



2011 International Conference on Physics Science and Technology (ICPST 2011) Based Aspect-oriented Petri Nets in Software Engineering

Wensong HU^a, Xingui YANG^a, Ke Zuo^a

^aNanchang University, Software School, Nanjing Dong Lu 235, 330047 Jiangxi, China
Yangxingui88@126.com, 461284463@qq.com

Abstract

Aspect Oriented (Aspect-Oriented, referred to as AO) as a new programming technology is increasingly cause for concern. This article describes a number of experts to study the current object-oriented Petri Nets (OO PN) adding aspect-oriented thinking, combined with software design and development cycle, given the aspect-oriented OO PN in software engineering methods and steps. Shows the method of using AO PN government office system software design and development of application examples, and gives some object class, the log section and the application form. As the plane of isolation, reducing the coupling, the use of AO PN ways in different applications will use a combination of each section, allowing code reusability enhancement. OOPN itself can process the software system design and development of effective control to ensure that the software system reliability and standardization.

© 2011 Published by Elsevier B.V. Open access under [CC BY-NC-ND license](http://creativecommons.org/licenses/by-nc-nd/3.0/).
Selection and/or peer-review under responsibility of Garry Lee.

Keywords: aspect-oriented; object-oriented; Petri net;

1. Introduction

Petri method has been widely used C IM system design, analysis and implementation. It provides a complex manufacturing system on the effective analysis and implementation. In recent years, Petri net methods have been used for software development, verification. Meta software company developed a series of software. The software can be integrated SAADT (IDEF0) model and the hierarchical colored Petri nets, resulting in an integrated software development methodology. A lot of literature on the Petri net method is applied to real-time control software development, pointed out that this Petri net structure can be data-driven and based on arbitrary standards to achieve high-level language. This paper presents a method based on Petri net software development process. At the same time a lot of literature on Petri Nets in Software Engineering, System Design, is given based on predicate / transition net approach represents a dynamic system characteristics.

Object-oriented technology is currently perceived as solutions to complex problems one of the most promising methods, considering the Colored Petri Net, "color" of the abstraction mechanisms in object-oriented classes, encapsulation, inheritance and other concepts of integration the emergence of object-oriented Petri Nets (OO PN) method. OO PN researchers from these studies, the basic conclusions can be drawn: Object-oriented model and Petri net model of consistency; the Petri net model into object-oriented development methods, object-oriented approach can overcome the lack of model analysis, validation shortcomings of insufficient means;

The introduction of object-oriented methods in Petri net theory, Petri nets can be overcome in the description of complex system deficiencies.

However, the aspect-oriented ideas into OOPN theory, mainly based on object-oriented technology is a kind of object-based, event-driven programming object execution process technology of object-oriented techniques to describe the objective world of objects and their the success of mutual relations, but it addresses such as user permissions check, logging, transactions, etc., then obviously more blunt. in order to check user rights, for example, an object before accessing the resources necessary to carry out authorization checks in the object-oriented design, permissions checking will be the behavior of objects of these resources, so permission checks will be distributed among multiple objects in the system once permission checking strategies with user requirements change, the need to modify the class

files scattered in various places, modify volume, increasing the probability of error, and increased the difficulty of system maintenance shortcomings; through different facets, can be different levels of the system to isolate the problem to achieve the reunification of intensive treatment so the benefits are : on the one hand makes the development of logic clearer, easier to specialization; the other hand, the isolation section, reducing the coupling, in all different facets of the application will be used in combination, allowing code reusability increased.

2. The introduction object-oriented Petri net approach

OPNets model

oPNets (object-oriented high-level Petri Nets) Korea KAISI 'of Yang Kyu Lee and other proposed

One kind of high-level system modeling for real-time Petri net model. It is presented in order to use object-oriented ideas to solve a simple Petri net description of the shortcomings of the system size is too large, so that the Petri net can be widely used for system modeling analysis and simulation.

In OPNets, the senior network subnet with the template that describes the object classes, subnets by using a square box to enclose that package and abstract. The external interface part of the object by the "message queue" (messagoqueue, referred mesQueue, with ovals, similar to that of the circular Place), "gate" (gate, with thick lines, with a square box that is similar to the transition) and flow relations between them (ar., with the arc that) given. Each class can have a number of instances, respectively (indicated by black dots, that token), each instance in which the Place. That their current state, so they called the Place particular state.

The internal behavior of an object from a restricted network description of the predicate. Network in the precursor and successor in the transition can only have up to a state, thus prohibiting the use of a different instance than in the internal messaging but the possibility of direct interaction. Arcs without the predicate, and each transition will occur when the conditions and the occurrence of the definition of actions to be performed. When all of the precursors in the transition have a token of the token and there is a combination of the transition occurs the condition is true, transition can occur.

Between different objects can be used to input mesQueue gate connected to said output mesQueue messaging each other the relationship.

Class can define properties, transition conditions for the occurrence of read access to an instance occurred in property values, and execution of transition actions can be read-write property value. Messages sent between objects are structured data types. The transition occurred in the message will be destroyed, and all subsequent messages in the output queue, a message is created, its property value by the transition of executive action to set.

Simple class of compound classes and class distinction. In the simple class does not contain complicated part, only used to represent sequential behavior; in the composite class allows concurrent, because the composite class act by a separate class of simple or complex structure made of the control distribution in each class, According to system requirements and can synchronize the internal structure of behavior. These simple classes within the category referred to as internal or compound classes. Target system model is the interconnection of multiple object instances of classes defined in the group Together.

3. The application in software engineering

OO PN with aspect-oriented software system designed for modeling and object-oriented systems analysis methods to be followed when dealing with complex problems of the basic principles of the core concerns and non-core concerns separate and form section:

1. Of the system under study at this level of abstraction for a number of "objects" to create an object model of the subnet, the message library is connected with different objects;
2. It is provided on the external news subnet object library, and hide all the libraries within the reach of its changes;
3. all the colors the same subnet object, that object instances are inherited change the subnet structure, marked by the color difference information and carry different objects;
4. In dealing with complex issues, research should pay particular attention to the problem domain object classification. The unity of the object points system is conducive to software systems analysis and synthesis. Implementation of software development using OO PN basic steps are as follows:
 1. The concept of the establishment. Software to be developed to determine the overall objective, given its features, performance, reliability, and interface aspects of the vision; determine whether to use the OO methods, and discuss the feasibility assessment.
 2. Determine the physical reality of the system objects and their relationship with the surrounding environment. Clear from the reality of the system classes, objects (their roles, responsibilities, collaborative relationship), will implement the system as a system service function module unit of the horizontal decomposition, scattered in the function module

function and behavior to form a horizontal section, and system modeling them. Scene description described by the main function of the system point (need not be completely comprehensive, but not miss important function points).

3. In the second step, based on the abstract class, all kinds of explicit message passing relations between objects.

4. The establishment of OO PN model. The establishment of the message library, according to the relationship between the object information networks diagram drawn OO PN; specific types of object's internal data and methods to generate subnet object for each type of object, according to the study of levels of refinement; the various instances of objects different set of colors that determine the initial token.

5. The idea to build an aspect-oriented model validation. Verify whether the correct message is sent between objects; verify whether the package subnet object intact, whether the network is running deadlock and so on.

6. Software programming, the use of aspect-oriented programming language for cross-cutting concerns (section) program, the OO PN model into a high-level programming languages such as Java.

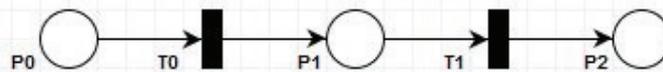
7. Software testing. Test whether the software can complete the required functions; test the reliability of software.

8. Operation and maintenance. Management systems, local modification; updated version of the product.

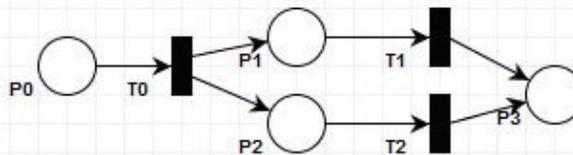
Based on the above basic steps to complete a software project throughout the life cycle of work.

Technology Technology Board for approval to the project application report, for example. Work process from submission, through the approval process, the end of the last filing. Throughout the approval process may be eight in the state: to submit application materials, registration, approval and registration form, approved by, the Department signed, at a higher level, filing, archiving. The activity will translate into changes, and the application and filing separately as a start condition and end conditions, with the library said. After analysis, application, registration, approval, approved these four activities can be executed in order, signed and submitted to the Meeting asked to choose two activities implemented after the approval of activities may be two results: approve or refuse an applicant after the notification email re-apply. Therefore, the results can refuse treatment as a cycle control relationship. E-mail throughout the process as a trigger mechanism.

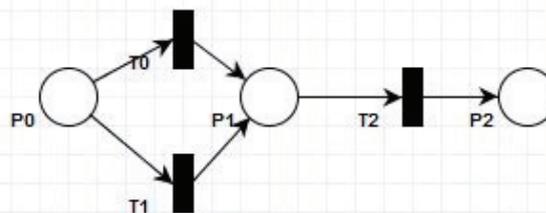
Petri net structure can be expressed in seven of the control flow structure are in order, and the connection structure, or connection structure, and division structure, or split structure, loops and nested structure, institutional audit system, including control relations: sequence , concurrency, choice, loop. With that of the four Petri net structure is shown below.



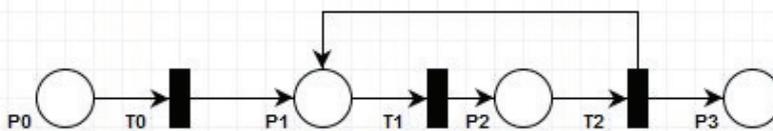
(a)sequence structure



(b)parallel structure



(c)Selection structure



(d)Loop structure

Example: department1, department2, department3, department4;

Send a message: load, disposed of;

Receive messages: handling, processing is completed, review completed;

Message library: Loading OU T, disposed OU T, treatment

IN, disposed IN, processed IN;

Method: department free / busy judge and handle materials, the work is completed by the relevant departments

Become busy.

The object-oriented programming is described as:

```
public class Department
{
    private static final Department Department = null;
    private int departmentId;
    private String departmentName;
    private String departmentState;
    PlaceList Places;
    TransitionList Transitions;
    public Department IdleOrBusy(int departmentId)
    {
        while(departmentId!=departmentId||departmentState=="busy")
            Wait();//if department isn't available,then wait
        return Department; //send message to department to load
    }
    public Process();
    public BusyToldle(departmentId);
}

```

Object internal operations (methods) of program implementation can be achieved with the direct object-oriented language, you can also change according to each subnet object and trigger the movement of said token. Here are the library and change the definition of object-oriented, they are in the program design is very useful to reduce redundancy and greatly increase the reusability of the program.

```
public class PlaceList {
    private int PlaceId;
    private String placeName;
    TransitionList InPlaces;
    TransitionList OutPlaces;
    Coloset AllowedColors;
    TokenList IniTialToken;
    TokenList CurrentToken;
    public PlaceList AddToken(int TransitionId);
    public PlaceList MoveToken(int TransitionId);
}

```

```
public class Transition{
    private int TransitionId;
    private String TransitionName;
    PlaceList InPlaces;
    PlaceList OutPlaces;
    public boolean AllMarked();
    public boolean CanFire(){
        float p;
        if(!AllMarked())
            return false;
        p=FiringProbability();
        if(p==0) return false;
        if(Condition of this transition is not satisfied)
            return false;
        return true;
    }
    public float FiringProbability();
    public fire(){
        if(CanFire())

```

```

    {
        InPlace.MoveToken(TransitionId);
        OutPlace.AddToken(TransitionId);
    }
    else
        System.out.println("Can't Fire!");
}

```

With the changes in the class library and define the future, they can be derived from the common library, the message library, general change, messaging, news and other changes in the class definition. As the object inside the method OOPN subnet object is a series of changes in the composition of the library and, therefore, an object within the method implementation can be seen as the result of changes to be triggered individually. Thus, the object's internal methods of procedure by changing the order of trigger operation is complete.

By this method, step by step approach to each object to write the program. Finally, define a management messaging between the various objects of the main program, the entire software system design and implementation process is complete. It can be seen, the concept of object-oriented design and object-oriented programming language introduced, many have designed block or defined classes are available for reference, greatly improving the speed of program development. Dynamic proxy mechanism, with the dynamic proxy address issues such as user rights checking, logging, transactions and other aspects of non-core concerns. The following simple with a simple log interceptor, it begins to com.yxg java packages in all methods for log processing.

```

public class LogInterceptor {
    @Before("execution(public * com.yxg..*.*(..)")
    public void before()
    {
        System.out.println("start method!");
    }
    @Around("execution(public * com.yxg..*.*(..)")
    public void aroundMethod(ProceedingJoinPoint pjp) throws Throwable{
        System.out.println("method around start");
        pjp.proceed();
        System.out.println("method around end");
    }
}

```

4. Summary

This article describes the use of aspect-oriented to object-oriented Petri network, and apply it to application software design and development process. Gives the example of the government office software design. Establish with OOP system is implemented with loosely coupled modular general concern to achieve. AOPN not to replace OOPN, AOPN is a cross-section of some logical OOPN approach against the backdrop of the poor, it OOPN had a very good complement. AOPN and OOPN two complement each other, in general, OOPN level for core business concerns, that the business logic design, and AOPN for system-level concerns such as logging, transactions, permissions checks, etc. design. the use of OOPN approach biggest advantage is the design development process of software systems for effective control, to ensure the reliability of software systems and the level of standardization.

References

- [1] Kiczales G, Lamping J, Mendhekar A, et al. Aspect-oriented programming[J]. LNCS, 1997, 1241:220-242.
- [2] Jorgensen, P. and Erickson, C. Object-Oriented Integration Testing. Commun. ACM 37, 9 (September 1994), 30-38.
- [3] Murphy, G. C., Townsend, G. and Wong, S. W. Experience with Cluster and Class Testing. Commun. ACM 37, 9 (September 1994), 39-47.
- [4] Kung, D., Gao, J., Hsia, P., Toyoshima, Y., Chen, C., Kim, Y. S., and Song, Y. K. Developing an
- [5] Object-Oriented Software Testing and Maintenance Environment. Commun. ACM 38, 10 (October 1995), 75-87.