

Trusted Software Behavior Monitoring Model Based on Agent

Caimao Li*, Shaofan Chen, Hanwei Wu and Jing Chen

College of Information Science and Technology, Hainan University, Haikou, 570228, China

*Corresponding author

Abstract—This paper studies trusted model for software behavior monitoring Based on agent. First, it put forward a trust monitoring software behavior framework, then analyzed the dynamic proxy monitoring mechanism, built a dynamic proxy monitoring model based on AOP. Finally, this paper analyzed the monitoring ability to weave mechanism by AOP, and designed a weaving module of monitoring model based on AOP.

Keywords—modeling; monitoring; trusted software; agent; AOP

I. INTRODUCTION

In an open and dynamic network environment, along with the development of the software requirements, software entities loose aggregation. the complexity of software is more and more higher. software behaviors are complex and varied. Software source code of scale is more and more bigger, so, the loopholes and defects are difficult to avoid. Software's failures and mistakes often occur, even failure. it is detrimental to people's work and life. As a result, the credibility of software has become quite a common problem. Availability, reliability and security of software has been the credible highlight important character. In order to solve the credibility, prominent problem of software increasingly, we need to implement effective monitoring software behavior.

Traditional software monitoring technology in modularization and flexibility generates code scattered. So, it's monitoring demand expression ability is not strong, and it's monitor code tangles and scattered. It does not support the operation process of the target system dynamically add or delete monitor. It is difficult to adapt to large, loose aggregation, action complex software monitoring [1]. This paper researches trusted modeling technology of software behavior monitoring based on dynamic proxy.

II. TRUSTED SOFTWARE BEHAVIOR MONITORING FRAMEWORK

Those credibility of the main performance on software behavior are these, which can monitor running behavior, assess behavior results, control abnormal behavior. All of Monitoring software are such as monitoring the interaction between software entities, collecting behavior information, providing basic data for diagnosis, forecasting and assessing the credibility for behavior control, etc [1,2]. From the perspective of trusted software behavior, because of lack of behavior monitoring, it cannot be trusted to behave in accordance with evaluation, and cannot examine the credibility of main body

behavior. So, in the process of software dynamic evolution, it is difficult to guarantee the correctness of the dynamic evolution, also hard to meet the demand of dynamic evolution of software reliability.

Consequently, in reference on the basis of forefathers' research, this paper builds a trusted software monitoring framework, as shown in figure I. The framework builds a software behavior trusted monitoring environment, such as providing software behavior for each control unit, state of the trusted monitoring, reliable control and reliable management services. Software behavior monitoring is divided into reliable monitoring, reliable control and reliable management. Reliable monitoring includes behavior description and behavior monitoring. Reliable control includes anomaly of behavior control and evolutionary of behavior control. Trusted management includes trusted behavior modeling, behavior trusted evaluation. All parts of reliable monitoring framework work together, and monitor the software behavior together [2,3].

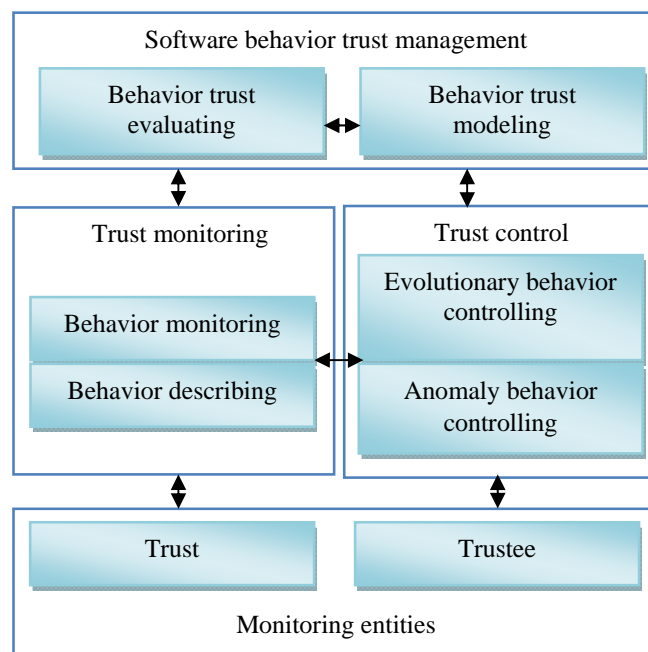


FIGURE I. TRUSTED SOFTWARE MONITORING FRAMEWORK

In the framework of the trusted software behavior monitoring, monitored entities are divided into trust and trustee. If the trustee can satisfy the desired behavior of trust, the

trustee is thought to be credible. If you want trustee is trusted, you need to monitor the trusted runtime behavior of the person. On this basis, it conducts credible evaluation, so as to decide whether the trustee is trusted, and decide whether its behavior is dynamically. In the process of software operation, according to the credible evaluation specifications and the credible assessment results of reliable software behavior, you may need to take the corresponding control strategy, make the abnormal behavior can be controlled, control the dynamic behavior evolution of the appropriate strategy for software, and improve the reliability of software.

Under the environment of dynamic evolution, the software reliable requires that software system can be along with the change of environment and the demand for dynamic evolution. That trusted software behavior from the software behavior monitoring information, needs for software behavior to conduct a comprehensive, accurate, real-time monitoring, collecting relevant data credibility. Describe software behavior are behavior monitoring, verification, assessment and control.

For monitoring, verification, assessment and control software, it needs to have the software behavior description, and be the credible monitoring software behavior. That software reliable monitoring includes behavior description and behavior monitoring, provides the core basis for behavior credible evaluation. That software behavior monitoring for software behavior credible evaluation provides the basis of original data credible evaluation. Trusted software management include behavior modeling, behavior credible evaluation and trusted management framework. It aims at providing a credible guarantee mechanism for software dynamic evolution. Trusted software behavior modeling is to construct a model of the behavior trusted relationships between software entities. Behavior trusted model provides a credible relationship establishment and management behavior, and a framework of its focus on behavior credible description and behavior evaluation. Credible management framework provides activities for software behavior credible modeling and behavior credible evaluation. That reliable control software behavior mainly includes the evolution and abnormal of behavior control. It improves continuous availability of software, credibility of the software behavior.

III. SOFTWARE TRUST MONITORING MODEL

A. Proxy Mechanism

The proxy pattern provides an agent to control access to an object. The proxy pattern is divided into static and dynamic agent. Static agent is the relationship of the proxy class between the delegate class before running. The dynamic proxy is just generated in software running stage, so it don't have to care about the proxy class during the implementation phase. Dynamic proxy processing class provides two main methods. The one is used to bind to the agent's business class and create at run time and return a dynamic proxy class. The another is used to define the content of the dynamic proxy, which encapsulates the actual call business method.

B. AOP Monitoring Agent

In this paper, it uses the dynamic proxy of Aspect Oriented Programming(AOP). AOP supports two kinds of dynamic proxy in JDK and CGLIB (Code Generation Library) . The JDK dynamic proxy is implemented by java internal reflection mechanism. CGLIB uses a very low-level bytecode technology. CGLIB's principle is to create subclasses through the bytecode for a class, and uses method in the subclass to intercept all calls to the superclass method, then weave the crosscutting logic.

AOP dynamic proxy help to weave into good implementation monitoring intercept code. In the process of physical interface method called, first is to set the intercept point, then weave into a method call before, during and after, so as to realize dynamically monitoring code execution.

AOP way of weave can be divided into static and dynamic weave. Relative to the static weave, dynamic weaving without having to learn and modify the source code of the target system, so, dynamic monitor code didn't been woven into business code to the target system. Therefore, this paper uses Aspect Oriented Programming (AOP) dynamic proxy weave way to set the crosscutting concerns processing logic weave to the target class methods of monitoring software behavior.

C. The Monitoring Model Based on Agent

Traditional monitoring model generally adopts the object oriented Programming (OOP) technology to achieve monitoring function of the software. AOP as a kind of object oriented programming (OOP), widely used in processing some system-level service with characteristics of crosscutting, such as transaction management, security, caching, object pool management, etc. AOP implementation is the key to AOP framework to automatically create AOP agent [3].

On the basis of existing research, this paper uses dynamic agent technology of Aspect Oriented Programming (AOP), putts forward a kind of modified monitor model. On the basis of traditional OOP dimension, the modified monitor model introduces the dimension of AOP, establishes a trusted software monitoring model based on dynamic proxy, as shown in figure II. This model uses java dynamic proxy to implement AOP interception function. Namely, on the basis of the target class increase section of logic, the trusted software monitoring model can weave the monitoring code into the target system, generate enhanced target class, and be effective to trusted system behavior, condition monitoring. In dynamic proxy monitor model, monitoring unit of monitoring model is the monitoring entity. AOP can create multiple Agent implementation monitoring and control of each monitoring unit [3,4].

Software monitoring mainly include the monitoring objects, monitoring unit and monitoring framework elements, etc. Monitoring object is a software in a variety of entities. Monitoring unit is controlled by monitoring components, monitoring engine and parts. It is the core of the software to monitor management [5,6].First of all, it provides monitoring unit with the perceptive ability of the software runs by monitoring components for software running state. Then, monitoring the engine according to the monitoring information to produce the corresponding control decision, by the control

unit to complete the specific implementation. Software monitoring framework provides monitoring services. Monitoring service manages monitoring information, analyzes monitoring information processing, provides the basis for control the generation of decision making.

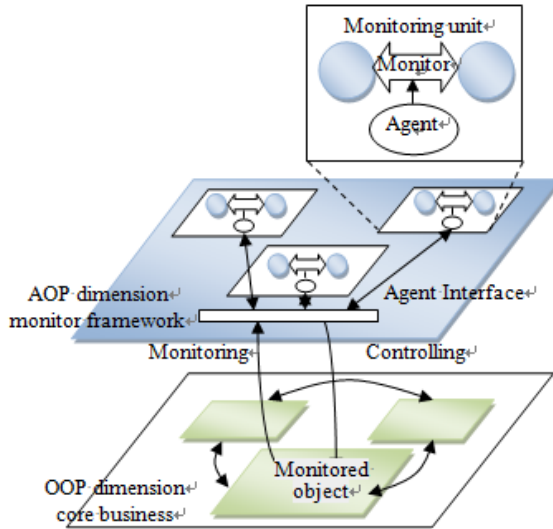


FIGURE II. MONITORING MODEL BASED ON DYNAMIC PROXY

Agent in dynamic monitoring model, monitoring units have been separated from monitored objects. Each monitoring unit independently completes the behavior, state monitoring. Monitoring framework manages all of the monitoring unit to effectively monitor information, processing and analysis.

IV. CREDIBLE MONITORING ABILITY TO WEAVE DESIGN BASED ON AGENT

A. AOP Agent Weaving Mechanism

In software trusted monitoring model, AOP dimension is applied to deal with the problem of crosscutting concerns in the dimension of OOP. Based on the dynamic proxy AOP design reliable monitoring, it realizes the crosscutting concerns from core attention, and provides a modular implementation mechanism of crosscutting concerns. AOP system needs to isolate crosscutting concerns and core concerns, with corresponded to Class and Aspect. Class and Aspect can be parallel development. Finally, using aspect weaving tools can will aspect knitting with Class, and form the running application system. AOP proxy mechanism of solving crosscutting concerns as shown in figure III.

AOP implementation mechanism is the join of access, edit, naming and abstract. Dynamic proxy is a new introduction JDK1.3 dynamic proxy mechanism. Spring AOP woven into the implementation of mechanisms provides a good agent to generate AOP agent class source code. Spring AOP core idea is to crosscutting concerns is extracted from the system, which encapsulated during the design and implementation phase, thus improves the modular software system.

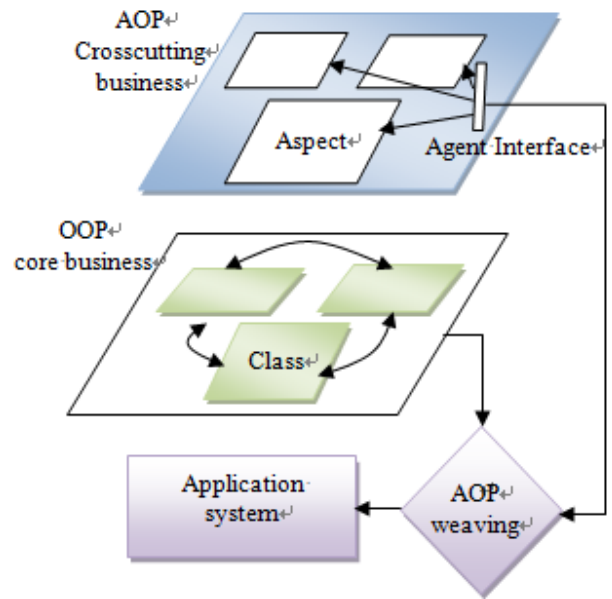


FIGURE III. AOP AGENT WEAVING MECHANISM

B. AOP Monitoring Ability to Weave into the Mechanism

This paper uses Spring AOP design trusted software monitoring framework, supports through the model behavior constraints generated woven into the code, and realizes the automatic monitoring of the program, without AOP code to write. Trusted monitoring framework provides monitoring ability to weave mechanism automatically, as shown in figure IV.

In the framework of the reliable monitoring, the basic ideas of monitoring ability to weave into are these. First of all to monitor demand analysis, behavior describe on the monitor model needs to monitor the nature of the constraints. Then, the resolution generated code of monitoring automatically by using monitoring of generator after joining the constraints,. Finally, through the aspect of monitoring aspect weaver, it can generate code with the original business function code system for knitting, and form the resulting target system with monitoring ability.

C. Monitoring Ability to Weave Modules Design

Software reliable monitoring ability to weave into the mechanism of monitoring framework implementation using Spring AOP technology. It separated from monitor logic and business logic on the dimension completely. On the dimension of AOP, through dynamic proxy independent modular implementation of monitoring, it solves crosscutting concerns in OOP programming code tangles and scattering problems

AOP dimension realizes the monitoring module, and loose coupling between the target system. It has a better code reuse. Trusted software framework of monitoring ability to weave modules structure design as shown in figure V. Monitoring ability to weave modules by model analysis, monitoring aspect code automatically generated and AOP weaver of three parts.

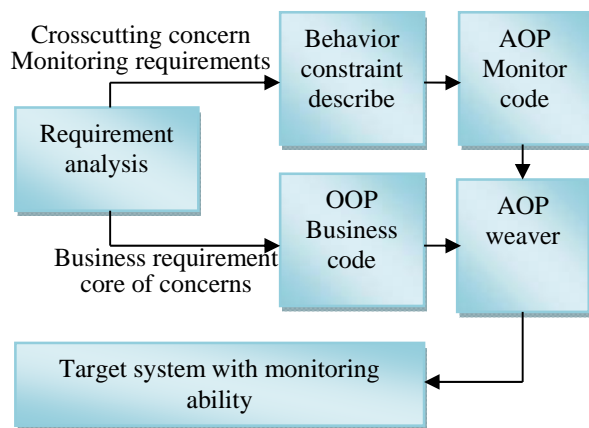


FIGURE IV. AOP MONITORING ABILITY TO WEAVE INTO THE MECHANISM

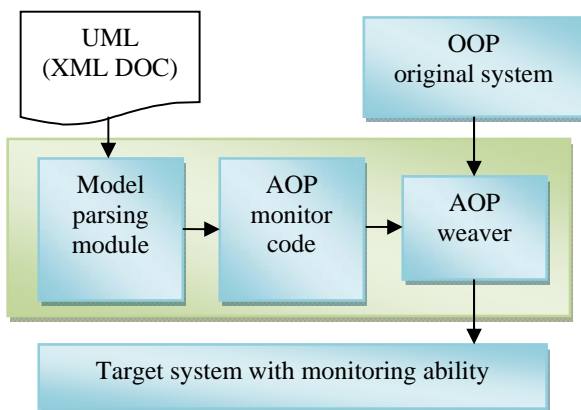


FIGURE V. MONITORING ABILITY TO WEAVE MODULES STRUCTURE

In monitoring ability to weave modules, first of all is UML model information parsing. XML form model information into the standard form of AOP tools support. According to generate template in advance design of monitoring, that generates from the aspects of model in the XML document parsing monitoring parameters are written into side templates, then generate code by using in respect of monitoring of generator automatically. Finally, AOP weaver generates monitoring aspect of code with the original system, then, generates monitoring ability target system, woven for system monitoring ability.

Monitor all woven into the process automatically, without human intervention, so it avoids the logic programming based on AOP for monitoring needs. In this way, it decreases the difficulty of monitoring personnel's workload and work to eliminate the threshold of AOP technology. Monitoring framework is not limited by practical application, different type of system, through the selection of different monitoring templates. AOP weaver can easily implement monitoring ability to weave. Monitoring ability to weave modules are

loosely coupled, only through data interaction between each other. each module can be individually to design, develop and upgrade changes. It has a good scalability.

V. CONCLUSIONS

In this paper, we study credible monitoring software behavior modeling based on agent. From the perspective of separation of concerns, the broker implementation mechanism of AOP technology is introduced, putted forward a credible monitoring framework, software behavior analysis of the dynamic proxy monitoring mechanism, built dynamic proxy monitoring model based on AOP. Finally, it analyzed of AOP agent ability to weave the supervision mechanism, and designed a monitoring weave modules based on AOP. At present, the main application software based on Java EE platform as the research background, the prototype system needs further implement and improves the monitoring framework.

ACKNOWLEDGMENT

This work was financially supported by two Natural Science Foundation of Hainan Province {613158, 614231}.

REFERENCES

- [1] Canjun Wan, Changyun Li. Research and development on behavior monitoring and controlling of trusted software in dynamic evolution environment. *Application Research of Computers*. 2009: 1201~1204
- [2] Canjun Wan, Changyun Li. Component interaction behavior monitor based on dynamic aspect-oriented programming. *Journal of Computer Applications*. 2011(31): 572-576
- [3] Jianming Chen, Song Liu, Zhishu Li, Gejian Ding. Trusted Software Programming Model Based on Formal Monitoring. *Computer Engineering*. 2011(37):66-68
- [4] Xi Yang, Tong Li. Research and implementation of AOP framework based on incremental dynamic evolution. *Computer Engineering and Applications*. 2013.49(23):39-44.
- [5] D.Hachani, D.Bardou. Using Aspect-Oriented Programming for Design Patterns Implementation[C]. *Proc.Workshop Reuse in Object-Oriented Information Systems Design*. 2002:345~354
- [6] Grundy J. Aspect-oriented requirements engineering for component-based software system[C]. *4th IEEE International Symposium on RE*. IEEE Computer Society Press. 1999.84-91