Reliability Prediction for Internetware Applications: A Research Framework and Its Practical Use

ZHENG Zibin[1*], MENG Jingke[1], TAO Guanhong[1], Michael R. Lyu[2]

[1] School of Data and Computer Science, Sun Yat-Sen University, Guangzhou 510000, China
[2] The Chinese University of Hong Kong, Shatin, N.T., Hong Kong, China

Abstract: The open and dynamic environment of Internet computing demands new software reliability technologies. How to efficiently and effectively build highly reliable Internet applications becomes a critical research problem. This paper proposes a research framework for predicting reliability of individual software entities as well as the whole Internet application. Characteristics of the Internet environment are comprehensively analyzed and several reliability prediction approaches are proposed. A prototype is implemented and practical use of the proposed framework is also demonstrated.
Keywords: reliability prediction; internet application; software reliability

I. INTRODUCTION

Software is a crucial part of our society. With the development of Internet technologies, heterogeneous, large-scale and complex Internet applications are prevalent nowadays, and we are increasingly relying on them. Internetware is a software paradigm that provides a set of technologies for developing applications to meet computing requirements in the Internet environment [1]. A software paradigm defines *what is* to be constructed and executed, *how todevelop* the resulting software artifacts, *how to run* the artifacts, and *how well* the constructed and executed software can perform [2]. Software reliability is one of the most important characterizes of the last category (i.e., *how well*) and is the core issue for the success and pervasiveness of a software paradigm. Software failures can lead to a disruption of online services and loss of revenues, and in extreme cases, even lead to severe and fatal consequences in critical Internetware applications, which are software applications developed employing the Internetware paradigm.

Internetware applications are widely employed in e-business, e-government, automotive systems, and a lot of other domains. Building highly reliable Internetware applications is becoming more and more important. Internetware applications are usually composed by a number of distributed software entities on-demand, which are implemented as autonomic software services and deployed at different Internet nodes (e.g., servers, virtual machines, etc.). Compared with traditional standalone software applications, building highly reliable Internetware applications is very challenging, caused by: (1) the Internet environment is open, highly dynamic and ever changing; (2) the software entities are heterogeneous with unpredictable reliability and behaviors, and some software entities are provided by other organizations without any source codes and internal implementation details; and (3) Internetware applications are becoming large scale, extremely complex, and requiring long-term self-adaptive online services for thousands of users. Due to these reasons, most of the traditional software reliability engineering techniques cannot meet the challenges in the Internet environment.

The major software reliability engineering techniques can be classified in the following areas: *fault avoidance*, *fault removal*, *fault tolerance*, and *fault prediction*[3]. Fault avoidance techniques are the initial defensive mechanism against unreliability. When faults are injected into the software, fault removal is another protective means. When inherent faults remain undetected through the testing and inspection processes, fault tolerance is the last defending line in preventing software failures. Finally, if software failures are destined to occur, it is critical to estimate and predict them. This paper focuses on the reliability prediction of Internetware applications. Since Internetware application comprises distributed Internetware entities dynamically, accurate reliability prediction for the whole Internetware application is highly relying on accurate prediction of the Internetwareentities. The highly dynamic nature of Internet and the situational-aware adaptation of Internetware entities bring great challenges for accurate reliability prediction of Internetware entities. Moreover, even we can make accurate prediction of Internetware entities; it is still quite difficult to make accurate reliability prediction of the whole Internetware application, since these entities are collaborating with each other on-demand. In other words, the structure of the Internetware application is much more dynamic compared with traditional component-based applications. Reliability prediction of Internetware application emerges as a challenging and urgently requiredresearch problem.

This paper proposes a reliability prediction framework of Internetware applications. The rest of this paper is organized as follows: Section 2 proposes a reliability prediction framework for Internetware, Section 3 discusses reliability prediction of Internetware entities, Section 4 investigates reliability prediction of the whole Internetware application, Section 5 shows our implementation and practical use, and Section 6 concludes the paper.

II. Reliability Prediction Framework for Internetware

As shown in Fig. 1, our proposed reliability prediction framework includes Internetware applications, Internetware entities, and the reliability prediction server. Based on a key concept of Web 2.0, i.e., user-collaboration, our framework encouraged users (i.e., Internetware application developers) to contribute their individually observed reliability information to exchange for accurate and personalized reliability prediction on the unexplored software entities. By this way, we can exploit the user-side reliability data to make personalized reliability prediction. Take the *Internetware Application 1* in Fig. 1 as an example, the reliability prediction procedure in our proposed framework is as follows:

(1) The Internetware application comprises seven software components. Some of these components are remote Internetware entities which are provided by third-party and deployed on the Internet. Failure information of the executed remote software entities can be recorded.

(2) The *Internetware Application 1* provides the user-perceived reliability information of the invoked Internetware entities to our reliability prediction server.

(3) After processing the user-provided data, we store the reliability information into a database.

This work proposes a research framework for predicting reliability of individual software entities as well as the whole Internet application.

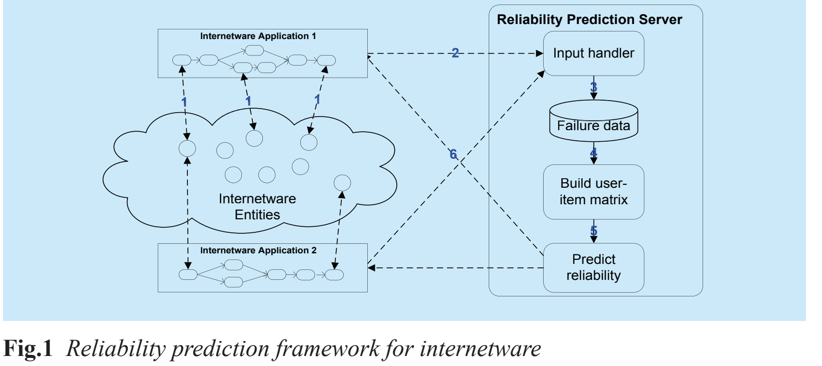

Fig.1 *Reliability prediction framework for internetware*

(4) Based on the reliability information provided by different users, a user-item matrix can be built.

(5) By predicting the missing values in the user-item matrix, personalized reliability prediction on the unused Internetware entities can be achieved for different users.

(6) We provide the reliability prediction results of Internetware entities to different users. Based on the valuable information, compositional analysis can be conducted to make reliability prediction of the whole Internetware application.

III. Reliability Prediction of Internetware Entities

In the Internetware software paradigm, an application involves a number of distributed Internetware entities. Reliability of the application heavily relies on the remote entities. The first step of predicting reliability of the Internetware application is to predict reliability of the individual Internetware entities. Different from traditional component-based systems, Internetware entities may be provided by third-party developers without any source codes and internal implementation details. They are invoked remotely via Internet connections and may demonstrate unpredictable behaviors. The reliability prediction of Internetware entities contains the following characteristics.

3.1 From server-side to user-side

An Internetware entity serves many different users (i.e., Internetware application developers who develop different Internetware applications). Influenced by the highly dynamic Internet environment and the unpredictableenvironmental factors, different users may observe quite different reliability on the same Internetware entity. Moreover, a user may obtain unreliable services from a highly reliable Internetware entity. Therefore, the research focus of reliability prediction of Internetware entities is shifting from server-side (the server where the Internetware entity is deployed) to user-side (the Internetware application that invokes the entity). User-perceived reliability of the Internetware entities provides more valuable information to the Internetware application developers.

3.2 From development time to runtime

Traditional reliability prediction approaches collect failure data during integration testing or system testing during the development time. However, the failure data collected by the developers may be limited and may not be able to represent the failures under real operational environment. Since Internetware application is running 7/24 for serving a large number of users continuously, the reliability prediction model should be built at runtime and should be able to adapt to the runtime and environmental changes.

3.3 From temporal to spatio-temporal

Traditionally, past failure data are collected and analyzed to predict the future reliability of the same software component. In other words, only the temporal dimension is considered. In the Internetware environment, a large number of Internetware entities are available on the Internet and some of them are providing equivalent or similar functionalities. Besides analyzing the past failure data of the current Internetware entity, past failure data of other Internetware entities can also be employed in making more accurate reliability prediction for the currententity. By this way, both temporal and spatial dimensions are considered in the prediction models.

3.4 From model-oriented to data-oriented

Traditional reliability prediction models attempt to analyze the underlying mechanism that influences the software reliability and build a mathematically formulation to represent the failure manifestation process. The proposed models typically involve a number of parameters. Failure data collected in real operation are analyzed to estimate these parameters. In the Internetware environment, a large volume of runtime failure data can be obtained from both server-side and user-side. Various data mining techniques (e.g., collaborative filtering [4], matrix factorization [5] and so on) can be employed to analyze the collected data and build data-oriented reliability prediction models.Data-oriented models bypass analyzing the underlying process which impacts software reliability. It simply assumes that no matter how many underlying factors there are and how complicated their interactions may involve, there are enough data in the Internetware environment to capture the reliability aspect of the entities. Consequently, we can predict the reliability of a software entity shifting the focus from model-oriented approaches to data-oriented approaches, with advanced statistical analysis, data mining techniques and machine learning schemes.

These unique characteristics make reliability prediction of software entities a great challenge. To attack this challenge, neighborhood-based approaches and model-based reliability prediction approaches are proposed in recent research investigations [6]. The problem of reliability prediction of Internetware entities can be modeled as a missing value prediction problem in a user-item matrix as shown in Fig. 2. In the matrix, each row represents a user and each column represents an Internetware entity. An available value $p_{u,i}$ in the matrix represents the reliability of an entity i observed by a user u, and the missing values in the matrix should be predicted. By predicting the missing values in the matrix, personalizedreliability prediction can be achieved for different users.

Neighborhood-based approaches try to predict the missing values using information from similar neighbors, e.g., similar users. Pearson Correlation Coefficient (PCC) [5] is widely employed to calculate the user similarity based on the user-perceived failure probability of different Internetware entities. Employing the information from similar users, the neighborhood-based approaches [6] predict the missing

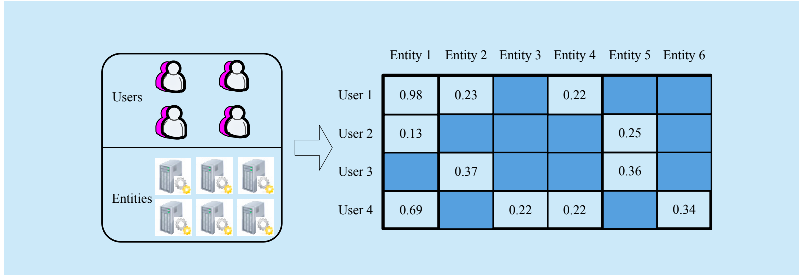

Fig.2 *Toy example of the user-item matrix*

value $p_{u,i}$ in the user-item matrix by the following equation:

$$p_{u,i} = \overline{p_u} + \sum_{a \in S(u)} w_a \left(p_{a,i} - \overline{p_a}\right) \tag{1}$$

where $\overline{p_u}$ and $\overline{p_a}$ are average failure probabilities of different entities observed by users u and a, respectively, and w_a is the weight of the similar user a, which can be calculated by

$$w_a = \frac{Sim(a,u)}{\sum_{b \in S(u)} Sim(b,u)} \tag{2}$$

where $Sim(a,u)$ is the PCC similarity between user a and user u, and $S(u)$ is a set of similar users to the current user a. This weight ensures that users with higher similarity have larger impact on the missing value prediction. By Eq. (1), a missing value in the user-item matrix can be predicted via employing past entity usage experience from other similar users.

When the user-item matrix is very sparse, similar users can hardly be identified. Moreover, the computational complexity of the neighborhood-based approaches is high. To address these drawbacks, model-based approaches, e.g., the matrix factorization approach, factorize the original user-item matrix P into two matrices W and H, and employ these two matrices to predict the missing values in P. The prediction problem can be modeled as the following optimization problem.

$$\min_{W,H} \zeta(W,H) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}^{P} (p_{ij} - W_i H_j)^2 + \frac{\gamma}{2} \|W\|_F^2 + \frac{\gamma}{2} \|H\|_F^2 \tag{3}$$

where I_{ij}^{P} is the indicator function that is equal to 1 if the value p_{ij} is available in the user-item matrix and equal to 0 otherwise, W_i is the i^{th} row of matrix W (representing the user-specific coefficients of user i), and H_j is the j^{th} column of matrix H(representing the factor vector of Internetware entity j), γ controls the extent of regularization for penalizing large values in the matrices W and H to avoid the overfitting problem, and $\|W\|_F^2$ denotes the Frobenius norm. The factorization problem can be stated as: given a partially observed user-item matrix P, find two matrices W and H that minimize the sum-squared errors with the original matrix P. Well-known algorithms such as the gradient descent algorithm can be employed to solve this optimization problem. By obtaining the matrices of W and H, missing values in the matrix P can be obtained by $W \times H$.

The model-based approaches can achieve more accurate prediction results via exploiting all the available values in the user-item matrix, and require lower computational complexity. On the other hand, the neighborhood-based approaches are simple and intuitive on a conceptual level, and thus widely used in practice [6].

IV. Reliability Prediction of Internetware Applications

In traditional software systems, the involved tasks are clearly defined and performed by pre-designed software components. However, in the Internetware environment, the requirements are unclear, and the Internetware entities can collaborate with each other dynamically to complete the tasks. Just like Web pages in the World Wide Web can link to each other to form an information Web, the Internetware entities can also invoke each other to form a software Web. In an Internetware application, Interentware entities do not have a predefined collaboration style. We do not know what kind of collaboration will occur. In other words, the software Web is dynamically updated, since the collaboration among Internetware entities are dynamically changing. Reliability of the Internetware application is highly relying on the structure of the software Web. To achieve accurate reliability prediction of the Internetware application, comprehensive investigations of the software Web should be conducted.

Microsoft reported that by fixing the top 20 percent of the most reported bugs of Windows and Office, 80 percent of the failures and crashes would be eliminated [7]. Similarly, in an Internetware application, key entities have greater impact on the reliability of the whole application. Identifying the key entities becomes an important research task. When analyzing the software Web, the idea of PageRank can be borrowed to identify the key entities [8]. For example, some entities are frequently invoked by a lot of other entities. These entities are considered to be more important, since their failures will influence a lot of other entities. Intuitively, the key entities in an Internetware application are the ones which are frequently invoked by a large number of other important entities. The following PageRank like equation can therefore be employed to obtain a *significant value*, which represents the importance of an entity in an Internetware application:

$$V(c_i) = \frac{1-d}{n} + d \sum_{k \in N(c_i)} V(c_k) f_{ki} \tag{4}$$

where n is the number of entities and $N(c_i)$ is a set of entities that invoke the entity c_i .The parameter $d (0 \leqslant \mathrm{d} \leqslant 1)$ in Eq. (4) is employed to adjust the significance values derived from other entities, so that the significance value of c_i, denoted as $V(c_i)$, is composed of the basic value of itself (i.e., $\frac{1-d}{n}$) and the derived values from the entities that invoke c_i. By Eq. (4), an entity c_i has a larger-significance value, if the values of $|N(c_i)|$ (the *number of entities that invoke* c_i), $V(c_k)$ (*the significance value of the entity* c_k),and f_{ki} (the *frequency of entity* c_k *invoking* c_i) are large, indicating that the current entity c_i is invoked by a lot of other important entities frequently.

After identifying the key entities, the composition of Internetware entities can be conducted to predict the reliability of the whole application. Individual entities may fail; however, the Internetware application as a whole should meet the reliability requirement. The way of composing the Internetware entities imposes great impact on the reliability of the whole system.

As shown in Fig. 3, there are four commonly used compositional structures:

- **Sequence**: The software entities in a sequence structure are invoked one by one. The sequence structure fails if anyof its entities fails.
- **Branch**: In the branch structure, only one branch willbe executed each time. The branch structure fails if the executed entity fails.
- **Loop:**A software entity will be invoked for multiple times. The loop structure fails if any of these invocations fails.
- **Parallel**:All software entities in a parallel structure will be executed at the same time. A parallel structure fails if any of the parallel invoked entities fails.

These compositional structures can be combined or nested in various ways, leading to a complex Internetware application. In the Internetware environment, since the structure of the application is dynamically changing, the past prediction model may not be used anymore. To address this new challenge, predicting reliability of single Internetware entities, then aggregate the reliabilities of different entities dynamically according to the compositional structures to obtain the reliability of whole system is a good way. The reliability aggregation details of these four compositional structures can refer to work [9].

V. Implementation and Practical Use

As shown in Fig. 4, to demonstrate our proposed reliability prediction framework for Internetware, we implement a prototype, named WS-DREAM (Distributed REliability Assessment Mechanism for Web Services), which can be examined in http://www.wsdream.com. Our prototype employs publicly available Web services as Internetware entities. By providing reliability information of some invoked Web services, our server will provide personalized reliability prediction for the unexplored Web services to new users. Our prototype has already collected a large number of reliability information contributed by different users globally. After processing the data, reusable research datasets are provided in the website for future research investigations. Based on the collected Web service reliability information from real-world, we have the following observations:

(1) A large part of service invocation failures are related to Internet connections. The highly dynamic Internet environment makes building reliable Internet application a great challenge. Building highly reliable software entities is not enough. We need to emphasize user-perceived reliability prediction in the Internetware environment.

(2) Our collected data show that performance of a software entity is highly dynamic. Making accurate reliability prediction in such dynamic environment is really a great challenge.

(3) Our prediction framework collects reli-

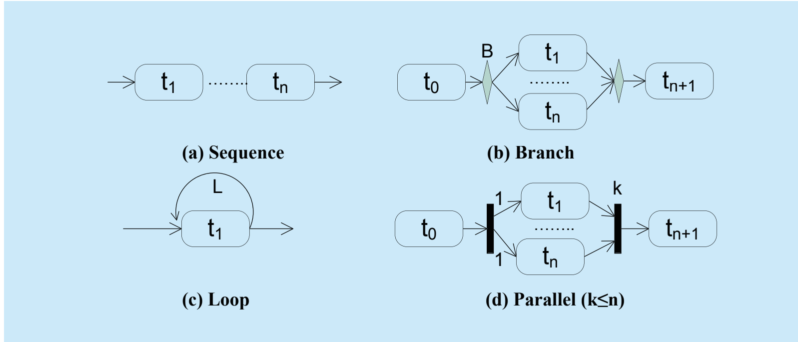

Fig.3 *Compositional structures*

Fig.4 *Interface of WS-DREAM (wsdream.com)*

ability data from different users, where malicious users may exist. How to ensure the data quality is an important research issue.

(4) Since a user usually only invokesa small number of Web services, the user-item matrix is very sparce, which makes accurate reliability prediction difficult.

Based on this prototype, we also investigate prediction of other user-perceived quality, e.g., response time and throughput, beside the reliability. Since the service status (e.g., workload, CPU allocations,etc.) and the network environment (e.g., congestions, bandwidth,etc.) may change over time, response time of Webservices varies a lot during different time intervals. In orderto identify low response time Web services timely, time-aware performance of Web services needs to be continuously predicted [10]. To address the data sparsity problem of the user-item matrix which greatly degrades the prediction accuracy, we deployed a small set oflandmarks to periodically monitor the responsetimes of the Web services. By combining the advantagesof network coordinate based approaches and collaborativefiltering based approaches, the response times between usersand Web services can be accurately predicted using theircorresponding Euclidean distances [11]. By such kinds of time-aware [10] and network-coordinate based [11] quality prediction approaches, the prediction accuracy can be greatly improved.

VI. Conclusion

Software failures impose great impact on our society. Reliability is one of the most important factors for the pervasiveness of the Internetware paradigm. Designing efficient and effective reliability prediction approaches to improve the reliability of Internetware is a challenging research task as well as a great research opportunity. In this paper, we propose a reliability prediction framework for Internetware, and investigate the reliability prediction of Internetware entities as well as the whole Internetware application. In particular, we present two general reliability prediction approaches for Internetware entities and design a PageRank-based approach for identifying the key entities from a complex software Web to enable better reliability prediction of the whole application. We further implement a prototype of the proposed Internetware reliability prediction framework based on real-world Web services, and report our experiences in these practical applications. Our practical use of the framework shows a promising resultin deploying this reliability prediction framework in Internetware environment.

Reliability prediction of Interentware applications is a very challenging and urgently required research problem, since (1) behavior of software entities are dynamically changing according to the environmental changes, (2) software entities are collaborate dynamically for specific needs in the Internetware environment, and (3) the Internetware application can change its structures and behaviors dynamically. More comprehensive and systematic investigations should be conducted on the reliability prediction of Internetware applications in the future.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Project No. 61472338, 61332010), Guangdong Natural Science Foundation (Project No. 2014A030313151), the Fundamental Research Funds for the Central Universities, and the Research Grants Council of the Hong Kong Special Administrative Region, China (No.415113).

References

[1] Hong Mei, Gang Huang, Tao Xie, “Internetware: A Software Paradigm for Internet Computing,” IEEE Computer, vol. 45, no. 6, pp. 26-31, June, 2012.

[2] Hong Mei, Xuanzhe Liu, “Internetware: An Emerging Software Paradigm for Internet Computing,” Journal of Computer Science and Technology, vol. 26, no. 04, pp. 588-599, 2011.

[3] Michael R. Lyu, “Software Reliability Engineering: A Roadmap,” Future of Software Engineer-

ing, in Proc. of the 29th International Conference on Software Engineering, Minneapolis, May 20-26, 2007, pp. 153-170.

[4] R. Salakhutdinov, and A. Mnih, "Probabilistic matrix factorization," in Proc. of Advances in Neural Information Processing Systems, pp. 1257-1264, 2007.

[5] H. Ma, I. King, and M.R. Lyu, "Effective Missing Data Prediction for Collaborative Filtering," in Proc. of the 30th International ACM SIGIR Conference, Amsterdam, July 23-27, 2007, pp. 39-46.

[6] Zibin Zheng and Michael R. Lyu, "Personalized Reliability Prediction of Web Services," ACM Transactions on Software Engineering and Methodology, vol. 22, no. 2, pp. 1-28, 2013.

[7] P. Rooney, "Microsoft's CEO: 80-20 Rule Applies to Bugs, Not Just Features," ChannelWeb, Oct. 2002.

[8] Zibin Zheng, Tom Chao Zhou, Michael R. Lyu, and Irwin King, "Component Ranking for Fault-Tolerant Cloud Applications," IEEE Transactions on Services Computing, vol. 5, no. 4, pp. 540-550, 2012.

[9] W.-L. Wang, D. Pan, and M.-H. Chen, "Architecture-based software reliability modeling," Journal of Systems and Software, vol. 79, no. 01, pp. 132–146, 2006.

[10] Yilei Zhang, Zibin Zheng, and Michael R. Lyu, "An Online Performance Prediction Framework for Service-Oriented Systems", IEEE Transactions on System, Man, and Cybernetics, Part A, accepted.

[11] Jieming Zhu, Yu Kang, Zibin Zheng, and Michael R. Lyu, "WSP: A Network Coordinate based Web Service Positioning Framework for Response Time Prediction", in Proc. 10th IEEE International Conference on Web Services, Research Track, pp. 90-97, Hawaii, US, June 24-29, 2012. ZHU C, CORSON M. A Five-Phase Reservation Protocol for Mobile Ad Hoc Networks. Wireless Networks,2001, 7(4): 371-384.

Biographies

ZHENG Zibin, is an associate professor at Sun Yat-sen University. His research focuses on software and services reliability engineering. He is the corresponding author, email: zibin.gil@gmail.com.

MENG Jingke, is a graduate student at the Sun Yat-sen University, Guangzhou, China. Her research interests include mobile services, Internet technology, and distributed systems.

TAO Guanhong, is a graduate student at the Sun Yat-sen University, Guangzhou, China. His research interests include mobile services, Internet Technology, and services reliability engineering.

Michael R. Lyu, is a professor at the Computer Science & Engineering Department, The Chinese University of Hong Kong. His research focuses on software engineering, services computing, and intelligent computing. Lyu received a PhD in computer science from University of California, Los Angeles. Contact him at lyu@cse.cuhk.edu.hk.