An Active Method to Building Dynamic Dependency Model for Distributed Components

Yunchun Li, Wei Li, Mingfeng Zhang, Chunyan Hou
College of Computer Science and Technologies
Beihang University, Beijing, China
lych @buaa.edu.cn, liw@buaa.edu.cn,
zhangmingfeng03@citiz.net,hcysuper@sohu.com

1. Introduction

Currently many research show that, in a sophisticated application system, the faults which are impossible to occur theoretically, may take place in practice. J2EE (Java 2 Platform, Enterprise Edition) distributed environment has been popularly applied to EAI (Enterprise Application Integration). With the growth of the numbers of Jsp, Servlet and EJB components, for a specific J2EE application, it is difficult for administrators to locate the fundamental position of the faults, and delay recovering the faults.

Dependency models provide the effective method to trace all possible sources of the faults from the problem vertices against the relationship edges. Bayesian network was presented in 1981 by R.Howard and J.Matheson. It has been successfully applied to fault diagnosis field. Bayesian networks provide a method to describe consequence information naturally.

In this paper, we construct the dependency models of software components with the construction algorithm of Bayesian networks. Dependency models can be represented with Bayesian networks. The vertices in Bayesian networks corresponds to the vertices in dependency models, and the conditional probability expressed with the edges in Bayesian networks corresponds to the relative strength expressed with the edges in dependency models. Consequently, it is possible to develop a tool to analyze and recover the faults automatically, and be helpful to find fundamental reasons for various faults, based on dependent relations between the components in dependency models.

2. Methods to Fault Injection and Capture

Our experiments are based on the Websphere application server and the sample application, Plantsbywebsphere, and we simulate the practical operations of a business process with one hundred concurrent clients, such as registration, making bills, submission and so on.

According to J2EE architecture, we apply the active fault injection[1] to each layer according to the characteristic of its realization. In the database layer, we inject faults by renaming the tables. Therefore, the EJB components throw out the exception and capture them to determine the affected domain. In EJB layer, we can perform the fault injection through changing the name-object binding in the EJB server. Actually, there is a rename function in the JNDI interface functions, which can be used to change the name existing in the name-object binding. In the experiment, we visit all eight EJB components in the Plantbywebsphere application, inject faults to each one and capture the impact to the components in the web layer.

The mechanism of active fault injection is shown as figure 1. One fault adapter and one fault collector are packed with the components. The fault adapter is used to start and stop the fault generation, and the fault collector is used to report the fault. At runtime, the components first register their names and addresses to the catalog server. The fault collector in both the component

Proceedings of the Second International Conference on Autonomic Computing (ICAC'05)
0-7695-2276-9/05 $ 20.00 **IEEE**

IEEE COMPUTER SOCIETY

layer and the application layer collect the faults respectively from the components and the users. These faults, as well as the fault component flag controlled by the controller, are reported to the data collector and grouped as 2-tuples like <exceptional component, fault component>. The statistic results are then used to deduce by the Bayesian analyzer

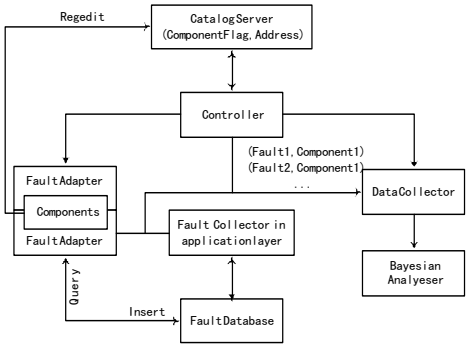

Figure 1. The mechanism of active fault injection

3. The Construction of Dependency Models

Cheng provided a three-phase algorithm based on dependency analysis [2] and developed a tool , BN Power Constructor, which constructs Bayesian networks with the data obtained from the database and behaves very well [3]. We processed the collected results of the faults with the analysis program in order to construct a table and used it as the input to BN Power Constructor. Each row in a table corresponds to a component in the applications. For each obtained 2-tuples, <exceptional component, fault component>, add a record to the table, and 0 stands for the normal component, 1 stands for the fault component, and 2 for exception. For example, the Servlet components may depend on the EJB components whereas the EJB components can't depend on the Servlet components. The dependency model of the components in the Plantsbywebsphere application was constructed by the BN Power Constructor, as well as the condition probability tables, is shown as following figure 2：

When comparing this model with the dependency model constructed manually by analyzing the source code, we find that these two

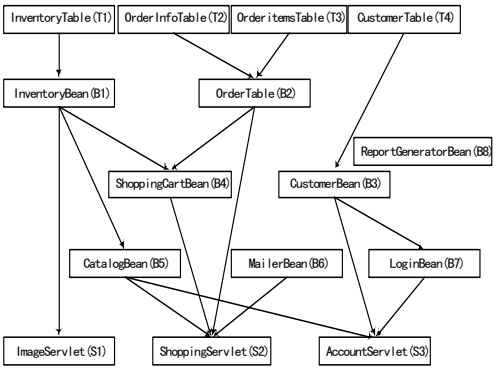

Figure 2. The dependency model of the PlantsByWebsphere application constructed with Bayesian network.

structures are the same. We verified these values correspond with the practical situation of the Plantsbywebsphere application.

4. Conclusions

In this paper, a method is presented to dynamically construct dependency models of distributed software components, based on active fault injection and three-phase BN construction algorithm. We apply this method to a sample application in J2EE environment and verify its validity by experiments. The results show that the system can automatically construct a precise dependency model and can be used to optimize the method to trace the source of the faults.

5. References

[1]. A. Brown, D. Patterson, An Active Approach to Characterizing Dynamic Dependencies for Problem Determination in a Distributed Environment, *in Seventh IFIP/IEEE International Symposium on Integrated Network Management, Seattle, WA, May 2001.*

[2]. Jie Cheng, Learning Bayesian networks from data: an information-theory based approach, *Artificial Intelligence, 137(1), 2002.*

[3]. Jie Cheng, David Bell, Weiru Liu, Learning Bayesian Networks from Data: An Efficient Approach Based on Information Theory , *Alberta Techical Report, 1998.*

IEEE COMPUTER SOCIETY