

The Research of the Structure and Permission of Workflow Based on AOP

Zheng HaiBei^{a,*1}, Yin Xu^{a,*2}

^a *Department of Computer Science, Beijing Information & Science Technology University, Beijing 100192, China*

Abstract

After analyzing the basic concept and method of Aspect Oriented Programming (AOP), the method based on AOP was presented to solved the security validate, permissions verification, log and so on. Compared to the conventional Object Oriented (OO) approach, AOP technology could avoid such problems as tangled code and scattered code, and had obvious advantages in raising the component reusability, optimizing processes and reducing the system complexity. Finally, an integration example of permissions verification in workflow system was provided, and the implement based on AOP was provided too.

Index Terms: Aspect-oriented; Concerns; Workflow; Permissions

© 2011 Published by MECS Publisher. Selection and/or peer review under responsibility of the International Conference on E-Business System and Education Technology

1. Introduction

By now most developments of software system are implemented based on object-oriented programming. They design some functional characteristic or requirement of the software as a specific module[1], which improves the code reusability, makes the different modules functioning separately and reduce the coupling of different functional modules. However some characteristic and requirement of the software system are crosscutting in all layers of the system and every module of the system. These character are called aspect or concerns of the system, for instance the business logic of the system, the durability of data, the reliability of system(such as log, transactional ability, exception handing and recovery), authentication, safety problem and error detection etc. To some extent, a complex software system can be considered to be consist of implementation of many concerns.

The method of object-oriented programming which regards the software system as a set of classes and objects is to divide a complex system into separate modules[1]. However, as those concerns are crosscutting many modules and this could probably get many modules involved in some repeating codes. This may cause tangled code and scattered code or make the procedure of software development untraceable , inefficient and the code reusability bad and the scalability poor ,which directly make the system hard to develop and maintain.

Aspect Oriented Programming is aimed to solve those problems

Corresponding author:

E-mail address: ^{*1} zseashell@163.com; ^{*2} yinxu15@126.com

2. The Development Method of AOP

The technology of Aspect Oriented Programming which is derived from the technology of Object Oriented Programming can add uniform function to the program dynamically without changing the source code by using pre-compile and dynamic proxy policy[2]. Actually AOP is originated from Gang of four(GoF) design pattern.GoF aims to make the invoker and the callee coupled and so is AOP. The most fundamental principle is to provide a mechanism to describe the crosscutting concerns and to get the crosscutting concerns merge into the object-oriented programming system. In this way it modularize those crosscutting concerns to make the aspect can act as same as object to execute add, delete, modify and reuse function. Compared to class whose relation with each other mainly reflect generalization and specialization or simple connectivity between classes, aspect can describe more complex relevance.

In general, aspect oriented programming contains three procedure, namely Aspectual decomposition, Concern implementation and Aspectual weaving.

- Aspectual decomposition: to recognize which concerns are crosscutting the whole system and which are public concerns of system; to differentiate module-level concerns from concerns crosscutting the whole system.
- Concern implementation: implement every concern respectively.
- Aspectual weaving: to weave or integrate different functional modules according to the rules of every aspect of concern to finally form a system containing all concerns and function requirements.

3. Workflow Management System Based on AOP

A mature workflow system is a platform used to develop and deploy software and also used to run application software of workflow[3]. It not only provide tools which make development of workflow application software fast to help developer redevelop the workflow application system based on workflow platform, but also provide the necessary runtime environment to support the workflow system running. In a workflow system concerns of application requirement include transaction flow, transaction activities and participant etc. And the task during the constructing section is to implement and weave the concerns of workflow. During the functioning

and interactive stage , analyzed by workflow engine the workflow system will be running according to the relation which is built after the implementation of transaction flow, transaction activities and participant.

Based on the thoughts above-mentioned, we combine the technical platform(Struts + Spring + Hibernate) based on Java 2 Platform Enterprise Edition(J2EE) and OSworkflow, an open source workflow engine to generate a workflow management system based on AOP.

3.1. the system structure of workflow system based on AOP

OSWorkflow is a representative of lightweight system and also a very flexible workflow implementation framework which is positioning at a low level[4]. what positioning at low level means that it is not designed to solve the problem of flow model object or functioning scene but provide a maintainable dispatching mechanism to make sure that the developer can expand the system on their own. As the theory of OSWorkflow is Finite State Machine (FSM) which is based on action, it makes OSWorkflow much more like a complicated but flexible finite states dispatcher.

According to the rules of J2EE standard regard to the restraint of the component in transaction logic layer, the workflow management function just act as a new component which should be deployed in the middle layer

and won't affect other transactional component. Fig.1 shows a workflow software frame which integrate OSWorkflow engine.

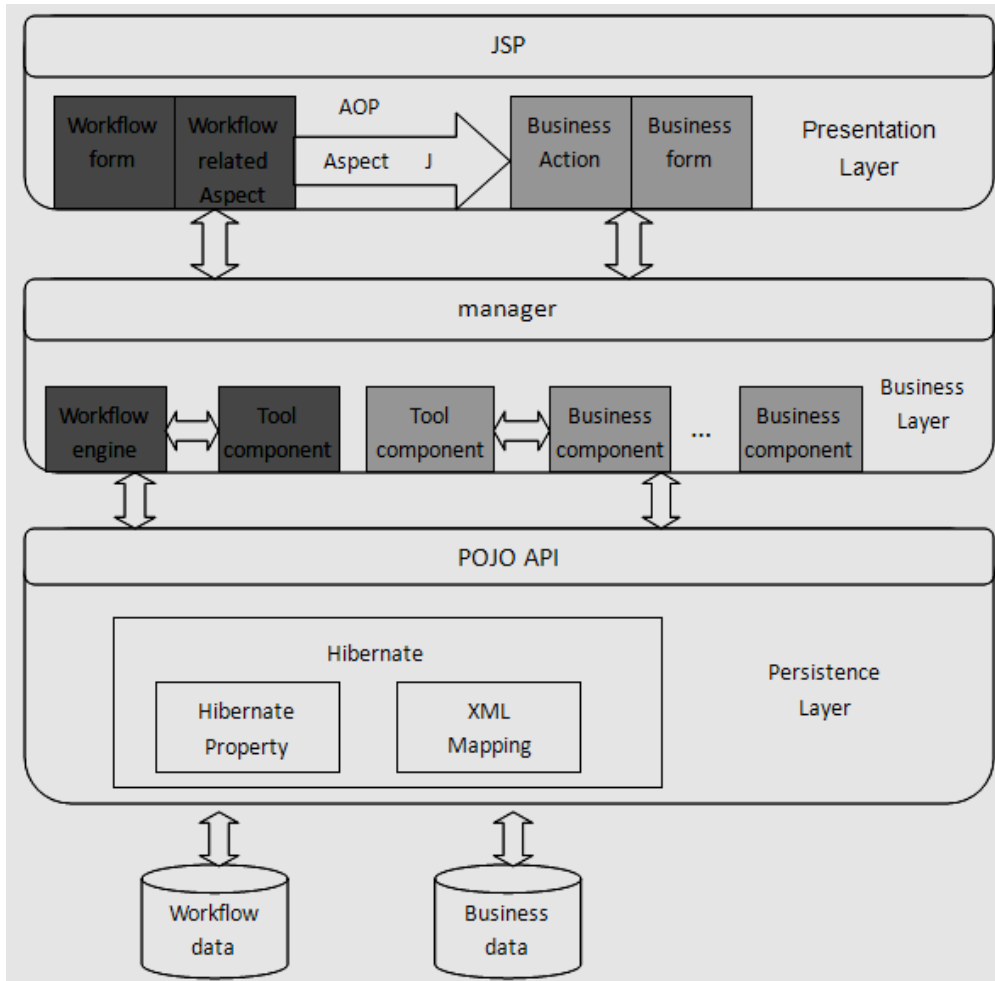


Fig. 1. The system structure of workflow system based on AOP

In the business layer the components with light color belongs to the system that already exist., the components with deep color represent components related to workflow engine which need not modify. In the presentation layer the components with light color are STRUTS components of Existing application systems, which include Action class and Form Bean; the components with deep color represent components related to workflow which don't use STRUTS technology yet still use form and other class files that is similar to action. The interaction of the two parts is unavoidable and in this essay AOP technology is used as for the interaction. From the picture we can see that when we add some function related to workflow to the presentation layer we don't modify the Action class of the already-exited system. AOP can separate the concerns which constitute the system and maximum the reusability of business components in business layer, action class and form of the presentation layer in this system.

3.2. the improvement of permissions verification

In workflow management system, if we want to reflect the granularity of permissions management in method of every transaction, we need to write permissions check code for every functional class. And this mix program function with permissions verification, consequently brings coupling problem and make it hard to expand or modify. The traditional pseudo-code is as follows:

```

public class BusinessEntity{
public SomeFunciton( ) { //authority judging
User user = context.getUser( );
if( user.canExecuteThisFunction( ) ) { // do the
business method
// ... } else {
throw new PermissionDeniedException( );
}
}
}

```

We can separate authority management out of concerns by using AOP technology. When we execute someFunciton() or other functions ,the system will verify the user's permissions and invoke crosscutting concerns logic , and this is join point of AOP. For a software system every method which need permissions verification is a separate joint point. Because permissions verification is executed before the execution of every method, so as for those series of joint point we just need to define one pointcut and then execute the logic which is needed for the implementation of crosscutting concerns, namely advice. While pointcut and advice constitute a aspect of permissions verification. We can define a aspect like this:

```

private static aspect AuthorizationAspect{
private pointcut authorizationExecution( ):
execution( public void
BusinessEntity1.SomeFunction1( ) ) ||
execution ( public void
BusinessEntity1.SomeFunction2( ) ) ||
execution( public void
BusinessEntity2.SomeFunction1( ) ) ||
execution ( public void
BusinessEntity2.SomeFunction2( ) );
before( ) :authorizationExecution( )
{
if ( ! permissions.Verify
( Permission.ADMIN )
{
throw new UnauthorizedExcepion(
);
}
}
}

```

By using such a integrated aspect, when the system invoke some relevant methods of BusinessEntity1 or BusinessEntity2 ,the pointcut is triggered and the system will invoke he corresponding advice logic. By this

means `SomeFunction1()` and `SomeFunction2()` become completely independent with permissions verification module and at the same time eliminate the unavoidable repeating codes of permissions judging in the traditional designing. This is advantageous in building a loosely coupled and scalable system.

4. Conclusion

This essay analyzes the characters of AOP technology and adopt methods based on AOP to solve some concerns problem of system level in workflow system. AOP brings in Aspect ,which can package several class behaviors into a reusable module and allow programmers to modulize those crosscutting concerns. This eliminates the tangled code and scattered code problem of (Object Oriented Programming) OOP and improve the maintenance of the system and reusability of codes. We must follow two points when we design and develop system based on AOP :first, if a section of codes can be encapsulated then we regard it as a class ,an object or an API. As a basic unit, it is easy to manage and assemble. while if a section of codes can't be encapsulated properly but also used in many functional modules, then we regard it as a aspect instead of a system functional unit[5]. Second if we try to implement a workflow software system based on AOP ,when any aspect of system requirements changes what we need to do to update and maintain the system is just to renew the implementation of the concern and reweave it with other aspect concerns which have already exit. And the operation of implementation and weaving can be done by auxiliary tools of software development. Because we need not to writing codes about bottom layer of the system , it provides a new and effective way to improve and optimize the flow of whole system. The workflow management system is getting more and more mature ,the main application fields is application system left by those integrated company. By using AOP technology we can avoid intruding into the original system when performing integration action thus improve the reusability of the application system.

Acknowledgements

My research is supported by Science and Technology Funding Project of Beijing Education Committee

References

- [1] Grady Booch, Feng BoQing, Feng Lan, Xue Tao. Analysis and design of object oriented programming[M], 2002. (in Chinese)
- [2] Laddad R. I Want My AOP (JavaWorld) . <http://www.javaworld.com/javaworld/jw20422002/aspect3/jw204122aspect3.zip> ,2002
- [3] DongWeiYun, haoKeGang. An Implementation of workflow management system based on AOSD[J]. Journal of Computer Science, 2006,33(8) (in Chinese)
- [4] OSWorkflowTeam, ChenGang. OSWorkflow(inChinese)[M]. <http://www.infoq.com/cn/minibooks/>
- [5] Sun ZhongYi, He Yi, Cai HongMing, Jiang LiHong. The construction of a dynamically rebuildable workflow engine based on AOP.33(6) (in Chinese)