Computer Languages, Systems & Structures 40 (2014) 53–72

Contents lists available at ScienceDirect

Computer Languages, Systems & Structures

journal homepage: www.elsevier.com/locate/cl

A model driven architecture for the development of smart card software

Hidayet Burak Saritas, Geylani Kardas*

Ege University International Computer Institute, 35100, Bornova, Izmir, Turkey

ARTICLE INFO

Article history:
Received 15 July 2013
Received in revised form
10 December 2013
Accepted 11 February 2014
Available online 17 February 2014

Keywords:
Model driven architecture
Metamodel
Model transformation
Smart card
Java Card
Basic card

ABSTRACT

Smart cards are portable integrated devices that store and process data. Speed, security and portability properties enable smart cards to have a widespread usage in various fields including telecommunication, transportation and the credit card industry. However, the development of smart card applications is a difficult task due to hardware and software constraints. The necessity of the knowledge of both a very low-level communication protocol and a specific hardware causes smart card software development to be a big challenge for the developers. Written codes tend to be error-prone and hard to debug because of the limited memory resources. Hence, in this study, we introduce a model driven architecture which aims to facilitate smart card software development by both providing an easy design of smart card systems and automatic generation of the required smart card software from the system models. Differentiating from the previous work, the study in here contributes to the field by both providing various smart card metamodels in different abstraction layers and defines model-to-model transformations between the instances of these metamodels in order to support the realization of the same system on different smart card platforms. Applicability of the proposed methodology is shown for rapid and efficient application development in two major smart card frameworks: Java Card and ZeitControl Basic Card. Lessons learned during the industrial usage of the architecture are also reported in the paper. Finally, we discuss how the components of the architecture can be integrated in order to provide a domain-specific language for smart card software.

1. Introduction

Smart cards are portable, integrated circuit devices that securely store and process data [1]. These tiny computers with their own memories and processors have a widespread usage in various fields including telecommunication, transportation, banking and healthcare. For instance the operation of a cellular phone is directly based on a smart card which carries an identification number unique to the owner, stores personal data and prevents operation if removed. Also most of today's credit cards are in fact smart cards that store account information for a bank customer and provide authorization and authentication for electronic money transfers.

* Correspondence to: Ege Universitesi Uluslararasi Bilgisayar Enstitusu, Bornova Metro Duragi Karsisi, 35100, Bornova, Izmir Turkey.
Tel.: +90 232 3113223; fax: +90 232 3887230.
E-mail addresses: saritasburak@gmail.com (H.B. Saritas), geylani.kardas@ege.edu.tr, geylani@gmail.com (G. Kardas).

http://dx.doi.org/10.1016/j.cl.2014.02.001
1477-8424

Since hardware and software capabilities of a smart card are very limited compared to a desktop personal computer, the development of smart card applications is a difficult task. The necessity of having a deep knowledge of both a very low-level smart card communication protocol [2] and a specific hardware causes smart card software development to be a big challenge for the developers. Limited memory resources force developers to deal with very primitive data structures. The design of the software for the process of incoming hexadecimal data packages and preparation of the outgoing data packages byte-by-byte, again in hexadecimal format, are difficult and time-consuming jobs for the developers. Development environments dedicated to the smart cards are mostly incapable of code debugging and this makes written codes tend to be error-prone.

Like other smart card developers, we also experienced above difficulties in the design and implementation of various smart card software which were developed during our academic research (e.g. [3–5]) or for commercial purposes. Based on our experience, we may conclude that working in a higher abstraction level different from the code level is mandatory for an efficient design and implementation of smart card software systems. Within this context, model driven engineering (MDE) (or model driven development (MDD)) [6], which aims to change the focus of software development from code to models, may also provide easy and efficient production of smart card software. In such an MDE environment, card developers can graphically design their system models conforming to metamodel(s) at various abstraction layers which in fact present entities and their relations needed for a smart card system, and then software codes needed for the designed system are automatically generated as the result of a model to text transformation. Hence, in this paper, we introduce a MDD process which aims to facilitate the smart card software development by both providing an easy design of smart card systems and automatic generation of smart card programs. We use model driven architecture (MDA) ([7,8]) which is one of the realizations of MDD to support the relations between platform independent and various platform dependent smart card entities to develop software for smart cards.

Following the derivation of smart card metamodels in different abstraction layers, definition and implementation of model-to-model (M2M) transformations for the instances of these derived metamodels were performed. Finally, model-to-text (M2T) transformations were provided to automatically generate required software codes.

Apart from the related work (e.g. [9–12]) which mainly considers model driven smart card software development only specific for the Java Card framework [13], the study presented in this paper introduces a platform independent metamodel (PIMM) for smart card systems in order to free the developer from taking into consideration the specific needs of different smart card platforms such as Java Card. Furthermore, the proposed PIMM enables the developer to model the smart card system conforming to data transmission and data storage standards brought by ISO/IEC 7816 standards family [2]. This standards family includes various standards for smart card development such as the physical characteristics, electrical interface, transmission between smart cards and host devices and personal verification.

On the other hand, transformability from the general smart card PIMM to the dedicated smart card platforms is also presented in this study. Metamodels of two major smart card frameworks, Java Card [13] and ZeitControl Basic Card [14], are defined as the platform specific smart card metamodels (PSMM). Hence, applied M2M transformations and following M2T transformations provide the implementation of the same smart card system on different execution platforms.

The rest of the paper is organized as follows: a brief discussion on smart card technology is given in Section 2. Section 3 includes the proposed smart card PIMM and related modeling environment. Platform specific metamodels and modeling tools for the MDD of Java Card and Basic Card applications are given in Sections 4 and 5 respectively. Defined model transformations between general smart card models and Java Card and Basic Card models are discussed in Section 6. Section 7 covers the automatic code generation from platform specific card instance models. Evaluation of the study by considering the lessons learned during the industrial usage of the architecture is reported in Section 8. Related work is given in Section 9. Section 10 concludes the paper.

2. Smart card technology

Smart cards are tiny computers with their own processor and memory. For instance, the integrated circuit on a bank credit card is in fact a smart card. Also, subscriber identity modules (SIM), used inside our cellular phones, are smart cards that store subscriber information to use the phones properly. A smart card includes a micro-processor, read-only memory (ROM), random access memory (RAM) and electrical erasable programmable ROM (EEPROM). Operating system of the card is stored in ROM. Similar to the main memories of our desktop PCs, applications run on RAM. Finally, EEPROM stores applications and data while the card is unpowered. Connection points on the card support the input/output and communication with the host computers [1,15]. Fundamental functionality (e.g. commands for interchange, structure of transmitted data packages) and characteristics (such as electrical interface or contact type) for smart cards are defined with ISO/IEC 7816 standards [2].

Depending on the usage type, smart cards may be classified as *contact* or *contacless*. Contact cards should be inserted in a card reader which is directly connected to a host computer whereas contactless cards do not need to be inserted or contacted physically for operation. Such kind of cards communicates with and is powered by the reader through Radio Frequency (RF) induction technology.

Like Transmission Control Protocol/Internet Protocol (TCP/IP) and related data package transmission used in computer networks, there also exists a standard communication infrastructure defined for smart cards again in ISO/IEC 7816 Standards family. Whole communication between a smart card and a host computer (terminal) is provided by the exchange of