2010 International Conference on Computer Application and System Modeling (ICCASM 2010)

The Integration of Security Modeling and the UML Software System based on SPE

Yin Zhengjie
Management science and engineering College
Dongbei University of finance and economics
Dalian China
Yzj4696526@163.com

***Abstract*—Performance and security are two important aspects of the system development. Now we can not analysis them during the system development, but give remedy to them after the system modeling. Software performance engineering is an important aspect of the current software development, but the software's performance has not been paid attention to during the course of the system designing. We often focus on the software development quality, while ignoring others. This paper mainly studies the concept and process of the software performance engineering, and how to integrate SPE with the UML while focusing on the system security problems. As the current software development does not involve safety issues, the system security is an important aspect of systems. the system's security policies and mechanisms is often embedded in the software system after the software development accomplishment, this kind of nonstandard treatment result in significant hidden danger , so the paper should integrate them in the UML system.**

Keywords-component; UML, SPE, Security modeling, VPL, VBAC

I. INTRODUCTION

Performance is a computer software quality indicator, with the appearance of the software, "performance" also appeared. After Software comes into the engineering period, due to the well-known importance of software quality, it is considered as an important concept or factor of software life cycle, but this opinion make the impact of the performance of the work is often postponed. It is often said that "let the system move first, and then let the system work better", "first consider function, and then consider the quality ". For this reason, despite of the attention on the performance, but the system did not finish the engineering about performance. Until recent years, software performance engineering emerges after the failure of the application development process. Software architecture putting major impact on their quality is an indisputable fact. "It has basically decided whether a system can reach the expectation of performance when choosing the system architecture". Software performance engineering are widely recognized and developed. It has guided many types of software performance engineering practice, such as performance of distributed systems SPE method, WEB applications, etc.

II. SOFTWARE PERFORMANCE ENGINEERING (SPE)

A. The conception of SPE

The performance of software refers to the degree of fit on the software running time and space occupation to the user needs. If the software running time during running course becomes consistent with the user expectations, the user can receive the performance of the software; On the other hand, the user may think that the performance of the software is difficult to accept. So the software performance is the responsiveness of the user running the software and flexibility (the maintenance of the responsiveness when the function is added.

Responsiveness of the system is the ability of the system achieving the responsive time or throughput target. Flexibility is the ability of keeping to achieve the responsive time or throughput target when the system adding the software performance demands. Responsiveness and flexibility are two important aspects of the software system and becoming more and more important especial on the WEB application. We share the same experience that we became impatient when accessing to information in the site while the response time is too long WEB. Finally we stopped to visit the site.

So, how to improve the software system responsiveness and flexibility? In accordance with traditional practices, we must be available to measure the performance of the analysis. Then we consider the achievement of the performance of the system. This view leads to emphases put to the coding stage of performance issues, but not in the design stage of the performance. We did not integrate the performance considered in the design stage or in the coding stage into one project management, no matter the special performance engineer. SPE is just the solution.

The so-called Software Performance Engineering (SPE) is a systematic, quantitative method. It can meet the performance objectives for building a software system. SPE is a kind of software-oriented approach, focusing on architecture, design and implementation choices. This approach helps developers choose to meet the performance objectives of architecture and design and control effectively resources need. SPE uses the proven techniques to predict the performance of the coming software and responses in the least loss of time and cost when solving problems.

978-1-4244-7237-6/10/$26.00 ©2010 IEEE

B. SPE process

1) SPE internal process

Analysis of software performance bases on the performance model through the software performance model to measuring the software performance indicators. So the first task is to set up the software performance model. Traditionally, the establishment of performance model is in the late of the software development. The load parameter of the software already exists when the performance of the system established. While in the SPE because the performance model of software development established in the early days, at that time target software has not yet formed, we need to find out the construction features of the software and establish it. The competitive effects from the computer resources in different types of tasks quantify. We first create a software execution model if we want to build out features of the no-existing software. The model should demonstrate the key features of the expected behavior of the software. It will produce load parameters similar to the traditional analysis of performance.

According to the tradition, the performance of the existing systems analysis process include: study of computer system, building the system execution model, measure the current execution model, describing the load characteristic, developing input model, through the model to solve, comparing the system model result with the measured data to affirm the model, revising the model until the result consistent to the measured data.

Compared with the traditional, SPE started when the load in the model has not begun. We must build up SPE load parameters so as to generate performance models first. A typical SPE process includes: i assessing the performance risk; ii identifying the key tasks; iii describing each critical tasks; iv building up the performance aim; v constructing the performance model; vi identifying the software demands; vii increasing the computer resource demands; viii evaluating the model; ix verifying and affirming the model.

2) SPE outer process

From the view of software project management, performance analysis should be combined with the software development process. It is a part of the whole software development method. SPE is compatible with many software process models including the waterfall model, spiral model and the UML Process etc.

The following is the UML integrated SPE process. It emphasizes their mutual relationship among the development process in the system for performance analysis in different stages, that is to say that SPE has a characteristics of spiral development.

a) The UML having SPE

UML Process is the time of the life cycle repeating a series of component. The first time is the beginning of the life circle. The last time is the ending of the life circle. Each time contains four stages: Original, refinement, structure and transferring. Each stage contains a number of workflows. A typical workflow includes Requirements analysis, designing, implementation, and testing etc.

Each stage needs the iteration to eliminate the risks. That means the risk-driven iterations. There is a variety of risks, including the risks associated with performance. When the start of new projects plans the performance risk assessment can be carried out on the time planning the next iterations. By assessing the extent of use of new technologies, developer experience, and the complexity of operating environment and new software, flexibility expected to use and other factors we assess performance risk. When planning the next iteration, we need to assess the results of the previous iteration, determine whether performance risk exists. We analysis and evaluate in the initial stages of performance, the assessment results will lead to the next iteration.

b) Waterfall model integrated with SPE

In the waterfall model, the first performance analysis starts at that time software requirements end. It mainly assesses performance risks by assessing the extent of the use of the new technology, developer experience, the flexibility expected to use other factors.

The second performance analysis begins in the end of the structural design. It is main quantitative analysis for performance. The subsequent performance analysis mainly focuses on performance analysis of the results for tracking and controlling. We verify it with the development of the new achievement so as to make performance analysis more accurate. It is shown as figure 1.

In each stage of development there is performance analysis to ensure that developers can choose the plan to meet the performance objectives. The performance analysis of each stage integrates an entire system performance analysis into "the whole". It is the results through the use of project management and analytical techniques in order to integrate the various stages of performance analysis. It has a bigger advantage than traditionally focusing on the stage of implement.

By this approach, in the stage of implement once we find that the software performance analysis is not met with performance targets, then its will most likely return to the design stage to re-design. It will lead to the painful consequences such as increased investment, project delay or even cancel the project in the view of project management.

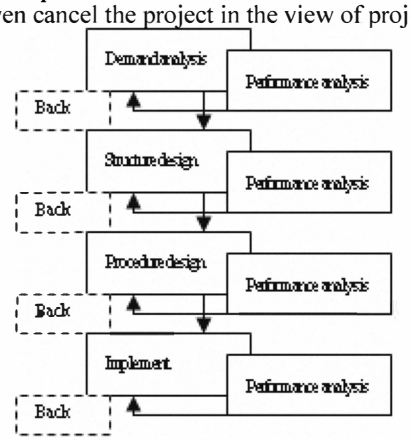

Figure 1. Waterfall model integrates SPE

In each stage of development there is performance analysis to ensure that developers can choose the plan to meet the performance objectives. The performance analysis of each stage integrates an entire system performance analysis into "the whole". It is the results through the use of project management and analytical techniques in order to integrate the various stages of performance analysis. It has a bigger advantage than traditionally focusing on the stage of implement.

By this approach, in the stage of implement once we find that the software performance analysis is not met with performance targets, then its will most likely return to the design stage to re-design. It will lead to the painful consequences such as increased investment, project delay or even cancel the project in the view of project management.

C. SPE model

SPE model provides the technical means for our implementation of the SPE process. It is the basis for performance analysis. The most basic model is a software implementation model of SPE model. For software has not been set up in the early the software implementation model is particularly critical. It provides the base and foundation for the load characteristics.

D. SPE Strategy

SPE uses modeling strategy to obtain quickly result, eliminate the error of software and hardware resource utilization and control costs. SPE strategy includes simple model strategy, the best and worst case strategy, adapting to precise strategy. SPE strategy provides a simple way for the rapid implementation of all stages of performance analysis. In the early time of software development, a simple model strategy starts from the simplest available mode. It recognizes the system architecture, design and realizes the target problem. Under the condition that resource requirements can not accurately assess the demand, the best and worst case strategies expects the best and worst performance assessment of the resource requirement to set up a performance lower bound and a performance higher bound and manage their uncertainty. With the development of software, it is possible to obtain more accurate performance with our knowledge and experience of the software process and details. Adaptive strategy is to make these possibilities into reality.

E. SPE tools

SPE tools help the performance engineers set up performance modeling, analyze and manage software systems. SPE tools include SPE operating platform, SPE model library, SPE diagram library and SPE database. They are shown in Figure 2. The current SPE products tools such as SPE-ED tool.

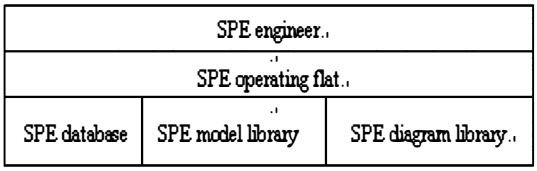

Figure 2. SPE tools component

III. SECURITY

Security is very important to modern software systems. But almost no development tools support software developers to design standard security software. System developers to model the system generally does not consider the security needs at that time. We extend UML, allowing developers to integrate security into the system at every stage of UML models. This does not affect the development of the original speed of the system. The aim Extending UML modeling to meet the security needs of modeling in the system is to test security needs at each stage of the modeling. Finally description of documents formed. It can be deployed with application program at the same time. We integrate security management system to maintain System security maintenance.

A. Access control in the system

System access control system supports object-oriented access control strategy design and management model. It is based on the view of the access control. It can be seen as role-based access control in the expansion plans. VBAC model is simple. VBAC can be seen from Figure 3, the entity is subject, role, view, and objects. A subject can play different roles or a role played by some subject. A role may have some different views or a view may be owned by different role. But a view always corresponds to an object. An object can have multiple views. The main features of VBAC are that viewing language as a static language. It can be use as the construction of fine-grained access control permissions, the access control permissions on the object the operation is authorized or refused.

B. Access control policy and system integration modeling

The following steps show the security needs of the development process of integration with the UML: i System requirements analysis stage, through case diagram of the system that functionally modeling to describe the system's functional requirements. ii Security requirement analysis stage, by extending the UML-based version of associated and non-formal comments to clearly indicated that access to system resources. iii System design stage, mapping the security requirements analysis stage of access to the logical view UML class diagram. iv Safety design stage, the participants show with a view access to system resources.

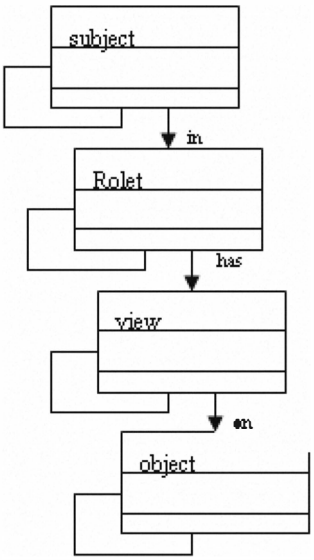

Figure 3. VBAC model

The following will demonstrate with student achievement management system, based on the above steps to illustrate the modeling of systems development and integration of security requirements, access control policy language specification modeling and generation of VPL.

1) Security analysis

a) Exampled by student performance management system

Student performance management system mainly achieves three main functions: student inquiry results, teachers input grade and administrators do the statistical grade entry. The system relates to the role (corresponding to actor in the UML) including Student, Teacher and Administrator. In figure 4 it is the student performance management system use case diagram.

Student can look over their own grades. Teacher inputs the student grades. They can also look over student grade. Administrator is responsible for data statistics, although inheritance from teacher class, in accordance with the system security requirements, the administrator can not input the student grade. He can only look over the grades.

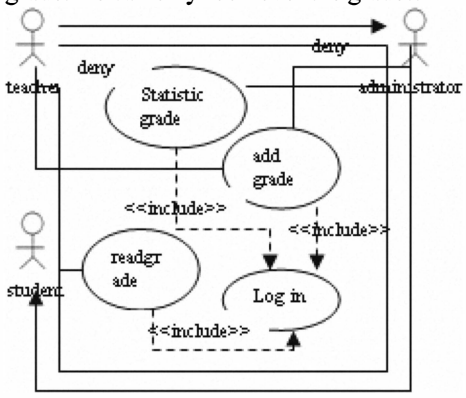

Figure 4. Use case diagram

If the role A inherits from B, communicate with the A is able to communicate with B. For example: Teacher can communicate with use case "readgrade". Because Student is associated with this use case, student, teacher and administrator want to enter system they must first verify the identity. So use case statisticgrade, addgrade and readgrade and login is included Relationship. System use case analysis is the first step for the analysis of system security.

b) UML role, access control role and deny use case

The UML use case participants are the actors to communicate with use case. UML association role is modeling by the two endpoints. Actors and use case make an UML role certain.

In Figure 4 Administrator inherits Teacher, Teacher can access to the use case addGrade. According to the semantics of inheritance, Administrator can also access use case addGrade. But because of the safety, this access is not feasible. The system requires that only teacher has the right to access use case addGrade. Therefore, addGrade is disabled to the Administrator. For identifying the deny use case, linkages between expanded roles and use cases <<stereotype>> is <<deny>>. <<Deny>> means the association is deny.

c) Note of function and access control

System developers model to the system's security through the use case model authority relationships described in the roles between use cases and roles. Note in the use case diagram describes the role of system access with a high-level, non-formal way. It is a clear description of the system security information.

2) Access policy model

Use case diagram is not a complete description of the demand description and non-formal. Finger 5 is this system's class diagram. It maps the note to the different interface. Access control policy design is the control operations mapping the security note in the use case diagram to the logical model of the interface or class of security. However, the class diagram does not regulate the operation permission of the object.

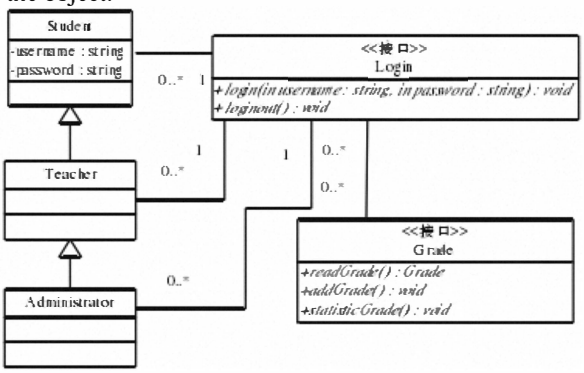

Figure 5. student grade management system class diagram

a) Access control view model

Access control View standardizes the permissions that the role called object. Semantics that view gives the role is a

view specification of access rights. View describes the static access rights.

The access control view is composed of three parts. Part one is view note. It can be defined by the user interaction or automatically generated. Part 2 shows a limited role of the view. According to the inheritance relationship between roles, this part can be simplified. The least roles make a list of the involved numbers. The last part contains the access to this view. Having the same access to view is considered equivalent. View may inherit each other.

Access control view model organizes all views together. Figure 6 respectively shows the interfaces Grade and Login view model.

Access control view model lets the system developer check a given role having what kind of access rights. The significance of the access control view is that we can early detect the logical model and analyze the matches between models.

b) Static access control model

The role view describes the relationship between the role relationship among roles and the right given to role by the static access control model. Object "O" `s view "V" authorizes the role "R". This means that each user of role "R" has the view on the object "O". Therefore, they have access to the view "V".

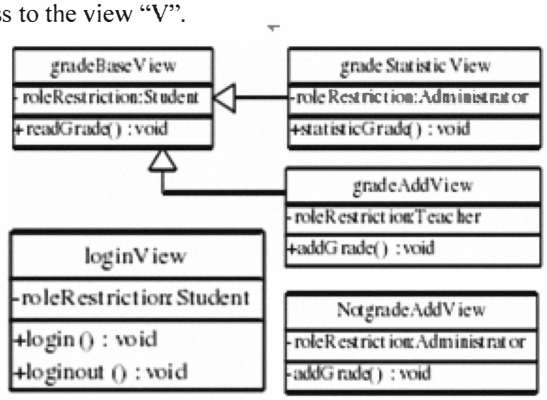

Figure 6. Interface grade&login view model

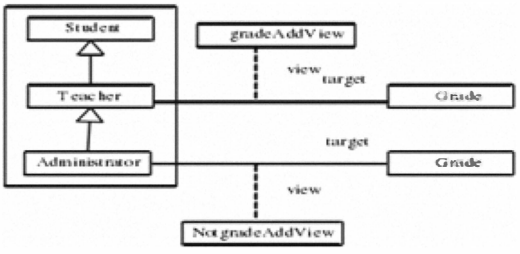

Figure 7. Static access control model

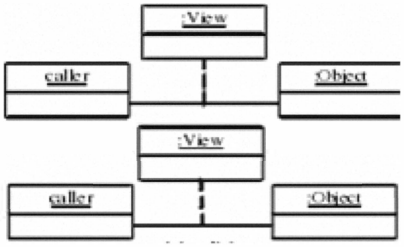

Figure 8. Schema figure

c) Dynamic access control model

The VPL schema figure sets up the model for the dynamic access control. It describes that a given operation being called brings up the role authorization changes. In figure 8 callers is the main body of the object operation. This figure means that once the operation to the object when having the view finished caller loses the view of the object. That is to say authorization. This figure expresses the change of the dynamic access control authorization.

The schema figure describes if caller has the given view after the implementation. The schema figure represents the operation results with key word "result". Then we can generate the VPL policy. Many UML tools can import XML from UML schema.

We alter the traditional UML to integration with security model based on SPE. We can make the software system more robust with this system.

REFERENCES

[1] C Szyperski, Component Software: Beyond Object-Oriented Programming. Addison Wesley, 1998.

[2] C. V. Smith, "Increasing Informatin Systems Productivityby Software Performance Engineering", GMG Conf[C],1981, pp5-14.

[3] S. Vadhiyar, J. Dongarra, "Self Adaptivity in GridComputing, Special Issue of Concurrency", Practice andExperience on Grid Performance, 2005,17(2-4), pp235-257.

[4] M .Koch, P.Presicce,Access Control Policy Specification in UML ,2002, www.inf.fuberlin/~mkoch/uml02GG.ps.

[5] R.Pooly, Software engineering and performance:a road-map In Finkelstein,A.,ed.The Future of Software Engineering.ACMPress, 2000, pp189~199.

[6] U.Smith, G.Williams, Building responsive and scalable Web applications, In Proceedings of the Computer ManagementGroup Conference,2000, http://www.perfeng.com/papers/webperf.pdf.

[7] C.Sun, A.Ellis, Operational transformation in group editors :issues, algorithms, and achievements, In Proceedings of the ACMConference on CSCW,ACM Press, 1998, pp59~68.

[8] F. Ferraiolo, S.Sandhu, I. Gavrila, Chan-dramouliR. Proposed NIST standard for role-based accesscontrol. ACM Transactions on Information and System Secu-rity, 2001, 4(3), pp224-274.