J. Marine. Sci. Appl. (2008) 7:273-277
DOI: 10.1007/s11804-008-7096-7

Applying a semantic information Petri Net modeling method to AUV systems design

FENG Xiao-ning[1*] (冯晓宁), **WANG Shuo**[2] (王 朔), **WANG Zhuo**[1] (王 卓), **LIU Qun**[1] (刘 群)

1. Department of Computer Science & Technology, Harbin Engineering University, Harbin 150001, China
2. Chu Kochen Honors College, Zhejiang University, Hangzhou 310058, China

Abstract: This paper informally introduces colored object-oriented Petri Nets(COOPN) with the application of the AUV system. According to the characteristic of the AUV system's running environment, the object-oriented method is used in this paper not only to dispart system modules but also construct the refined running model of AUV system, then the colored Petri Net method is used to establish hierarchically detailed model in order to get the performance analyzing information of the system. After analyzing the model implementation, the errors of architecture designing and function realization can be found. If the errors can be modified on time, the experiment time in the pool can be reduced and the cost can be saved.

Keywords: autonomous underwater vehicle (AUV); colored Petri Net modeling language (CPNML); substitution transition; reachable tree
CLC number: TP39 **Document code:** A **Article ID:** 1671-9433(2008)04-0273-05

1 Introduction

The autonomous underwater vehicle (AUV) system belongs to complicated nonlinear system. The system operating environment can be influenced by many factors such as the pressure of the underwater environment and the rapid attenuation of the radio wave and light wave in the water. So the control of the AUV navigating, the self-protection, the environment recognition and modeling method of the system are more important and more difficult than other systems. On the other hand, the experiments of AUV in the pool are very strict with the devices and environment. And the cost of the experiments is very high. So it is very significant to establish the function and performance emulator model of the AUV system and to analyze with it. At present, the main modeling methods of the AUV system are hierarchally analyzed by the object-oriented modeling method[1].

The Petri Net theory has been widely used in many areas. It belongs to the mathematics modeling method of graph. It is a useful tool in description of information management and modeling. With the development of Petri Net theory, many other types of Petri Net theories were put forward, such as time Petri Net (TPN), colored Petri Net (CPN)[2], object-oriented Petri Net(OOPN), predication/ transition Petri Net (P/TPN) and so on, most of which were called "high level Petri Net" orienting towards the engineering applications[3]. The application scale is increasingly extending. According to the structure and performance of the AUV system, the colored object-oriented Petri Net (COOPN) is informally introduced as the modeling method of the AUV system in this paper. This application has resolved two questions, the first one is the classification of the AUV function model by object-oriented method, and the second one is the description of inner function realization model with the substitution transition in the Colored Petri Net method[4].

2 Colored object-oriented Petri Net

Traditionally, Petri Nets are drawn with ovals representing places (which hold tokens), rectangles representing transitions (which change the distribution of the tokens and hence the state of the net), and directed arcs (which indicate how transitions affect neighboring places)[5]. For CPN, tokens have an associated data value. Places are then annotated to

Received date: 2007-12-28.
Foundation item: Supported by the Foundation of Harbin Engineering University Foundation under Grant No. HEUFT05035.
***Corresponding author Email:** fengxiaoning@hrbeu.edu.cn

indicate what type of value they hold, arcs indicate which values are consumed or generated (possibly by use of variables), and transitions can have a guard to further constrain their enabling[6].

COOPN applied the color set, substitution transitions and variable bound conceptions in the Petri Net implementation. So the property state changing can be learned with the object-oriented visual characteristic.

Definition 1 The COOPN can be defined as a duality group:

$$\text{COOPN} = \{\text{Obj}, R\},$$

where $\text{Obj} = \{\text{Obj}_i \mid i = 1, 2, \cdots, n\}$ represents the objects set; $R = \{R_{ij} \mid i, j = 1, 2, \cdots, m\}$ represents the relation set of the objects;

The communication relation between the objects can be represented by

$$R_{ij} = \{\text{OMP}_i, \text{Gate}_{ij}, \text{IMP}_j \mid i \neq j\},$$

here Gate_{ij} is a special transition called "Gate" Transition, which can be used as the function of message transmitting.

Definition 2 Object Obj_i is a colored Petri Net:

$$O_{\text{Net}} = \{\Sigma, P, T, A, N, C, G, E, I\}$$

where

Σ represents a non-empty Token color set;

P represents the finite place set:

$$P = \{P_{\text{P}}, P_{\text{M}}\},$$

P_{P} represents the basic state place se; $P_{\text{M}} = \{P_{\text{MO}}, P_{\text{MI}}\}$ represents input-output message place used as the message transferring between the Objects;

T represents the finite transition set;

A represents the finite arcs set;

$$P \cap T = PA = TA = \Phi;$$

N is the node function, which can be defined as

$$N = \{A \to (P \times T) \cup (T \times P)\};$$

C represents $P \to T$ as the color function;

G is a defence function mapped T to Boolean:

$$\forall t \in T: \text{Type}(G(t)) = B \wedge \text{Type}(\text{Var}(G(t))) \subset \Sigma;$$

E is an arc expression function defined as A to $E(a)$, where

$$\forall a \in A: \{\text{Type}(E(a)) = C(p)_{MS} \wedge \text{Tpye}(\text{Var}(E(a))) \subseteq \Sigma\};$$

I is an initial function mapped I to P, where

$$\forall p \in P: \text{Type}(I(p)) = C(p)_{\text{MS}}.$$

Note: B represents the Boolean variation as true or false; $C(p)_{\text{MS}}$ represents the multi-set in the color set $C(p)$.

Definition 3 ONet transition set

$$T = \{T_P, T_S\},$$

where, T_P represents the basic transition as well as the transition in colored Petri Net; T_S is the substitution transition that can be used to describe the complete net structure to realize the detailed hierarchical system model.

The example in Definition 3 is an instance of the substitution transition. The example uses the motivity subsystem as a substitution transition.

3 AUV system modeling and analyzing

According to the system architecture and running features of the AUV system, the whole modeling process is divided from top to bottom. The process can be divided into three steps:

Step 1: Analyzing the architecture and running features of the AUV system, the system can be divided into several associated subsystems (control subsystem, real-time inspection subsystem, navigation location subsystem, motivity subsystem, thruster subsystem and information measuring subsystem). These subsystems can be modeled with subobjects. The communication between difference objects can be implemented by the message places and gate transition. Thus the AUV system layer model can be obtained.

Step 2: Modeling every subsystem into subobjects depends on the definition of COOPN. Then the model ONet of the subsystems can be obtained.

Step 3: The instances of each subsystem will be carefully analyzed. And with the replacement of the substitution transition the detailed process of the function implementation will be known.

3.1 The modeling of AUV system layer

This part takes the motivity subsystem of the AUV system as an example. As the AUV moves, the decision-making is made by the control system. The control subsystem sends out the order. The order is transferred to the motivity subsystem. The order requires the motivity subsystem to supply energy. This order is implemented by output message place of the control subsystem. After the motivity subsystem has received the order, the system starts to supply energy. At the same time, the real time inspection subsystem inspects the motivity subsystem and feeds back the inspecting information to the control subsystem. Then the control subsystem can give the next decision-making with the received information. All these actions can be implemented by the input/output message places of each subsystem.

Fig.1 only gives the subsystems related to the motivity subsystem. In fact, the processes of message transferring are the actions implementing procedure. In Fig.1, all the subsystems are encapsulated into the modules. And the inner details of the modules are transparent to the outside. Communications of each subsystem can only be implemented by the input/output message places and the gate transition.

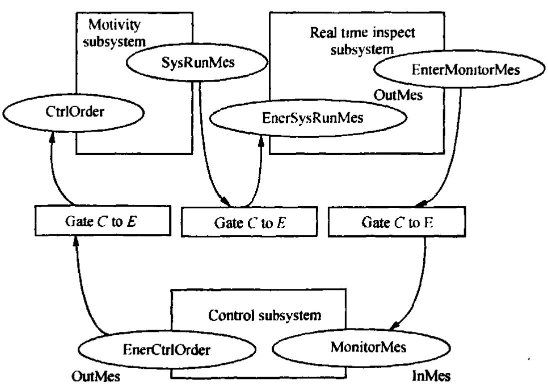

Fig.1 System layer model of AUV

3.2 The modeling of subsystem layers

Fig.2 is the motivity subsystem model. In the model, there are several outer message places attached with the motivity subsystem. There are input message place 'CtrlOrder' from the control subsystems and output message place 'SysRunMes' to the real-time inspection subsystem. In Fig.2 the transition EnergySupply in the rectangle is a substitution transition representing the energy supplying.

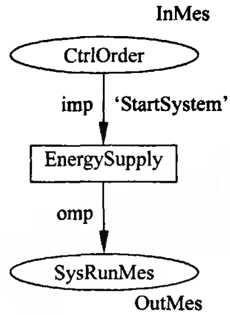

Fig.2 COOPN model of energy system

Fig.3 shows the Color set and variable definition. After detailing the substitution transition the 'Energy supplying of Energy System' subnet model can be obtained. See Fig.4. The token was defined according to the CPN ML definition.

```
Val j = 2;
Color Num = int with 1 ···2;
Color Elecap = with plentitude | shortage;
Color Validity = with uptoDate | out-of-Date;
Color BattWeight = with underMax | hyperMax;
Color Supply = record cap:Elecat *vali:Validity*
                      weight:BattWeight;
Color BattSupply = product Num * Supply;
Color FaultMes = with BattIrreparable;
Color BattFaulyMes = product Num * FaultMes;
Color InMes = with StartSystem | StopSystem;
Color OutMes = with CompleteMes | BattFaultMes;
Var i:Num;
Var bs:Supply;
Var imp:InMes;
Var omp:OutMes;
```

Fig.3 Color set and variable definition

As the restriction of the technology at present, the energy supplier is always the battery pile. In this example, Batt1 and Batt2 are assumed as the energy supplier. Three attributes can be defined to the battery pile. They are respectively 'EleCap' representing the electricity capability, Validity and 'BattWeight' representing the battery pile's weight.

In Fig.4 the place 'BattAvailable' represents Batt1 and Batt2. It is free and in useable state. Its initial value can be represented with $1'(1,bs)++1'(2,bs)$. When the message 'StartSystem' from input message place arrives, the system will start to supply energy. After the action of supplying energy, the input message place 'SysRunMes' will receive the energy supplying information from the system. All these can be modeled with the method talked above. The reachable tree can be got from the model. Thus the process of the energy supplying can be analyzed.

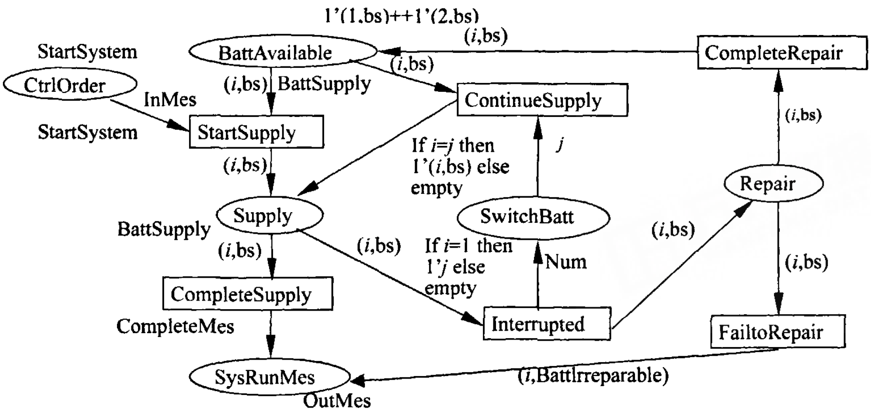

Fig.4 Energy supplying of energy subsystem

3.3 Analyzing the energy supply subsystem

The initial state of system can be represented by I_0. $I_0 = \{\text{StartSystem}, (1'(1,\text{bs}) + 1'(2,\text{bs}), 0,0,0,0)\}$. By analyzing the process of energy supplying of the motivity subsystem, the state reachable tree can be obtained. Seven types of terminal node That means the system has seven terminal states:

I_1'=(0,1'(2,bs), 0,0,0,CompleteMes)

I_2'=(0,1'(1,bs), 0,0,0,CompleteMes)

I_3'=(0,0,0,0,0, (CompleteMes+(1,BattIrreparable)))

I_4'=(0, (1'(1,bs)+1'(2,bs)), 0,0,0,0)

I_5'=(0,1'(1,bs), 0,0,0, (2, BattIrreparable))

I_6'=(0,1'(2,bs), 0,0,0, (1, BattIrreparable))

I_7'=(0,0,0,0,0, (1,BattIrreparable)+(2,BattIrreparable)

From analyzing these terminal nodes, the function implementation of the system can be learned. Consequently the errors in the structure design and running process of the system can be found. The state analyzing figure based on the reachable tree can be obtained. From Fig.4 it can be learned that if the battery pile can be repaired, the problem will occur in the overweight of the battery pile (hyperMax). So the system can resolve the problem through abandoning the load. If the battery pile can't be repaired, the problem may be out of date or shortage of energy of the battery pile. Thus the AUV system can't finish the repair with the control subsystem. Therefore, before AUV carrying out the tasks, the battery piles should be checked up carefully, otherwise the control subsystem should improve decision-making to solve the problem.

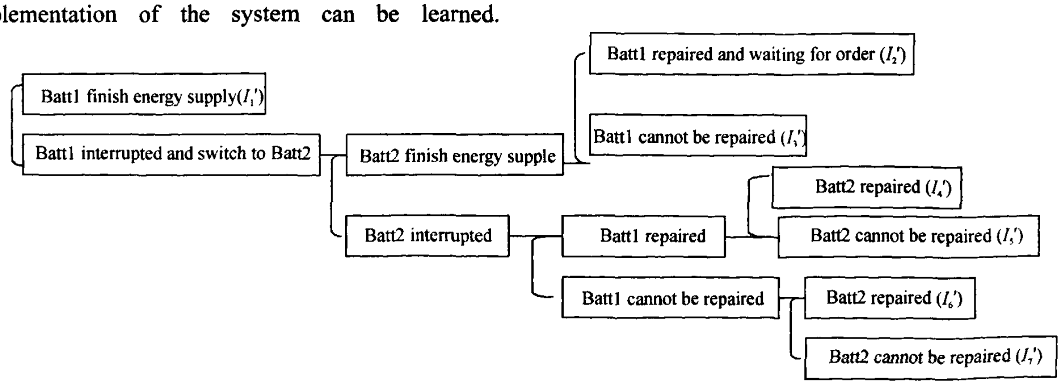

Fig.5 Energy supply state analyzing

5 Conclusions

This paper has informally introduced OOPN and has identified the key extensions over CPN. The most distinctive feature of COOPN is the unified class hierarchy which allows both tokens and subnets to be defined by classes and arbitrarily intermixed. This provides direct support for models with multiple levels of activity, and gives the modeler great flexibility in the choice of modeling components as active or passive. COOPN supports the abstraction of both places and transitions, leading to the notions of super places and super transitions. While the emphasis of this paper is informal, some of the theoretical issues pertinent to COOPN have been discussed, particularly the implications reachability analysis is feasible.

According to the structure, function implementation and special running environment of the AUV system, this paper mainly gives a modeling method. The method based on the COOPN gives a solution of establishing the model of the tasks implementation of the AUV system. The nodes in the model are from the theory of the Colored Petri Net. The attributes of the node can also be clearly understood. The reachable tree can be also established from the model. The information of the net system running state can be learned from analyzing the reachable tree. These information correspond to the function implementation of the AUV system. Thus the modeling method can also provide the foundation of the AUV system performance evaluation.

References

[1] LI Hao, KANG Fengju, Research on simulation modeling and validation of underwater self-running vehicle[J]. Ship Engineering, 2001(3): 55-57(in Chinese).

[2] LAKOS C. The object orientation of object Petri nets[C]// Proceedings of the First International Workshop on Object-oriented Programming and Models of Concurrency. Torino, Italy, 1995.

[3] HONG J E, BAE D H. Software modeling and analysis using a hierarchical object-oriented Petri Net[J]. Information Sciences, 2000, 130: 133-164.

[4] WANG L C. Object-oriented Petri Nets for modeling and analysis of automated manufacturing systems[J]. Computer Integrated Manufacturing Systems, 1996, 26(2): 111-125.

[5] BUCHS D, GUELFI N. Open distributed programming using the object oriented specification formalism CO-OPN[C]// 5th International Conference on: Putting into Practice Methods and Tools for Information System Design. Nantes, 1992: 261-279.

[6] CESCA M, JANOUSEK V, VOJNAR T. PNtalk-an experimental system based on object-oriented Petri Nets[EB/OL]. 2007-10-22. http://www.fee.vutbr.cz/UIVT/homes/vojnar/Publications/vojnar-icatpn98_tools.ps.gz.

[7] LAKOS C. A Petri Net View of Mobility[C]// Formal Techniques for Networked and Distributed Systems FORTE 2005. Taipei, 2005.

[8] KRISHNA M K, ALIREZA M. Modeling multithreaded applications using Petri Nets[J]. International Journal of Parallel Programming, 2002,30 (5): 353-371.

FENG Xiao-ning was born in 1976. He is a lecturer at Harbin Engineering University. His achievement mainly focuses on computer science. His current research interests include Ontology, Petri Net modeling method, and semantic information, etc.

Applying a semantic information Petri Net modeling method to AUV systems design

作者： FENG Xiao-ning， WANG Shuo， WANG Zhuo， LIU Qun
作者单位： FENG Xiao-ning,WANG Zhuo,LIU Qun(Department of Computer Science & Technology,Harbin Engineering University,Harbin 150001,China)， WANG Shuo(Chu Kochen Honors College,Zhejiang University,Hangzhou 310058,China)
刊名： 船舶与海洋工程学报（英文版）
英文刊名： JOURNAL OF MARINE SCIENCE AND APPLICATION
年，卷(期)： 2008，7(4)

参考文献(8条)

1. LI Hao.KANG Fengju Research on simulation modeling and validation of underwater self-running vehicle[期刊论文]-Ship Engineering 2001(03)

2. LAKOS C The object orientation of object Petri nets 1995

3. HONG J E.BAE D H Software modeling and analysis using a hierarchical object-oriented Petri Net 2000

4. WANG L C Object-oriented Petri Nets for modeling and analysis of automated manufacturing systems 1996(02)

5. BUCHS D.GUELFI N Open distributed programming using the object oriented specification formalism CO-OPN 1992

6. CESCA M.JANOUSEK V.VOJNAR T PNtalk-an experimental system based on object-oriented Petri Nets 2007

7. LAKOS C A Petri Net View of Mobility 2005

8. KRISHNA M K.ALIREZA M Modeling multithreaded applications using Petri Nets 2002(05)