International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)
ISSN 2150-7988 Vol.1 (2009), pp. 76–82
http://www.mirlabs.org/ijcisim

Applying Model Checking Agent for Error Detecting in SMS using Formal Method

Siti Dianah Abdul Bujang and Ali Selamat
Department of Software Engineering
Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
anarel_03@yahoo.com, aselamat@utm.my

Abstract

Considering the current mobile driven evolution, entrepreneurs' are eager to increase the performance of their business services. They want an efficient system that able to manage the finances, transactions and communications in a dependable and secure way. Therefore, a method that avoids technical errors, software bugs and requirements misunderstanding is needed. This paper presents a model checking agent approach for error detection in SMS (Short Message Service) application using SPIN. We focus on the behavior and the property verification using model checking agent to support the quality of services for system performance. The motivations of this work are: (1) enhancing the architecture of distributed system into agent based and translating it into formal specification; (2) proving the generated specifications using XSpin model checker. The aim of this approach is to provide a satisfying and verified system design through the modeling of model checking agent with an existed case study.

1. Introduction

Currently, the revolution and globalization of business marketing and telecommunication industries are widespread with the use of mobile-driven applications. In telecommunications industries, Short Message Service (SMS) is one of the business services that offer cost effective, efficient, and fast performance customers. Lately, there are billions of messages that are being sent worldwide and the number is increasing every year [1]. The statistics in Figure 1 show that Malaysia has the third highest number of SMS traffic in Asia Pacific [2] in which 1.075 billion messages had been sent especially during Christmas and New Year event.

In fact, the popularity of SMS has made one of the demanding services in the business market in which it is able to maintain the data applications, e.g. for managing the commercial transaction, information retrieval, and remote payment [3]. Therefore, [4], [5] and [6] have established

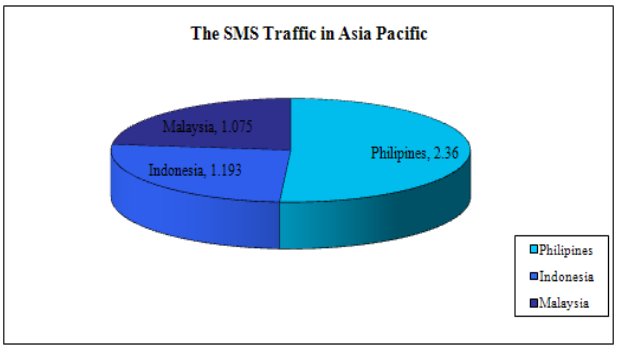

Figure 1. The SMS traffic in Asia Pacific from Bernama.

their conventional system for the mobile SMS application in order to increase the performance and services. However, due to the complexity of modern software systems, one of the major obstacles in developing dependable SMS systems is when the undesired errors arise in the distributed systems. Although the SMS is cost effective with high speed service, somehow the undesirable congestion can probably affect the system management performance [7]. To handle the complexity issue, agent based is the best solution to address the problem in mobile computing. The agent is autonomous to collaborate dynamically with each other in order to achieve goal without third parties enforcement [8]. However, the exception handling of the system should be proven by using the formal method mechanism to support the application validity. Therefore, the investigation should be done beyond the user capability in order to enhance the correctness of the systems. It is required to specify precisely the behavior and the properties of the system to ensure that the system behaves as it should be [9]. This paper proposes a model checking agent approach to address the verification issues in the operational service by selecting the mobile SMS application as a case study. We have been interested to focus on the behavior and the property verification of the model checking agent to support the high quality services

for end-to-end SMS system performance in the near future.

This paper is organized as follows. In Section 2, we give a short overview of the related works. Section 3 illustrates the architecture of the mobile SMS application case study including the proposed design. Then, Section 4 describes the verification strategy using the case-study as an example. Next, Section 5 presents the experimental result of the design and the discussion. Finally, the final remarks and future work conclude the paper in Section 6.

2. Related Works

Formal method in software engineering has become one of the important applications in the area of intelligent systems [10]. In fact, formal method has achieved excellent results in many years. In previous researches, [11] has presented his work that doing the automatic checking of the correctness of web applications structure by using the symbolic model checking. [12] has presented a model checking which is able to aid the design and assurance of e-business processes. [13] has applied the SPIN model checking to verify the clinical guidelines by adopting the agent based paradigm. [14] has proved that the analysis and verification using model checking are competent to support the security protocols. [15] has presented a program for NASA flight software using model checker Java PathFinder in the application of "Design for Verification" (D4V). In April 2008, [16] investigated the probabilistic of model checking tools and they found that the results are reliable. But then in October 2008, [17] proved that model checking software is the efficient technique to ensure the soundness of any type of system. Although all these efforts have significant contributions in their own right and their interests in this areas are remarkable, the verification of SMS mobile application is still remains undiscovered.

2.1. Agent Structure Paradigm

Agent application has evolved in many years and has been successfully applied in the area of mobile computing [8]. The numerous agents that interact to each other to achieve their aim goals are known as multi agent system (MAS). [18] has noticed that MAS enables predictable and reliable interactions amongst groups of agents without requiring a deep standardization MAS also does not have to follow up the third parties in order to force them to complete their tasks [8], [18]. MAS has sense of interaction between others because they have an internal data structure which is used to record the information regarding the environment state and history. These abstractions can lead to simpler techniques for design and development [19]. Their expertises are wide in various areas as they are heterogeneous to each other. They learn from experience and are self-motivated to fulfill their internal goals.

2.2. Formal Method with Model Checking

Modeling is the process of abstracting the functional specifications of a system into minimal specimen that enables designer to understand and analyze a particular aspect of the system more closely. The best technique to represent the functional specifications for correctness is by using formal method. Formal method is a mathematical based technique for specification, development and verification of hardware and software design. The use of formal method for software and hardware design is motivated by the expectation of other engineering disciplines. This technique is able to perform in an appropriate mathematical analysis which can contribute to the reliability and robustness of a design. In the development of high integrity system, formal method has significant role to represent the system specification. It is rapidly becoming a promising automated method to enhance the accuracy and correctness of the software systems. The verification is based on the static analysis that is able to analyze systematically and exhaustively [11]. The known formal method model checking is the currently evolving technology to evaluate the e-business process. Model checking provides an effective and efficient evaluation with counter-example to verify the correctness of complex software system [12]. This method can effectively represent systematically exhaustive mathematical models and infinite models to trace the errors. In verification using model checking, there are several model checkers can be used for testing our model. The model checker SPIN (Simple Promela Intepreter) (http://spinroot.com/spin/whatispin.html) can be written using Promela language and is the representative of LTL model checker for communication protocol or concurrent software. SPIN provides safety and liveness properties such as deadlock, invalid end state and non-progress cycle [20]. In fact, Spin is highly recommended for automization in many fields such as security protocol verification, control system verification, software verification and optimization schedule [14]. Others, [12] has identified that Verisoft (http://cm.bell-labs.com/who/god/verisoft) which was written in C and C++ can test hundreds of thousands of input combinations in a short time and analyze distributed systems.

3. Case Study Architecture

In order to illustrate our model checking agent for error detecting in SMS, we present a simple, yet practical case study, the SMS Management System. There are four main modules in the SMS Management System for direct sales and network marketing [5]. The first module is the registration, whereby customers register as distributors and after that the distributors will recruit new distributors as their sales force in the down line. The second module is the product selling. The third module is the bonus calculation which means the calculation of profit percentage in sales force of multi level marketing. The fourth module is group SMS that sends out news regarding activities and seminars to motivate and recruit as many distributors as possible to join the sales force. The architecture is shown in the following Figure. 2.

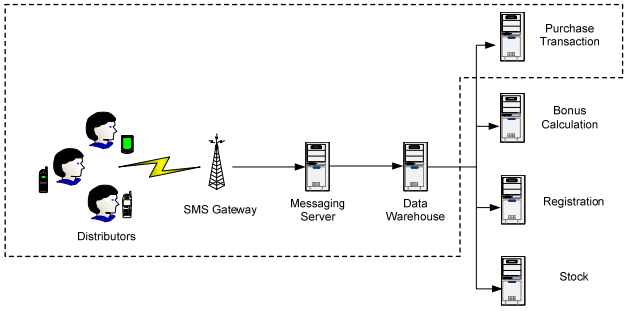

Figure 2. The architecture of SMS Management System for Direct Sales and Network Marketing.

3.1. Architecture of Model Checking Agent

We have enhanced the architecture in Figure 2 into an agent-based system. The architecture of our model is illustrated in Figure 3 and the roles of each agent are as follows:

- **Input Agent (Ia)**
  Input Agent represents the user interface in a mobile phone and used as a medium for users to interact with the SMS management system. The Ia will receive the request from user and assist the Model Checking Agent (MCa) to get the input from the user.

- **Model Checking Agent (MCa)**
  The MCa function as a manager of the whole system that will check and verify the message request whether it satisfies the system requirement. The requirement is verified based on the properties that we defined. This process is done in order to ensure that the system gets correct format and true message.

- **Output Agent (Oa)**
  Output Agent reacts if the messages are successfully verified from the MCa. Then, the Oa will accept the message and send it to the system server. The status of the user request will be automatically sent in back propagation to the user for acknowledgment.

- **Refine Agent (Ra)**
  However, if the message is identified as failure, the refine agent or Ra will handle the refinement of any errors in the user request by sending 'failure status'. It means that the user has to send back the request again for a new process. Otherwise, the message will be considered as violated and terminated automatically.

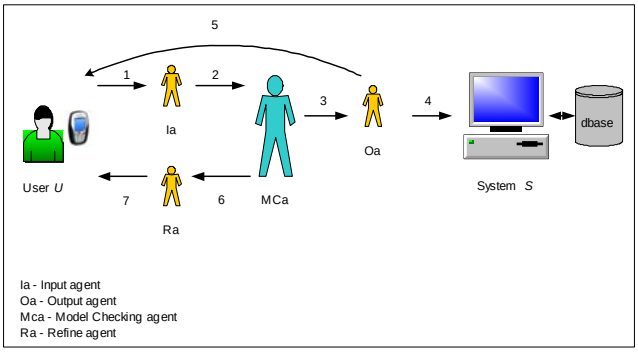

Figure 3. The architecture of agent-based verification system

4. Design Verification of Model Checking Agent

The design verification in our approach is based on the case study in Section 3 which focuses on the purchasing module. The verification process can be distinguished in two phases: the behavior of the whole model of the system and the properties that are handled by model checking agent. For the first phase, we illustrate the flow chart of the communication protocol in Figure. 4.

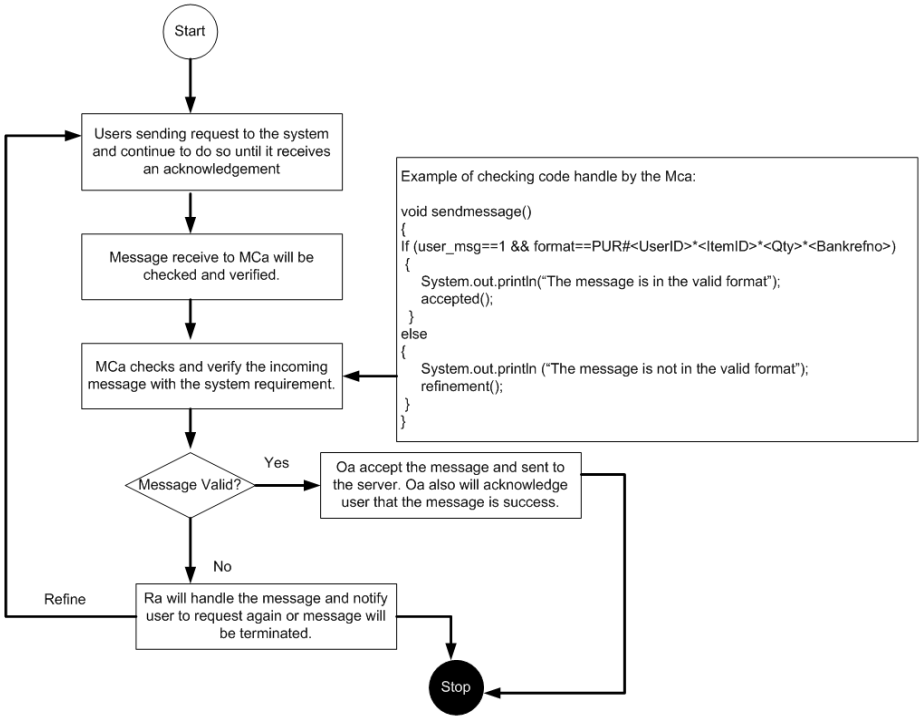

Figure 4. The flow chart of the system with supportive agent based

In developing the system, the state diagrams are used to determine the internal behavior of the protocol. Figure. 5 shows the state diagrams of purchasing protocol among agents.

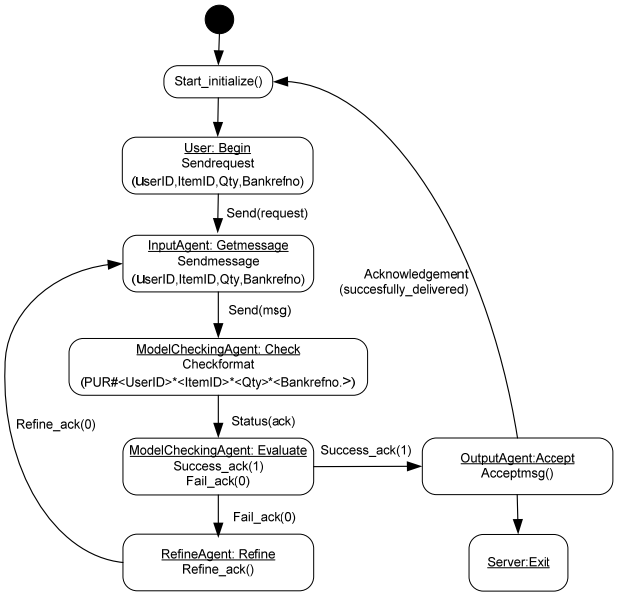

Figure 5. State diagram of agents for format checking

For the second phase, we have designed the behavior of the agents through the sequence diagram shown in Figure. 6 to identify the extensive flow of the format checking of SMS purchase transfer.

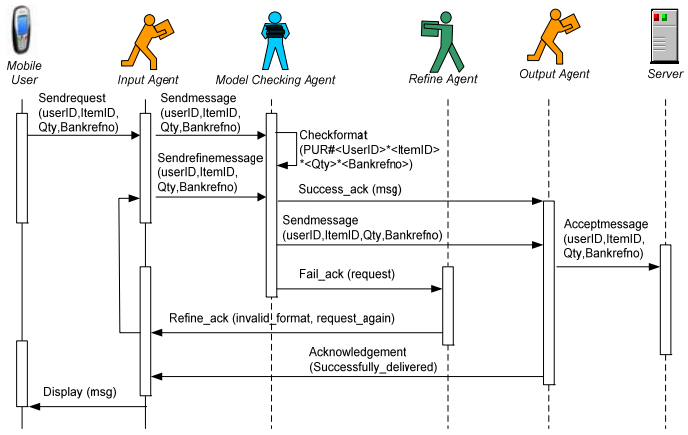

Figure 6. Sequence diagram for format checking of SMS purchase transfer

The protocols for the verification process are as follows:

1. User starts sending request to Ia. Ia represents the interface that sends message from the user to

the MCa. The process continues until it receives an acknowledgement from the Oa if the message is identified as a success.

2. Before the message is received by Server, the MCa verifies every message that comes into the system. The verification is done based on the text format and syntax checking.

3. The MCa checks the message and makes the comparison between the message and the system requirement.

4. Server does nothing until it receives the valid message. If the message is identified as an error in syntax and format, the Ra will handle the message and notify Ia to request again. Otherwise, the process is automatically terminated.

For properties verification, we have designed the agents to communicate based on the protocol. To confirm the properties such as liveness we have to convert the flow into a linear temporal logic (LTL). The LTL is a mathematical formula for statements on a linear time [9].

5. Simulation and Verification Results

We show the result of verifying the model using Spin based on PROMELA translation. When the search depth reached at state 32, the model checker Spin had successfully found an attack to the verification model. The attack needs to verify the three properties as follows: never-claim, assertion violations and acceptance cycles. The result was simulated using XSpin and has been exported in Figure 7.

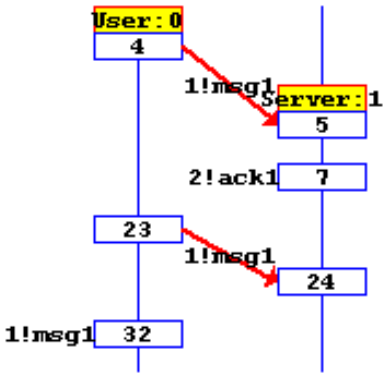

Figure 7. Message sequence chart from XSpin simulation

In XSPIN, errors are displayed in a graphical transition where all the processes are specified. When an error condition occurs, it generates a trace file used to identify the position that has been attacked by the error. Therefore, we demonstrated the analysis of the output result using Spin in Figure. 8 with the number of depth reached, state and transition explored. However, we have also investigated and analyzed the design to find the reason for the simulation result. We assume that the source code and specification that we specified from informal language to PROMELA do not perform as good as the automata model that we designed. Therefore, the search process is incomplete.

Full statespace search for:

never claim - (none specified)
assertion violations +
cycle checks - (disabled by - DSAFETY)
invalid end states +

State-vector 36 byte, depth reached 5, errors: 1
6 states, stored
0 states, matched
6 transitions (= stored+matched)
0 atomic steps
hash conflicts: 0 (resolved)
2.302 memory usage (Mbyte)

Figure 8. Output from the Spin model checker

6. Conclusion and Future Work

The automatic verification of model checking agent using model checker SPIN is one of the possible approaches to be considered especially for SMS mobile driven. It is important for the system to be designed accurately for system safety while does not affect the quality of service (QoS) in mobile application. The use of model checker XSpin shows the correctness of the design by providing a counter-example that defines sequences of processes that may lead to failure. However, the number of states that can be verified is limited due to errors found in the XSpin. However, the improvement for the system is being developed though it is still in the testing stage to make some adjustment of the specification. As mobile driven is a very important area, verifying every message sent by various agents with XSPIN are able to increase the quality of the end-to-end SMS in the future. As for long term vision, we are interested to enhance the developed architecture with different aspects of verification properties to increase the SMS quality.

.

7. Acknowledgement

This work is supported by the Ministry of Science & Technology and Innovation (MOSTI), Malaysia and Research Management Center, Universiti Teknologi Malaysia (UTM) under the Vot 79200.

8. References

[1] Schusteritsch R., Rao S. and Rodden K., 2005, "Mobile search with text messages:designing the user experience for google SMS". *In ACM Special Interest Group on Computer-Human Interaction (SIGCHI)*.

[2] Bernama, Feb 4, 2009, "Malaysia Records Third Highest SMS Traffic in Asia Pacific". Available:http://malaysia.news.yahoo.com/bnm/20090203/tbsmalaysia-sms-ceeeaba.html.

[3] Harris R., May 27, 2008, "Succeeding with SMS". Available:http://www.wirelessweek.com/Succeeding-SMS.aspx.

[4] Valentino C.A., 2007, "Progress Monitoring for the Field Operations of the 2007 Philipine Cencus of Population", National Statistics Office Philipines.

[5] Mee V. N. and Selamat A., 2007, "SMS Management System for Direct Sales and Network Marketing". *Fifth International Conference on Information Technology in Asia (CITA07).*

[6] Shengguang M.,Wei C., Gang L., ShitongW. and LiuW., 2008, "An Asset management System based on RFID, WebGIS and SMS", *2nd International Conference on Ubiquitous Information Management & Communication (ICUIMC'08)*.

[7] Configuring SMS WAN Bandwidth, 2004, Available: http://cramsession.brainbuzz.com/articles/getarticles.asp?aid=805.

[8] Laibinis L. and Troubitsyna E., 2008, "Formal Development of Cooperative Exception Handling for Mobile Agent Systems". *SERENE, ACM*.

[9] Kaliappan P.S. and Koenig H., 2008, "Designing and verifying communication protocols using model driven architecture and spin model checker". *Journal Software Engineering & Applications*.

[10] Beckert B., 2006, "Intelligent Systems and Formal Methods in Software Engineering". IEEE Intelligent System.

[11] Eugenio D. S., Francesco M. D., Marina M., Giacomo P., 2003, "Web applications design and maintenance using Symbolic Model Checking", Proceedings of the Seventh European Conference on Software Maintenance and Reengineering (CSMR'03) IEEE.

[12] Anderson B. B., Hansen J. V., Lowry P. B. and Summers S. L., 2005, "Model Checking for design and assurance of e-Business processes", Elsevier, Decision Support Systems 39, pp. 333-344.

[13] Giordano L., Terenziani P., Bottrighi A., Montani S. and Donzella L., 2006, "Model Checking for Clinical Guidelines:an Agent-based Approach". Journal of the American Medical Informatics Association (JAMIA) Symposium Supplement, pp. 289.

[14] Xiao M. and Li J., 2006, "The Modeling Analysis of Cryptographic Protocols Using Promela", IEEE Proceedings of the 6th World Congress on Intelligent Control and Automation, Dalian, China.

[15] Markosian L. Z., Mansouri-Samani M., Mehlitz P. C. and Pressburger T., 2007, "Program Model Checking Using Design-for-Verification: NASA Flight Software Case Study". IEEE Aerospace Conference, pp. 1-9.

[16] Wimmer R., Kortus A., Herbstritt M. and Becker B., 2008, "Probabilistic Model Checking and Reliability of Results". 11th IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems.

[17] Roberson M., Harries M., Darga P. T. and Boyapati C., 2008, "Efficient Software Model Checking of Soundness of Type Systems". Proceedings of the 23rd ACM SIGPLAN Conference on Object Oriented Programming Systems Language and Applications (OOPSLA '08).

[18] Osman N., Robertson D. and Walton C., 2006, "Dynamic Model Checking for Multi Agent Systems", Declarative Agent Language and Technologies IV, LNAI 4327, pp. 43-60.

[19] Weiss G., 2000, "Multi Agent System - A Modern Approach to Distributed Artificial Intelligence". The MIT Press Cambridge, Massachusetts London, England.

[20] Bang Ki S., Jin Hyun W., Yoo C. and Choi Jin Y., 2005, "System Resource Utilization Analysis based on Model Checking Method", An International Journal of Computing and Informatics (Informatica).

Siti Dianah A. Bujang has received a B.Sc (Hons) in Computer Science from the Universiti Teknologi Malaysia and currently is pursuing her M.Sc. Computer Science by research under a supervision of Assoc. Prof. Dr Ali Selamat at the same university since 2007 until now. She has published papers in several international and local conference including the IEEE and Springer. Her research interests include formal method, software agent, software engineering process and application, software modeling and specification and e-business.

Ali Selamat has received a B.Sc. (Hons.) in IT from Teesside University, U.K. and M.Sc. in Distributed Multimedia Interactive Systems from Lancaster University, U.K. in 1997 and 1998, respectively. He has received a Ph.D. degree from Osaka Prefecture University, Japan in 2003. Currently, he is an associate professor and IT manager at the School of Postgraduate Studies, UTM. His research interests include software engineering, software agents, web engineering, information retrievals, genetic algorithms, neural networks and soft computing.