Verification of the Environment Information System Based on Model Checking

Yongbo Wang [1], Wei Zhang[2,] *
[1] *Tai'an Local Taxation Bureau, DongYu Street, Tai'an, P.R. China*
[2] *Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputer Center in Jinan), Xinluo Ave., Jinan, P.R. China*
[2] *College of Information Science and Engineering, Shandong University of Science and Technology, Qianwangang Ave., Qingdao, P.R. China*
**Corresponding Author: wzhang@sdas.org*

Keywords: environment information systems, model checking, linear temporal logic

Abstract: With the development of 4G and 5G mobile communication technology, using mobile phone to pay for goods and services has become a very popular application. The traditional forms of payment cannot be applied in e-commerce environment. This paper employs model checking method to verify the security and reliability of the Environment Information Systems. A PROMELA model for the System is present. As an important part of the modeling methodology, the Environment Information System is translated into a simpler model that nevertheless preserves all the essential behavior to be verified. It also proposes initial results on the actual verification of the Environment Information System using SPIN. The result of this work is a complete procedure for the modeling and verification of the Environment Information System.

1. INTRODUCTION

In recent years with the rapid development of 4G and 5G mobile communication technology, making the 4G/5G era has quietly penetrated into people's lives and work in various fields. In the traditional business activities, the payment process is mainly classified in paper forms, such as bill payments and cash payments. The traditional forms of payment cannot be applied to the e-commerce environment. The reasons are as following: the traditional payment cannot be binding and monitor between the participants of the transaction. Quality of the goods, transaction integrity, and requirements of return and replacement cannot be reliable guarantee. The Financial Institutions focuses these days to move all payment forms (i.e. transfers, deals, purchases, and bill payments) to electronic form instead of paper form. Recently the mobile has become an essential tool for commerce and financial services. With the help of new communication and information technologies, these services have experienced tremendous growth.

It is convenient for people to use Environment Information System s in transactions. Mobile phone payment application mode is not a single technology, which requires more extensive and powerful system to do the support. In addition, it also need to telecom operators, commercial banks and card associations, third-party service providers to jointly build a Environment Information System. Mentioned above, the mobile phone payment system, in the final analysis, are based on the mobile phone payment platform. Because the Mobile Payments are related to both capital flows and goods flows, higher security and reliability is required for the transaction process. In mobile payments, participants may use communication protocols for which there are no transactional variants and the programs may be deployed in very heterogeneous application environments. For these reasons, Environment Information System s cannot rely on traditional transaction mechanisms [1]. The research on mobile payment agreements has been the focus of financial payment system in recent years. But there are not many related works to verify the logic and design of business processes during the mobile payment to ensure the safety and reliability of the systems. This paper discusses how to employ SPIN [5, 6, 7], one of the most powerful and well-known model checking tools, in order to specify and analyze the correctness of protocols for Environment Information System s.

2. BACKGROUND

As theoretic background, we introduce some frameworks of the Environment Information System s. Mobile payment is a payment mode through the mobile device (which can be SD cards, foil cards Or SIM card, etc.), in which the bank card information is stored in, and use of mobile phones or radio frequency wireless communication technology to achieve the remote technical paid. There are many types of the mobile payment products, such as GPRS /Client payment mode, STK/SMS payment mode.

In this paper, we focus on GPRS /Client payment mode, which are widely used in the Environment Information System s.

One whole Environment Information System is composed by a number of transactions. Figure 1 shows the Account Registration transaction of one Environment Information System. This transaction generally involves three participants: Users (USER), MPS (Environment Information System) and the bank (BANK). The general process is as follows:

(1) USER sends a request to register an account to MPS;

(2) MPS determine whether the user has an existing

account;

(3) If the USER already has an existing account then returns it to USER;

(4) If the USER has not an existing account; then MPS sends the Verification request to BANK;

(5) If Verification success, BANK sends the message to MPS, and MPS send Registration success message to USER.

(6) USER sends the transaction successful message to BANK (transaction successful);

(7) If Verification is unsuccessful, BANK sends the message to MPS, and MPS send Registration unsuccessful message to USER (transaction failed).

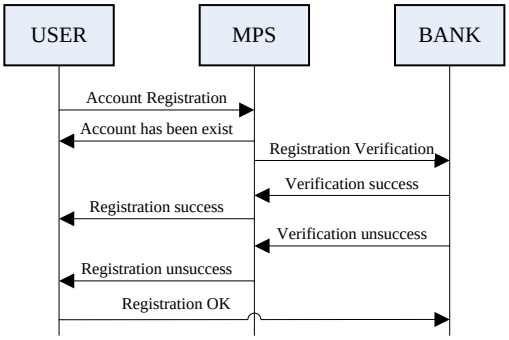

Figure 1. Account Registration in Environment Information System s

There are many transactions in a Environment Information System. In this paper we focus on the core transaction which is payment process. The detail of the core transaction will be discussed in Section 3.

3. MODELING THE ENVIRONMENT INFORMATION SYSTEM

In this section we describe the core transaction of the Environment Information System s. Formal modeling is the first and crucial step in model checking. In the formal modeling process, we should ignore the participants which are independent of the desired characteristics of the system. In order to get an accurate model of the Environment Information System, here we only pay attention to the three objects in the core transaction, the users (USER), the shops (SHOP) and the Environment Information System (MPS). We use the notation "A⇒B Message" to indicate that A sends the message to B. The basic protocol consists of the following messages:

(1) USER⇒SHOP Buy
(2) SHOP ⇒ USER BuyOK or BuyNOK
(3) USER⇒MPS Pay or Deny
(4) MPS⇒SHOP Pay or Deny
(5) SHOP ⇒ MPS Sent
(6) MPS⇒USER Sent
(7) USER⇒MPS Confirm or Back
(8) MPS⇒SHOP Moneytos
(9) MPS⇒SHOP Back
(10) SHOP ⇒ MPS Backconf
(11) MPS⇒USER Moneybackc

The massage flow is shown in Figure 2 below.

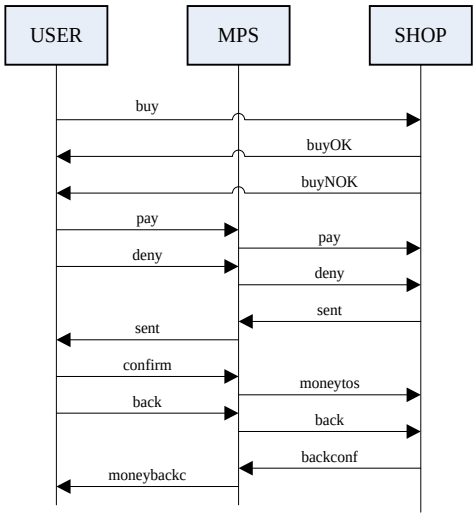

Figure 2. Massage flow in core transaction of Environment Information System

Constructing a model for a protocol in PROMELA requires a previous abstraction process of the original source code. Usually, this process eliminates details that are not necessary for debugging purposes. Therefore, models will be as small as possible making sure that they represent the exact details needed for the properties to be analyzed.

Extended Finite State Machine (EFSM) has been the underlying model as formal description for the communications protocol. EFSM model is extended with the finite state machine (FSM) model. Compared with FSM, there are environmental variables and the migration of pre-conditions in EFSM. So EFSM model has a stronger ability to describe the dynamic behavior of the system. For these reasons, we use EFSM to model the process, which is formula in the area of model checking, and also can be described in PROMELA easily.

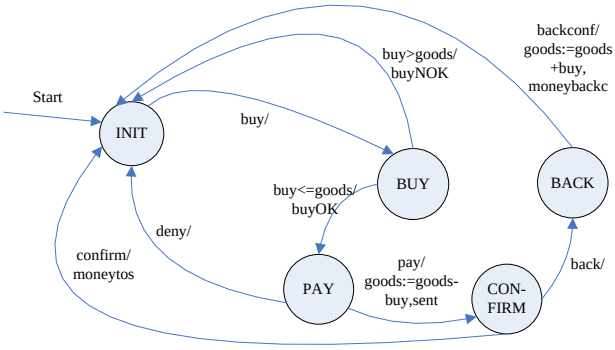

Figure 3. EFSM of the Environment Information Systems

Definition: EFSM M is defined as the tuple (S, s0, V, MV, P, MP, I, O, T), in which:

S is a finite set of states, s0 is the initial state, s0∈S;

V is the finite set of the internal variable (environment variable), and the range of the internal variable is DV;

MV is the set of the initial (or default) value of variables in V , in which any element can be expressed as a tuple (s, v), s∈S, v∈DV;

P is the input and output parameters;

MP is the set of the initial (or default) value of variables in P, in which any element can be expressed as a tuple (p, u), p∈I∪O, u∈Dp, Dp is the range of the input and output parameters;

I is a set of the input symbols;

O is a set of the output symbols;

T is a finite set of state transition.

A state transition t (t∈T) is defined as the tuple (s,x,y,gP,gE,op,e), where:

s and t are the start (head) state and the end (tail) state;

gP is the input and output conditions to determine;

gE is the conditions to determine of the variable required for migration;

x and y are the input and output symbols;

op is output operations.

Figure 3 shows the EFSM model of the Environment Information System, containing 5 states and 8 state transitions. The label "buy>goods/buyNOK" means that, when the input symbol of the state BUY satisfying "buy>goods", the state will be converted to the state INIT, and it will output the symbol buyNOK. The label " backconf/goods: =goods+buy,moneybackc " means that, when the input symbol is backconf, the state BACK will be converted to the state INIT, and it perform the operation "goods:=goods+buy" and output the symbol moneybackc.

4. DESCRIPTION OF THE MODEL AND PROPERTIES

In the last section, the core transaction of Environment Information System modeling in PROMELA has been completed. In the modeling process, we have done lots of abstraction to prevent the state explosion. The main purpose of this paper is to verify the process of mobile payment transactions for any errors, so the model does not reflect the identity, password authentication and other security aspects of the process, these parts will also be the focus of our future research. Although it has been abstracted, this model still can describe the actual process of the mobile payment transactions. The more details we will introduce are as following.

In the model, there are three main variables represent the different meanings. The variable i means the number of users. The variable buy means the number of goods bought by each user. And the variable $goods$ means the inventory of shop. The three variables given different initial values, there will be several different situations, which represent the different actual payment business.

There are two major types of cases of the variable i. When it is i<=1 in the loop, there is only one user process running. When i <= any number larger than 1, there will several concurrent user processes running together. If there are concurrent processes running, the SHOP process will control each user to turn correct state. Specific control process can be seen in the fragment of SHOP process, and the detail of the next states is shown in Figure 4.

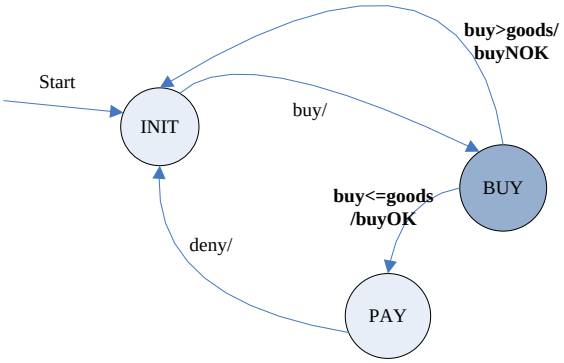

Figure 4 Different situations of the state BUY

In Figure 4, we will see that the other two variables buy and goods are compared in the process. Also it represents two different situations. If it is the situation of "buy<=goods", it means the user can buy the goods, as there are enough inventories of the shop. If it is the situation of "buy>goods", it means there are not enough inventories. The process will turn back to the state INIT. So, if these two variables are defined by the initial values to meet the condition "buy>goods", there will be some of the states that could never been reached. We will discuss these cases in more detail with experiments in next section.

As explained in Section 2, SPIN supports two kinds of analysis for the modeled protocols. The first one consists of checking deadlocks and other safety properties by generating the execution paths in the model. The second kind of analysis consists of checking temporal properties specified with temporal logic. Here we describe correctness criteria we are interested in, and show how they can be defined in SPIN. In order to formalize both desired and undesired properties of the Environment Information System s, we use LTL (linear temporal logic notation), which has been explained in Section 2. LTL allows expressing temporal properties we expect the system behavior will conform to during the system lifetime. Such properties can be seen as a part of requirement specification. Expression of properties in the formal LTL notation gives both an unambiguous presentation of expected system behavior and possibility to verify whether the system model conforms to the requirements. LTL formulae can express both safety and aliveness properties, and are effectively supported in SPIN.

Examples of typical requirements to the internet payment behavior can be formulated in plain English as following. For users, the users need the support of the payment is guaranteed:

(1) Before receiving confirmation, MPS will not pay the money to the shop;

(2) IF users are not satisfied with the goods, after the required return, the MPS will refund money to users;

For the shop, the business requires the support of the Environment Information System guaranteed: After confirmation of receipt in user, the MPS will pay the money to the shop. That requires the support of the system:

(3) When transaction succeeds, the users receive the goods, the shop get money from the MPS.

(4) When transaction fails, the goods are returned to the shop, and the money is back to the users from the MPS.

5. EXPERIMENTS

If a specification of model has been given in PROMELA, SPIN can search the whole state space of the model; it also can identify unreachable state or deadlock in model. In addition, SPIN can construct a verifier, which can check several claims on the execution of the model. We have established the model of the Environment Information System in PROMELA, and analyzed the properties in last sections. These properties include the values of certain variables at certain points in the code and true statements that can be made about execution states (state properties) or the paths of execution (path properties). In this section, we present various kinds of verification that can be performed on a PROMELA model described in the sections above. Using SPIN and the PROMELA specification presented above, several properties of the execution of the model were verified. These properties were verified as part of several experiments described below. For each experiment, the size of the model constructed by SPIN, the time for verification were measured. The experiments were carried out on a 2.0GHz Pentium duel machine with 2048MB of memory.

First we have let SPIN perform a full state space search for invalid end states, which is SPIN's formalization of deadlock states, in case of 5 users, buy=2, and goods=100. The results are shown in Figure 10. We have let SPIN perform a full state space search in cases of 1-5 users, and the variable buy=2, goods=100. The results are summarized in Table I. The exponential increase in number of states, memory usage and verification time, seems to be not manageable when checking more than 5 users. In case of 6 users, the available physical memory was insufficient.

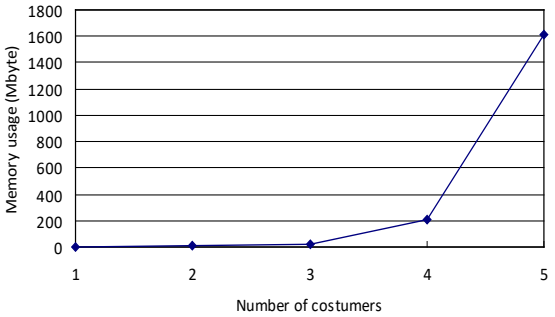

Figure 5. Memory usage of 1-5 users

Since checking more than two users is superfluous and violates the requirement that the verification model be the minimum sufficient model to perform the verification successfully. In Figure 5, we can see the memory usage is in the exponential growth with the number of users increasing. We do not gain in verification power by checking more than 5 users. In our future research we will search for a better modeling of loops that will minimize the state explosion that has been revealed by our experiments.

In the experiments, we define the variable as a special case, in which we define the variable buy=3 and goods=2. This means that in the initial state, the determination is consistent with "buy>goods". We have let SPIN perform a full state space search in this case. The results are shown in Figure 3. The meaning of the first few lines in the results has been introduced in above cases. We will find the differences mainly in "unreached in proctype". In other words, many states are unreachable in this case.

Because the two variables are defined by the initial values to meet the condition "buy>goods", all users will not succeed to buy the goods. As shown in Figure 4, all users will get the massage "buyNOK", which means they cannot buy the goods. So the situation "shop_user!buyOK" will never arise. All the users in the state BUY will turn back to the state INIT, and the states behind the state BUY would never been reached. In the experiments, we also have tried to simulate the actual situation as more as possible, and add the property which we will verify into the process. In the actual internet payment, the quantity of goods bought by each user is not the same. Although we cannot simulate the random number of goods bought by each user, we can define the variable buy=i, approximating the actual situation. Thus, the process can simulate different numbers of goods for each user. Several interesting liveness claims can be made about the Environment Information System. We have described the properties in LTL formula in last section. The experiment shows that the verification of the model of the Environment Information System that does not contain any loops can be done very effectively. The results of these experiments show that there is no error in the design of the Environment Information System.

If we have detected any situation which do not meet the properties in the other model of protocol, SPIN can display these paths. We call these paths as counter-examples. According to the counterexample generated by SPIN, developers have the opportunity to understand wrong business process behavior, to locate errors and to effect right changes for correcting business process design. Then, the modified design is again submitted to SPIN for verification. This methodology determines a gradual correction and refinement of business process models, before it is definitely implemented.

6. CONCLUSIONS

In this paper, we introduced a model checking approach to verify the Environment Information System s. First we analyzed the general Environment Information System s, including the internet payment protocol, the participants of the payment transactions, and the massage flow in the system. We proposed an EFSM model, and translated the model in PROMELA. We also summarized a set of LTL formulas that can guarantee the reliability of the transactions. And then we did some experiments, which can prove that our model can simulate the actual transactions. Also the initial results on the verification of the Environment Information System using SPIN were provided. Our

approach can be easily extended to support model checking debugging new design of the Environment Information System s using the SPIN tool. The designers may simulate their active applications with our method.

ACKNOWLEDGMENTS

This work was supported by Shandong Provincial Natural Science Foundation of China (Grant No. ZR2015YL019) and the National Natural Science Foundation of China (No. 61472230).

REFERENCES

[1] P. Katsaros, "A roadmap to electronic payment transaction guarantees and a Colored Petri Net model checking approach," Information and Software Technology, Elsevier, vol.51, pp. 235–257, 2009.

[2] M. M. Gallardo, J. Martinez and P. Merino, "Model checking active networks with SPIN," Computer Communications, Elsevier, vol.28, pp. 609–622, 2005.

[3] E.M. Clarke, E.A. Emerson and A.P. Sistla, "Automatic verification of finite-state concurrent systems using temporal logic specifications," ACM Trans. on Programming Languages and Systems, vol.8 (2), pp.244–263, Apr. 1986.

[4] E. Clarke, O. Grumberg and D. Peled, Model Checking, MIT Press, Cambridge, 2000.

[5] G.J. Holzmann, Design and Validation of Comp. Protocols, Prentice-Hall, Englewood Cliffs, NJ, 1991

[6] G.J. Holzmann, "The model checker SPIN," IEEE Transactions on SE, vol.23 (5), pp. 279–295, 1997.

[7] On-the-fly LTL model checking with SPIN. http://spinroot.com/spin/whatispin.html

[8] E. M. Clark and J. M. Wing, "Formal methods: State of the art and future directions," ACM Computing Surveys, Vol. 28(4), pp. 1–18, 1996.

[9] A. Cimatti, E. M. Clarke, E. Giunchiglia, et al. "NuSMV2: An OpenSource Tool for Symbolic Model Checking," Proceeding of International Conference on Computer-Aided Verification (CAV2002), Copenhagen, Denmark, July 2002.

[10] K. Havelund. "Java PathFinder: A Translator from Java to PROMELA," Proceedings of the 5th and 6th International SPIN Workshops on Theoretical and Practical Aspects of SPIN Model Checking, Springer-Verlag, pp. 152, 1999.

[11] M. Makela, "Maria: Modular Reachability Analyser for Algebraic System," Application and Theory of Petri Nets 2002, 23rd International Conference, ICATPN 2002, Vol. 2360, pp. 434–444, June 2002.

[12] G. J. Holzmann, The SPIN Model Checker: Primer and Reference Manual, Addison Wesley, 2004.

[13] H. Xu and Y-T. Cheng, "Model Checking Bidding Behaviors in Internet Concurrent Auctions," International Journal of Computer Systems Science & Engineering (IJCSSE), Vol. 22, No. 4, pp. 179–191, July 2007.

[14] Z. Manna and A. Pnueli, The Temporal Logic of Reactive and Concurrent Systems: Specification, Springer-Verlag, 1992.

[15] E. M. Clarke, O. Grumberg, and K. Hamaguchi, "Another Look at LTL Model Checking," Formal Methods in System Design, Vol. 10, pp. 47-71, February 1997.

[16] R. Shaikh and S. Devane, "Formal verification of payment protocol using AVISPA," International Journal for Infonomics, Vol.3, Issue 3, September 2010.

[17] H.M. Deitel, P.J. Deitel and T.R. Nieto, e-Business & e-Commerce: How to Program, Prentice Hall, 2001.