Research on modeling of equipment supporting command automation system based on colored petri net

Qilin Zhu\(^1\) and Ye Sun\(^1\)

\(^1\)Department of Vehicle Application, Army Academy of Armored Forces, Economic and Technological Development Zone, Changchun, Jilin Province, China

Corresponding author and e-mail: Ye Sun, 201705024218jcy@ncist.edu.cn

Abstract. UML (Unified Modeling Language) has become the de facto standard for object-oriented analysis and design modeling. The design result of the UML-based equipment support command automation system model is a static representation of the system, while the nature of the equipment support command automation system is dynamic. Colored Petri Net (Colored Petri Net, CPN) has a strong description ability, a rigorous mathematical foundation and a variety of analysis methods, and it can be run in simulation. A method for mapping from UML products to CPN (Colored Petri Nets) executable models that can be used for logic, behavior, and performance evaluation of the architecture is proposed. First, the general description of UML modeling of the equipment support command automation system is carried out. Then it elaborated the process of the equipment support command automation system mapping from UML products to CPN to build executable models.

1. Introduction

In the development of equipment support command automation system [1], modeling first can enable system developers and maintenance personnel to enhance their understanding of the system, ensure the consistency of system analysis, system design and system maintenance, and more importantly, reduce system development risks, and shorten the development cycle. UML has become the de facto standard for object-oriented analysis and design modeling. The design result of the UML-based equipment support command automation system model is a static representation of the system, but in fact the nature of the equipment support command automation system is dynamic. CPN has a strong description ability, a rigorous mathematical foundation and a variety of analysis methods, and it can be run in simulation [2]. This paper studies the process of the equipment support command automation system from the static modeling based on UML for the establishment of the dynamic executable model CPN.

2. Mapping method from UML static model to dynamic CPN

When constructing CPN, it is necessary to create a clear mapping from UML elements to CPN elements [3], which includes structural elements such as position, transition, input and output arcs, as well as color sets and variables in the global declaration node, and finally need to determine the initialization flag of the CPN. Because the class diagram is a general description of the object-oriented design of the architecture, the class diagram and activity diagram are chosen to map the CPN executable model. The specific algorithm is as follows:
Before converting the executable model, it is necessary to correct the consistency errors of UML products and maintain a unified data dictionary to support products such as class diagrams and behavior diagrams containing rules.

(2) Transform the class diagram into a suitable form:
   a. Convert all the association relationships in the class diagram into association classes. Define attributes for the association relationship in the class diagram according to the message defined in the activity diagram, and convert it into an association class;
   b. Transform all the classes in the class diagram into a form with only attributes or only operations through the aggregation relationship. That is, a class that has both attributes and operations is transformed into a parent class that contains only operations and one or more subclasses that contain properties and have an aggregation relationship with the parent class through the aggregation relationship.

(3) Determine the structure of the CPN
   a. Use the attributes of all classes in the class diagram to form a global declaration node;
   b. Construct a hierarchical CPN;
      • Create a transition for each interactive class in the class diagram;
      • Create a position and assign an appropriate color set for each associated or aggregated class;
      • Use the activity diagram to create an arc between the transition and the position (note that the number of associations in the class diagram and the number of positions between the executable model transitions should be one-to-one correspondence);
      • Create a subpage for each transfer; create a transition in a subpage for each operation in the transition; assign input, output, and I/O port positions; create arcs based on the activity diagram; add arc titles and monitoring functions;
      • Specify the initial mark for each position that represents the aggregation class.

3. Equipment support command automation system based on UML modeling
The UML-based equipment support command automation system modeling practice process follows the core ideas and basic principles of the Rational Unified Process (RUP), that is, iterative object-oriented analysis driven by Use Case (use case) and system architecture as the core. And the design process [4].

The starting point of the development process [5] is to fully understand the business process of the entire equipment support command automation system, obtain a set of business domain vocabulary, form a combat concept, form a preliminary system class diagram, then develop use cases and use interactive dynamic diagrams to describe the execution steps of use case scenarios Sequence, fill in the properties and methods of the preliminary class diagram created to complete the class diagram. Use case diagrams, interactive dynamic diagrams and class diagrams constitute the logical model of the equipment support command automation system, and then use component diagrams and configuration diagrams to complete the physical design. Since the mapping from UML static products to executable CPN dynamic products requires only class diagrams and activity diagrams, the requirements collection, development of use cases, interactive dynamic diagrams, component diagrams and configuration diagrams are omitted, and only activity diagrams and class diagrams are listed.

The simplified diagrams of command and control, equipment management, technical support, equipment support and ammunition support systems are as follows:
4. Establishment of executable model of equipment support command automation system

Only take the command and control system (Command Control) and technical support system (Technique Support) class diagrams as examples to build the executable model CPN according to the above mapping method.

First transform the class diagram, convert the association relationship between the Command Control and Technique Support classes into the associated classes Command Information and Technique Information, and transform the Technique Support class containing attributes and operations into a collection of mission systems (Mission) and capability systems (Capability). The fragment of the class diagram that contains only the command and control, mission and capability classes is as follows:

![Class Diagram Fragment](image)

The global declaration node of the CPN model is derived from the class diagram and the data dictionary, which is composed of the attributes of all classes. These nodes constitute the color set and variable set as follows:

**Global Declaration Node**
- color group=double; var g:group;
- color team=double; var t:team;
- color maternal Mission=int; var mm: maternal Mission;
Constructing CPN. As shown in Figure 3, the interactive classes Command Control and Technique Support are transformed into transitions, and the associated classes and aggregate classes become positions. The top-level CPN diagram is as follows:

![Top-level CPN diagram](image)

**Figure 3.** Top-level CPN diagram.

Construct CPN sub page. Figure 4 only constructs the top-level CPN diagram, and also performs the next-level CPN for each operation in the class diagram corresponding to the top-level transfer. For example, the transfer of Technique Support corresponds to an analyze operation in the Technique Support class, and each operation has a lower-level view. Taking the analyze operation as an example, the lower-level transfer diagram can be made as follows:

![Lower-level CPN diagram](image)

**Figure 4.** The analysis operation transfer subpage fragment of Analyse transfer.

List the rules, the abbreviated rule model is as follows:
if (Commandlnformation=Technical Assurance) and (group and team are known) then (maternalMission subMission is tracked by group and team)
Repeating the above process can map all the elements in the UML diagram into CPN elements, thereby constructing an executable CPN model.

5. Analysis of dynamic executable model

The CPN can be executed as follows [6]:

\[
\Sigma = \left( P, T; F, K, W, M_0 \right)
\]

as a network system, \( M \) is an identifier on the base network (\( P \) is the set of positions, \( T \) is the set of transfers, \( F \) is the sequence pair from \( P \) to \( T \), \( K, W, \) and \( M_0 \) are the capacity function, weight function and initial state, respectively Logo).

1. The \( t \in T, t \cup t' \) is called the extension of \( t \); \( t' \) called the former set, \( t' \) called the latter set.

2. The condition that \( t \) has the right of occurrence under \( M \) is: for all \( p \in P, \{ p \in t \Rightarrow M(p) \geq W(p, t) \} \wedge \{ p \in t' \Rightarrow M(p) + W(t, p) \leq K(p) \} \). At this time, it is also said that \( M \) authorizes \( t \) to occur (execute). Denoted as \( M[t>t]. \)

3. If \( t \) has the right of occurrence under \( M \), then \( t \) can be executed, and the result of occurrence is that \( M \) becomes the new identifier \( M' \) defined as follows for all \( p \in P \):

\[
M'(p) = \begin{cases} 
M(p) - W(p, t) & \text{if } p \notin t - t' \\
M(p) + W(p, t) & \text{if } p \in t' - t \\
M(p) - W(p, t) & \text{if } p \notin t \cap t' \\
M(p) & \text{if } p \notin t \cup t'
\end{cases}
\]

The occurrence of \( t \) causes the label to become \( M' \) to \( M[t>t'M]. \) The successor mark of \( M' \), \( t \) has the right to occur under \( M \), which is also called \( t \) satisfies the ignition condition under \( M \). As long as the ignition conditions are met, Petri can be dynamically executed. According to this definition, the built Petri net model can be analyzed and debugged.

6. Conclusions

UML, which has become a de facto standard and conforms to people's thinking habits, is used for system modeling. The development cycle is relatively short, the products are rich and the program can be directly programmed, but the developed model is static, lacks dynamic mathematical analysis, and must be programmed. It is necessary to connect to the simulation operation on the spot or with other programs to judge the pros and cons of the built model. Colored Petri nets have powerful description capabilities, rigorous mathematical foundations and various analysis methods, and it can be directly simulated, which is of great significance for evaluating the function and performance of the described system, but its modeling process is relatively cumbersome and difficult direct programming. The equipment support command automation system proposed in this paper first uses UML to model, and then converts the colored Petri net, which enables the two methods to complement each other. This method will also have good application prospects in system analysis and modeling in many other fields.

References

[1] Song Huawen Li Fusheng Comprehensive drill for equipment support [M] Beijing: People's Liberation Army Press 2003-06. 87-91.
[2] Jensen K. Colored Petri Nets, Basic Concept, Analysis Method and Practical Use [M]. Springer-Verlag, 2017(1).
[3] Lee W. Wagenhals, Sajjad Haider, and Alexander H. Levis Synthesizing Executable Models of Object Oriented Architectures[J]. Research and Practice in Information Technology, Vol 12. 2012.
[4] You Kebin, UML application modeling practice process [M]. Beijing: Machinery Industry Press, 2013.1 38-39.
[5] James Rumbaugh, Ivar Jacobson, Grady Booch. The Unified Modeling Language User Guide [M]. Addison-Wesley 2011 23-28.