installation program on the card receives the contents of the *CAP* file and prepares the applet to be run by the Java Card virtual machine (see figure 5). We can use the application development kit provided by the card manufacturer or OCF to install the card applet.

The terminal application may be installed to the cardholder's computer, or may be dynamically downloaded from the Internet by a Web browser.

4.3 Access to the Java site

Once the card applet has been installed on his card, the cardholder may access the service site from any other computers with a card reader peripheral, and with a Web browser and Java plug-in installed. What he needs to do are: starting the Web browser, going to the Web site, and running the Java application to operate his card.

5. Java Card -- a window to the Java world

The above Java Card solution is based on a computer card terminal. Recently, Java Card technology has been adopted into the industry-leading worldwide standard for mobile phones: Global System for Mobile Communications (GSM)[11]. In Japan, the next generation mobile communication system IMT-2000 has announced to introduce Java Card into their system. These non-compute devices are characterized by severe restrictions on processing power and available memory. The key technology to build Java solution with non-computer card terminals is that we need an appropriate Java Virtual Machine for each of the terminals.

As a complementary technology to the Java 2 Platform Micro Edition, Java Card technology makes it easy to integrate essential consumer technology into a complete Java software solution. The application is written in the Java language and thus use Java objects to represent, store and manipulate data. The Java Card technology maintains the qualities that Java technology has become famous for: built-in consistency over any Java Cards; portability of the code and safe network delivery. However, at present, because the Java Card Runtime Environments (JCREs) are specially designed for the Java Card only, the card applet can not

be executed in the other Java Runtime Environments (JREs), such as a Web browser, or a computer. In the future, it is possible that the progresses in card chip will enable an advanced JCREs, in which card applets are upwardly scalable to work with the other Java editions. Therefore, the Java Card applet may be portable across all Java platforms. And Java objects on a card may synchronize with the Java objects on a computer or server.

Due to limitations in resource of memory and CPU, today the main functions of card applets are information storage, security control and some simple applications. The Java Card is viewed as an access key to the Internet service. However, the progresses in semiconductor, electronic and other technologies will enable a card to perform more complicated functions in the future. The Internet Java service might be simply downloaded and customized into a private service installed on a Java Card. Someday, what the Java Card will carry is not only information, but also customized service. Today the Java Card is you airline ticket, tomorrow the Java Card will become your private travel agency. The Java Card will become a window in your pocket to the Java world.

References

[1] 鈴木裕利, 横井茂樹, 安田孝美: 実用化が進む電子的著作権管理システム, 情報管理, Vol.42, No.7, pp.571-581, (1999).

[2] Arthur Coleman: Giving currency to the Java Card API, JavaWorld, February, (1998). (http://www.javaworld.com/javaworld/jw-02-1998/jw-02-javacard.html)

[3] 酒井高彦: 次世代ICカード革命, Java Card™, 情報処理, Vol.40, No.9, pp. 878-881, (1999).

[4] Jorge Ferrari, Robert Machinnon, Susan Poh and Laskhman Yatawara: Smart Cards: A Case Study, International Technical Support Organization, SG24-5239-00, (C) IBM Corp., (1998). (http://publib.boulder.ibm.com/cgi-bin/bookmgr/BOOKS/SG245239)

[5] "Java Card™ 2.1 Runtime Environment (JCRE) Specification", Sun Microsystems, Inc., Final Revision 1.0, February 24, 1999, (1999).

[6] ISO/ISC 7816-4 (1995); ISO/IEC 7816-4:1995/Amd 1 (1997).

[7] "Java Card™ 2.1 Virtual Machine Specification", Sun Microsystems, Inc, Revision 1.0, March, (1999); "Java Card Applet Developer's Guide", Sun Microsystems, Inc, Revision 1.12, August, (1998).

[8] "Interoperability Specification for ICCs and Personal Computer Systems", Bull CP8, Gemplus SA, Hewlett-Packard Company, IBM Corporation, Microsoft Corporation, Schlumberger SA, Siemens Nixdorf Informationssysteme AG, Sun Microsystems, Inc., Toshiba Corporation and VeriFone, Inc., Revision 1.0, December 1997 (1997).

[9] "OpenCard Framework 1.1.1 Programmer's Guide", OpenCard Consortium, Third Edition, April 1999 (1999).

[10] "Visa Open Platform Overview", Visa International Service Association, April 1999, (1999).

[11] "Sun Microsystems' Java Card™ technology available as GSM standard in mobile phones", CARTES `99, Paris, France, November 16, 1999 (1999).