

A Deployment-Oriented Service System Reliability Analyses

Jinyu Kai¹ Huaikou Miao² and Honghao Gao³

^{1,2} School of Computer Engineering and Science, Shanghai University, Shanghai, China

^{1,2} Shanghai Key Laboratory of Computer Software Testing & Evaluating, Shanghai, China

³ Computing Center, Shanghai University, Shanghai, China

¹aykjy@shu.edu.cn

Abstract

The single-instance deployment mode can't meet the needs of the increasing user access, and multiple-instance deployment is a feasible scheme to improve the scalability of the service system. The traditional reliability analysis methods are aimed at the single-instance deployment and they cannot be used to analyze the reliability of the service system of multi-instance deployment. To address this problem a new approach is proposed in this paper to analyze the reliability of service that a services system provided in the case of multiple-instance deployment. This approach extends the traditional user-oriented system reliability analysis method, adopting a vector representing the reliabilities of the multiple instances of a component service to settle down the problem. To bring this method to fruition, a prototype framework is constructed and realized in Java, and a case study demonstrates this approach.

Keywords: multiple-instance deployment, user profile model, component service reliability model under multiple-instance deployment, system reliability model.

1. Introduction

Cheung's approach [1] is one of the first approaches to analyze the reliability of the traditional software system. Cheung believes that the reliability of system depends both on the reliabilities of the components and the probabilistic distribution of the utilization of the components. Cheung proposes a user-oriented reliability model which combined the user profile based on the system architecture and the component reliability growth model to measure the reliability of service that a system provides to a user community. Some recent approaches extend Cheung's approach from different aspects; however, these methods are mainly used to analyze the reliability of the system in the design stage.

As for the past reliability model of system in the design phase, the reliabilities of the component services are often evaluated in the presence of single-instance deployment mode; With the increase of the amount of user access to the service system, Single-instance deployment mode can't meet the needs of user access, and multiple-instance deployment is a feasible scheme to improve the scalability of the service system. For example, component service requests from different regions of the user may be assigned to different component servers deployed in different regions. In this case, if still using the reliability analysis methods assuming that services are deployed in single-instance mode analyzes the reliability of system in which component services are actually deployed in multiple-instance mode, the result obtained can be inconsistent with the actual value.

To this end, a method of reliability analysis of service system in which component services are deployed in multiple-instance mode is proposed in this paper. On the basis of the traditional user-oriented system reliability analysis method, a reliability vector is adopted to depict the reliabilities of multiple instances of a same component service; the

user profile based on the system architecture and the reliability vector of each component services which deployed in multiple-instance mode are combined to build a composite model; from this composite model, a user-oriented software reliability model for multi-instance deployment is derived based on conditional probability theory. On the condition of Markovian assumption, the probabilistic model checker PRISM is used to get the reliability and other analysis. To bring this method to fruition, a prototype framework is constructed and realized in Java, and a case study demonstrates this approach.

One of the contributions of this work is to propose a model of reliability analysis of service system in which component services are deployed in multiple-instance mode; different from the previous analysis model where component services are deployed in single-instance mode, the model proposed in this paper aims at the case where component services are deployed in multiple-instance mode, this will provide more accurate reference to guide the adaptation of the service system. Another is a framework is constructed and a prototype is realized in Java which make this analysis method fruition and the reliability of service system in which component services are deployed in multiple-instance mode can be analyzed automatically.

The remainder of this paper is organized as follows. Section 2 gives the state of the art of this work; section 3 introduces the approach proposes in this paper in detail; section 4 shows a case study; and section 5 concludes this work and presents directions for future research.

2. Related Works

Reliability is one of the most important quality attributes of a software system beyond its functional attributes. Techniques for reliability analysis of a software system can help software developers in detecting potential reliability problem in all stages. It can help to make the system more reliable and bring profit to the system owners.

A great deal of reliability analysis methods has been developed to support this task. [2] presents a QoS model to compute the quality of service for workflows based on atomic task QoS attributes; the QoS model is essentially a collection of the aggregation functions which are used to compute the quality of service for workflows automatically based on atomic task QoS attributes. [3] presents an approach based on the theory of fuzzy set to compute and evaluate the trustworthy property of a service flow; the approach essentially uses a given function to computer the trustworthy property of the service flow. Those above analytical approaches are better suited for production workflows [4] which are more structured. [5] focuses on the reliability analysis of an assembly of service and proposes an algorithmic reliability analysis methodology that is based on the service assembly structure and the reliability information published by each assembled service. When using the algorithmic to calculate the reliability of the service assembly, Markov property is assumed. The reliability of the service assembly is defined the probability from its start state to the end state. This analytical approach still builds on the basis that the service assembly is structured control flow.

State machine and its variants most notably, state charts have been recognized as a class of important model for virtually all software engineering methodologies proposed by[6] . State charts can be used to represent the dependencies among the components of the system. [7] clearly describes the motivation to choose state charts to represent the dependencies. The sustained success of state machines in software engineering stems from the fact that state machines provide for both a concise mathematical theory and an intuitive semantics of system behavior which naturally allows for visualization, hierarchy, and abstraction. The system described by state machine is labeled transition system (LTS), and intuitively, a Markov chain is an FSM extended with transition-probability distribution [8] and furthermore with Markov feature. This becomes the basis of the state-based reliability analysis methods.

Cheung's approach [1] is one of the first approaches based on state to analyze the reliability of the traditional software system. Cheung believes that the reliability of service depends both on the reliability of the components and the probabilistic distribution of the utilization of the components to provide the service and develops a user-oriented reliability model which combined the user profile based on the system architecture and the component reliability growth model to measure the reliability of service that a system provides to a user community. Cheung's model uses an absorbing Discrete Time Markov Chain (DTMC) to express the components reliability state dependency of a software system and defined the system reliability equals to the probability of reaching the terminal state from the initial state of the Markov process.

Some recent approaches extend Cheung's approach from different aspects. The following approaches extend Cheung's approach from architectural styles. [9] supports different architectural styles such as batch-sequential style, parallel/pipe-filter style, fault tolerance architectural style, call-and-return style. The approach of [8] is based on Rich Architecture Definition Language (RADL) but employs the same underlying theory as Cheung's approach for reliability prediction. The following approaches extend Cheung's approach from the factors influencing the component reliability. The approach of [10] takes interface failures and network connection failures into account and [11] supports modeling component restarts and retires. The approach of [10] takes into consideration of error propagation, software fault tolerance mechanisms and concurrently present errors. However, these methods are mainly used to analyze the system reliability in the early development stages on the assumption of that the every components will be deployed in a single-instance modes, such as in the design stage, in order to develop high-reliability system, with very little if any effort being devoted to analyze the reliability of the software system running in the multiple-instance deployment context.

To address the problem of analyzing the reliability of the software system running in the multiple-instance deployment context, a new approach is proposed in this paper. This approach extends the traditional user-oriented system reliability analysis method, adopting a vector representing the reliability of a component service in the case of multi instance deployed, combining the user profile based on the system architecture with the reliability of component services under multiple-instance deployment to construct a composite model, from this composite model a user-oriented software reliability model for multi-instance deployment being derived according to the conditional probability theory. On the condition of Monrovia assumption the probabilistic model checker is used to achieve the system reliability.

3. Methodology

The steps of constructing the composite model for reliability analysis of service system deployed by multiple-instance modes proposed in this paper are as follows:

3.1 To Construct the Reliability Model of the Service System

The goal of the architecture-based software reliability analysis approach is to estimate the system reliability taking into account the information about the architecture of the software made out of components, accordingly, there is a need for a modeling technique which can be used to analyze software components and how they fit together.

Referring to [12], we acknowledge that component-based software engineering identifies the concept of component as a building block or an elementary unit in software architecture modeling, assessment, development, and management. According to such perspective, a component follows from three main principles of software engineering: separation of concerns, modularization, and reuse and a software system is an aggregate of (software) modules or components. The component-based software paradigm is the basis of modern service-oriented architecture (SOA) software system, considering the

component identification, in service system the term component is used to denote a web service or a service.

Service composition is one class of service system based on SOA in which the components assume the form of web service selected and aggregated into a composed process which is implemented through a workflow language, like BPEL (Business Process Execution Language) or BPM (Business Process Modeling).

[13] believes that a model is a higher level abstraction of some aspect of a system such as the knowledge of requirements, design, implementation and deployment *etc.* which can Effectively reduce the complexity of software system description, moreover, support the operations like query, analysis, reasoning, modify and so on. Therefore, model-based analysis has become the basis of most software analysis methods. For reducing the complexity of the analysis, in this paper the architecture model is extracted from the BPEL document representing the control flow of system when considering the architecture extraction.

The architecture reveals the components which may be used to compose whole software and the interaction behavior among components described by certain architecture relation which can roughly classified into three types, such as control, data, and use [6]. In addition to this, state machine and its variants most notably, state charts have been recognized as a class of important model for virtually all software engineering methodologies proposed by [6]. State charts can be used to represent the dependencies among the components of the system. [7] clearly describes the motivation to choose state charts to represent the dependencies. In this paper the state charts are chosen as an architecture model, the essence of which is labeled transition system (LTS).

The motivation for considering the user profile can be traced back to [1], who believes that a rarely invoked service obviously has a smaller impact on the invoking service's reliability than a frequently used service. Thus, the user profile describing the component's utilization should be taken into account when to estimate the reliability of the system. In this paper, the user profile description will be got based on the Maximum Likelihood Estimation (MLE) method.

An execution component is called an instance component. Components are viewed by analyzers as black boxes. No other information but their observed behavior, such as their failure profile is available. The same to the traditional component failure growth model, we use the failure profile to construct the component failure growth model; different from the traditional one, we adopt a reliability vector to represent the reliability of a component deployed by multiple-instance mode instead of treat it as determined (e.g.,)in the case of the component deployed in single-instance mode.

The same as the common steps of building the composite model, to combine the user profile based architecture and the reliability model of a component deployed by multiple-instance mode to construct the composite model is the necessary step to assess the reliability of the reliability of service system deployed by multiple-instance mode. Supported by the conditional probability theory, the two models are combined to build the composite model. The essential of this composite model is probabilistic labelled transition system (PLTS).

3.2 To Quantitatively Assess the Reliability of the Service System

Markov random process introduced by Russian mathematician Markov in 1907 is one class of stochastic process with Markov feature which is always to model memoryless system satisfying that the current state of model contains all information that can influence future evolution of the system. Formulization Markov feature is as follows:

Assume that x_i is a random event occurring at the $i - th$ period and $p(x_i)$ represents the probability that a random event x_i occurs at this time, then for any moment $k, (k \geq 1), p(x_k|x_1, x_2, \dots, x_{k-1}) = p(x_k|x_{k-1})$

On the premise of conform to the Markov feature, When calculating the joint probability of multiple random events $x_1x_2\dots x_n$, only need to calculate the following expression:

$$p(x_1x_2\dots x_n) = p(x_1)p(x_2|x_1)p(x_3|x_2)\dots p(x_n|x_{n-1})$$

So it can be seen that Markov feature simplifies the joint probability calculation process of multiple random events.

According to their state space and index parameter properties, Markov random process can be classified into discrete-time stationary Markov chains (DTMCs), continuous-time stationary Markov chains (CTMCs) and Markov decision processes (MDPs).

Fortunately, some scenarios about the software system reliability study conforms to the Markov property, when assessing the reliability of service system deployed by multiple-instance modes, in this paper, we assume that during the running of the service system, the transfer relationship between the states conforms to the Markov feature, at the same time, we limit to discuss the reliability of the component is time independent. In this case, the discrete time Markov chain is adopted to model the user-oriented reliability of service system deployed by multiple-instance modes, Markov feature provides the necessary assumption for analyzing system reliability.

The same to the definition of the user-oriented software reliability, the user-oriented reliability of service system deployed by multiple-instance modes is defined as a function of the reliability of the components and frequency distribution of utilization of these components, and to calculate the probability of reaching the end state from the start state of the system is the reliability we assess.

Monrovia models are widely used to analyze probabilistic related properties of both system designs and deployed systems. The emergence of probabilistic model checkers makes this analysis to be performed with high accuracy [14]. In this paper, the reliability of the service system is calculated by using the probability model checker PRISM [15], which makes the process to analyze the reliability of the system visually.

A framework for constructing a user-oriented reliability analytical model for service system deployed in multiple-instance mode is shown in Figure 1.

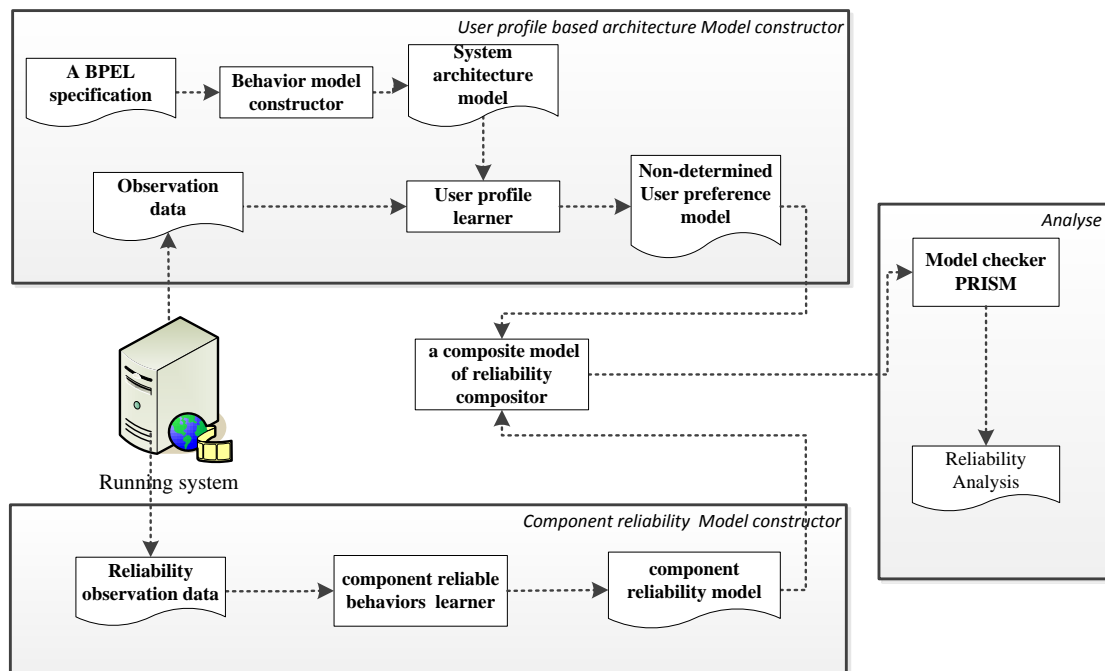


Figure 1. A Framework for Analyzing User-oriented Reliability of System in Multiple-instance Mode

To bring this composed model to fruition, three Web Services, Bpel2dot, M4UP, M4RF are realized in Java, the Bpel2dot Web Service takes as input a BPEL specification to construct its function model and the two Web Services, M4UP, M4RF provide the operation to quantitatively measure the probabilistic distribution of user preference on branching component Web Service and that of random feature of each component Web Service respectively based on the observation data about the running BPEL process adopting the MLE approach that is proposed in this paper; and then the composed model in dot format is exported.

4. Experimental Analytical Results and Discussion

The SystemModelEditor is a service system is chosen to describe the approach proposed in this paper to analyse the reliability of service that a services system provided in the case of multiple-instance deployment. The SystemModelEditor is a simple service system which visually shows the system model after editing in one of two alternative ways. There are three component web services combined to this system, namely, vDotEditorGrappa which represents the web service of editing the system model in graphical manner and tDotEditor which represents the web service of editing the system model in textual manner, and DotLayer which represents the web service of showing the system model edited as a figure. The web services of vDotEditorGrappa, tDotEditor and SystemModelEditor are deployed in the Tomcat containers in different machines. After finishing the work to edit the system model using the web service of vDotEditorGrappa or tDotEditor, users need to request the web service of DotLayer to view it. Due to the huge requests, the multiple-instance mode is adopted to deploy this component web service. We adopt the strategy that to deploy the component DotLayer web service into two machines, the requests after finishing the web service of vDotEditorGrappa transfer to one machine, and the requests after finishing the web service of tDotEditor transfer to another one machine. For reasons of space, the BPEL implement document doesn't be given in this paper. The server request record log pattern is configured to four fields, namely, remotehost, agent, request and refpage separately. The fields of remotehost and agent are used to identify users and sessions and the fields of request and refpage are used to calculate the user profile based on the system architecture and the growth reliability of the components. We collect the request log of this system and the log file is 25420KB including 6000 pieces user request and this is taken as the raw examination data to compare the result using different reliability analysis approaches.

Figure 2 represents the system architecture extracts from the BPEL implement document and the user profile got by analyzing the log file and the components reliability that deploy in the multiple-instance mode. Figure 3 shows the composite model which is combined by the user profile based on system architecture and the component reliability deployed in the multiple-instance mode. Table 2 shows the prism code which is translated from the composite model shown in Table 3.

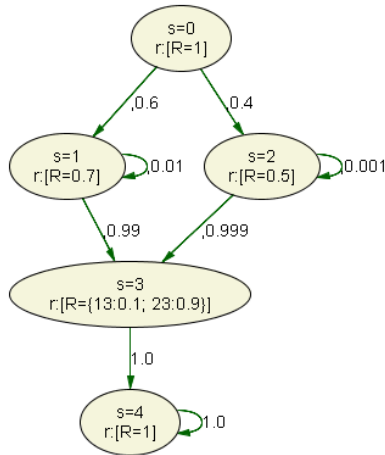


Figure 2. The Architecture of the Example

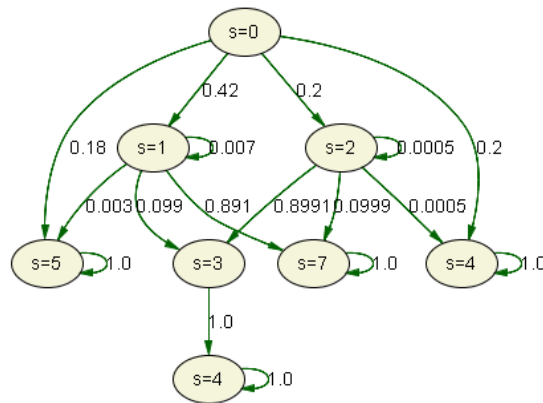


Figure 3. the Composite Analytic Model for Example

Table 2. the PRISM Mode Representing the Reliability of the System

```

dtmc
module m
s : [0..7] init 0;
[] s=0 -> 0.18:(s'=1) + 0.2:(s'=3) + 0.2:(s'=4) + 0.42:(s'=2) ;
[] s=6 -> 1.0:(s'=7) ;
[] s=2 -> 0.003:(s'=1) + 0.099:(s'=6) + 0.007:(s'=2) + 0.891:(s'=5) ;
[] s=7 -> 1.0:(s'=7) ;
[] s=3 -> 5.0E-4:(s'=3) + 0.8991:(s'=6) + 5.0E-4:(s'=4) + 0.0999:(s'=5) ;
[] s=4 -> 1.0:(s'=4) ;
[] s=1 -> 1.0:(s'=1) ;
[] s=5 -> 1.0:(s'=5) ;
endmodule
    
```

The property associated to the global reliability of the system is expressed in PCTL as: $P=?[F(s=4)]$ and the property associated to the reliability from initial state to the fail state of web service DotLayer is expressed in PCTL as: $P=?[F(s=7)]$.

We analyze the reliability of the service system deployed in multiple-instance mode using four comparison group, where group 1 uses the approach that proposes in this paper, and the rest groups use the cheung's approach and the component reliability are according to the every instance, that is $R\{13=0.1\}$ or $R\{23=0.9\}$, and the average value of the sum of the component reliability, that is $R=(0.1+0.9)/2=0.5$ respectively.

The table below shows the reliability of the property 1 associated to the global reliability of the system and the property 2 associated to the reliability from initial state to the fail state of web service DotLayer obtained using the probabilistic model checker PRISM.

Table 3. The Reliability Results of the Service System in Multi-instance Deployment

PCTL formula	result
State=initial => P[F(system_state=success)]	0.20010005002501252
State=initial => P[F(DotLayerRuning_state=fail)]	0.22178306675996606

It can be seen from the experimental data that in the case of a small number of system state and component instances, there is less difference between the traditional method and our method as for a user-oriented reliability analytical for service system deployed by multiple-instance mode; but with the increase of the system states number and the component instances number and in the case of high accuracy request of reliability, there will be an obvious difference between our approach and the traditional method.

5. Conclusions

In this paper, we present a user-oriented reliability analytical approach for service system deployed by multiple-instance mode. This approach extends the traditional user-oriented system reliability analysis method and a vector is adopted to represent the reliability of a component service in the case of multi instance deployed, the user profile based on the system architecture and the reliability of component services under multiple-instance deployment are combined to construct a composite model, thus a composite model of a user-oriented software reliability model for multi-instance deployment is derived according to the conditional probability theory. On the assumption of Markov feature, the system reliability is analyzed by the method based on the probabilistic model checker. Via a case study, we demonstrated the applicability of our approach.

However, the approach to a user-oriented reliability analysis for service system deployed by multiple-instance mode aims to one concrete deployment situation. There exist many schemes to deal with the problem of the increase of users requests, to put forward a more general analysis method to solve more cases of multiple-instance deployment is our future work.

Acknowledgement:

This work is supported by National Natural Science Foundation of China (NSFC) under grant No.61572306, 61502294 and 61170044, and Young University Teachers Training Plan of Shanghai Municipality under Grant No. ZZSD13008 CERNET IPv6 Innovation Project under Grant No.NGII20150609, 2015 Film Summit Science Foundation of Shanghai University, and Natural Science Foundation of Shanghai under Grant No. 15ZR1415200.

References

- [1] R. C. Cheung. A user-oriented software reliability model. *IEEE Trans. on Software Engineering*, SE-6(2):118–125, March 1980.
- [2] Cardoso J, Sheth A, Miller J, Quality of service for workflows and web service processes [J]. *Web Semantics Science Services & Agents on the World Wide Web*, 2004, 1(3):281–308.
- [3] LIU, MIAO, and MA. "An Approach to Constructing User-constrained Trustworthy Service Flow." *Chinese Journal of Electronics: English Verison*. 3(3),(2011):425-431.
- [4] McCready, S. There is more than one kind of workflow software. *Computerworld* November. 2 (1992): 86-90.
- [5] Grassi, V., and Patella, S. Reliability prediction for service-oriented computing environments. *Internet Computing IEEE*. 10, 3 (2006) 43-49.
- [6] Chaki, S., Clarke, E. M., Groce, A., Jha, S., & Veith, H. Modular verification of software components in C(Magic). *Software Engineering, IEEE Transactions on*, 30(6), (2004).388-402.
- [7] Liangzhao Zeng, Boualem Benatallah, QoS-Aware Middleware for Web Services Composition. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, VOL. 30, NO. 5, MAY 2004
- [8] Reussner, R H., Schmidt H. W., and Poernomo I. H. "Reliability prediction for component-based software architectures." *The Journal of Systems and Software* 66(2003):241-252.
- [9] Pham T T, Défago X, Huynh Q T. Reliability prediction for component-based software systems: Dealing with concurrent and propagating errors [J]. *Science of Computer Programming*, 2014, 97:426-457.
- [10] Lipton M W, Gokhale S S. Heuristic Component Placement for Maximizing Software Reliability [J]. *Springer*, 2007:309-330.
- [11] Sharma V S, Trivedi K S. Reliability and Performance of Component Based Software Systems with Restarts, Retries, Reboots and Repairs[C]// 2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE). *IEEE Computer Society*, 2006:299-310.
- [12] Distefano S., Filieri A., Ghezzi C., and Mirandola R. A compositional method for reliability analysis of workflows affected by multiple failure modes. *Proceedings of the 14th international ACM Sigsoft symposium on Component based software engineering*. (2011):149-158.
- [13] France R., and Rumpe B. "Model-Driven Development of Complex Software: A Research Roadmap." *Future of Software Engineering, 2007. FOSE '07 IEEE*, 2014:37 - 54.
- [14] R Calinescu, K Johnson, Y Rafiq. Using Observation Ageing to Improve Markovian Model Learning in QoS Engineering. *ICPE'11*, March 14–16, 2011, Karlsruhe, Germany.
- [15] <http://www.prismmmodelchecker.org/>
- [16] Hansson, H., & Jonsson, B. (1994). A logic for reasoning about time and reliability. *Formal aspects of computing*, 6(5), 512-535.
- [17] Kai, J., Miao, H., Zhao, K., Zhou, J., & Gao, H. (2015). A Platform for Analyzing Behaviors of Service-Oriented Application Based on the Probabilistic Model Checking. *International Journal of Software Innovation (IJSI)*, 3(2), 24-38. doi:10.4018/ijsi.2015040104

Authors



Jinyu Kai, She is currently a Ph.D. student in the School of Computer Engineering and Science of Shanghai University, China. Her research interests include formal methods, software engineering, Web service application development, web application development and quantitative model checking.

Huaikou Miao is currently a professor in Computer Engineering and Science at Shanghai University, China. His research interests include formal methods and software engineering.

Honghao Gao received the Ph.D degree in the School of Computer Engineering and Science of Shanghai University, Shanghai, China, in 2012. His research interests include Web service and model checking.

