Three Types of Connector-based component assembling techniques in the Model of EOOPN

Na Zhao^{2, #}, Jian Wang ^{1,*,#}, Fei Dai², Qi Mo², Lei Zhang ², Zhendi Yang ², Xue Sun²

¹College of Information Engineering and Automation, Kunming University of Science and Technology, China

²School of Software, Key Laboratory in Software Engineering of Yunnan Province, Yunnan University, Kunming, China

Abstract. Base on the Extended Object-Oriented Petri Net, and the component evolution processes that has been proposed in our previous research, this paper studies the component assembling technique on the basis of EOOPN subnet combination. The definitions of three types of connector-based component assembling approach, namely, parallel assembling, selective assembling and repetitive assembling, are given and subsequently their properties are detailed.

Keywords: parallel assembling, selective assembling, repetitive assembling, component net, component evolution

1. Introduction

The description and evolution of components are one of the key technologies to fully exert the technical advantages of components, to realize software reuse and to build high-quality software product. To better support the component-based software development, many researchers have modeled components by using formalized method, and develops a variety of studies such as component evolution, component performances analysis, and so on.

Petri Nets has frequently been used in the process of software modeling due to its strict mathematical definition and visual graphical expression[1,2]. In order to more effectively carry out the formalized descriptions and analysis of components, the Petri Nets is extended to EOOPN [3-9], the object-oriented Petri-net that supports component evolution, based on the related definitions and principles of components while taking the features of object-oriented modeling methods into consideration.

The Extended Object-Oriented Petri Nets is the combination of Petri Nets and object-oriented technology. By compensating the modeling ability limitations of Petri Nets, such combination is capable of representing various resources and constraints within a highly complicated system in a simplified way. At the same time, the features of Petri Nets make it competent in revealing the simulated system's structural and dynamic behavioral information, especially in the analysis of system's performances and concurrency, so as to improve the efficiency of the system.

In order to propose a set of methodologies that help with the description, the analyzing and the evolving of software components by the using of EOOPN model, we extended the modeling power of EOOPN by proposing EOOPN Component Net (EOOPN-CN), which serves as the building block of higher level component or application system. To better support component-based software evolution on a larger scale, we further proposed the EOOPN Component Net assembling techniques. In this paper, we will be detailing

^{1*}Corresponding author, E-mail address: obcs2002@163.com

[#]These authors contributed equally to this work.

three types of connector-based component assembling approach, namely, parallel assembling, selective assembling and repetitive assembling, by giving their definitions, and subsequently their properties.

2. The selective assembling of EOOPN component net

The idea that the assembling of multiple components is the extension of that between two components helps to narrow the discussion scope down to the assembling between only two components. There are four types of assembling approaches based on connectors: sequential assembling, parallel assembling, selective assembling and repetitive assembling. The selective assembling, parallel assembling and repetitive assembling will be discussed under the general frame of EOOPN in this paper.

2.1. The selective assembling of EOOPN component net

Definition 1: $CN_1 = (P_1, T_1, F_1, S_1, A_{P_1}, A_{T_1}, A_{F_1})$, $CN_2 = (P_2, T_2, F_2, S_2, A_{P_2}, A_{T_2}, A_{F_2})$ are two Component net,

 $t_{in1}, t_{in2}, t_{out1}, t_{out2}$ are connectors and p_{in}, p_{out} are places, let $CN = (P, T, F, S, A_P, A_T, A_F)$, where:

1) $P = P_1 \cup P_2 \cup p_{in} \cup p_{out}$, where *CN.oip* is composed of p_{in} and p_{out} ;

2)
$$T = T_1 \bigcup T_2 \bigcup t_{in1} \bigcup t_{in2} \bigcup t_{out1} \bigcup t_{out2}$$

- 3) $F = F_1 \bigcup F_2 \bigcup (P_{in}, t_{in1}) \bigcup (P_{in}, t_{in2}) \bigcup (t_{in1}, P_1.oip) \bigcup (t_{in2}, P_2.oip) \bigcup (P_1.oip, t_{out1}) \\ \bigcup (P_2.oip, t_{out2}) \bigcup (t_{out1}, P_{out}) \bigcup (t_{out2}, P_{out})$;
- 4) $A_p = A_{p_1} \bigcup A_{p_2} \bigcup A_p(p_{in}) \bigcup A_p(p_{out})$, and $|A_p(p_{in})| = 1$;

5)
$$A_T = A_{T_1} \bigcup A_{T_2} \bigcup A_T(t_{in1}) \bigcup A_T(t_{out1}) \bigcup A_T(t_{in2}) \bigcup A_T(t_{out2});$$

6) $A_{F} = A_{F_{1}} \bigcup A_{F_{2}} \bigcup A_{F}(P_{in}, t_{in1}) \bigcup A_{F}(P_{in}, t_{in2}) \bigcup A_{F}(t_{in1}, P_{1}.oip) \bigcup A_{F}(t_{in2}, P_{2}.oip)$ $\bigcup A_{F}(P_{1}.oip, t_{out1}) \bigcup A_{F}(P_{2}.oip, t_{out2}) \bigcup A_{F}(t_{out1}, P_{out}) \bigcup A_{F}(t_{out2}, P_{out});$

$$\begin{aligned} 7) \quad & A_{F}(P_{in},t_{in1}) \in L_{m}(A_{P}(p_{in})), \ A_{F}(P_{in},t_{in2}) \in L_{m}(A_{P}(p_{in})), \ A_{T}(t_{in1}) = A_{F}(p_{in},t_{in1}), \ A_{T}(t_{in2}) = A_{F}(p_{in},t_{in2}), \\ & A_{F}(t_{in1},P_{1}.oip) \in L_{m}(A_{P1}(P_{1}.oip)), \ A_{F}(t_{in2},P_{2}.oip) \in L_{m}(A_{P2}(P_{2}.oip)), \ A_{F}(p_{1}.oip,t_{out1}) \in L_{m}(A_{P}(p_{1}.oip)), \\ & A_{T}(t_{out1}) = A_{F}(p_{1}.oip,t_{out1}), \ A_{F}(t_{out1},p_{out}) \in L_{m}(A_{P}(p_{out})), \ A_{F}(p_{2}.oip,t_{out2}) \in L_{m}(A_{P}(p_{2}.oip)), \\ & A_{T}(t_{out2}) = A_{F}(p_{2}.oip,t_{out2}), \ A_{F}(t_{out2},p_{out}) \in L_{m}(A_{P}(p_{out})). \end{aligned}$$

Then CN is called the selective assembling of CN_1 and CN_2 .

Theorem 1: Let $CN_1 = (P_1, T_1, F_1, S_1, A_{P_1}, A_{T_1}, A_{F_1})$ and $CN_2 = (P_2, T_2, F_2, S_2, A_{P_2}, A_{T_2}, A_{F_2})$ be two well-structured component nets and $t_{in1}, t_{in2}, t_{out1}, t_{out2}$ be connectors. Then the selective assembling component net $CN = (P, T, F, S, A_P, A_T, A_F)$ is also well-structured.



Figure1. the selective assembling of EOOPN component net

2.2. The parallel assembling of EOOPM component net

Definition 2: $CN_1 = (P_1, T_1, F_1, S_1, A_{P_1}, A_{T_1}, A_{F_1})$, $CN_2 = (P_2, T_2, F_2, S_2, A_{P_2}, A_{T_2}, A_{F_2})$ are two Component net, t_{in}, t_{out} are connectors and p_{in}, p_{out} are places, let $CN = (P, T, F, S, A_P, A_T, A_F)$, where:

1) $P = P_1 \cup P_2 \cup p_{in} \cup p_{out}$, where *CN.oip* is composed of p_{in} and p_{out} ;

2)
$$T = T_1 \cup T_2 \cup t_{in} \cup t_{out};$$

3)
$$F = F_1 \cup F_2 \cup (P_{in}, t_{in}) \cup (t_{in}, P_1.oip) \cup (t_{in}, P_2.oip) \cup (P_1.oip, t_{out});$$

$$\cup (P_2.oip, t_{out}) \cup (t_{out}, P_{out});$$

- 4) $A_{p} = A_{p_{1}} \bigcup A_{p_{2}} \bigcup A_{p}(p_{in}) \bigcup A_{p}(p_{out});$
- 5) $A_T = A_{T_1} \bigcup A_{T_2} \bigcup A_T(t_{in}) \bigcup A_T(t_{out});$

6)
$$A_{F} = A_{F_{1}} \bigcup A_{F_{2}} \bigcup A_{F}(P_{in}, t_{in}) \bigcup A_{F}(t_{in}, P_{1}.oip) \bigcup A_{F}(t_{in}, P_{2}.oip) \bigcup A_{F}(P_{1}.oip, t_{out}) \bigcup A_{F}(P_{2}.oip, t_{out}) \bigcup A_{F}(t_{out}, P_{out});$$

- 7) $A_F(P_{in}, t_{in}) \in L_m(A_P(p_{in})), A_T(t_{in}) = A_F(P_{in}, t_{in}), A_F(t_{in}, P_1.oip) \in L_m(A_{P1}(P_1.oip)),$
- $A_{F}(t_{in}, P_{2}.oip) \in L_{m}(A_{P2}(P_{2}.oip)), A_{F}(P_{1}.oip, t_{out}) \in L_{m}(A_{P}(P_{1}.oip)), A_{F}(P_{2}.oip, t_{out}) \in L_{m}(A_{P}(P_{2}.oip)), A_{F}(P_{2}.oip) \in L_{m}(A_{P2}(P_{2}.oip)), A_{F}(P_{2}.oip)), A_{F}(P_{2}(P_{2}.oip$
- $A_{T}(t_{out}) = A_{F}(P_{1}.oip, t_{out}), \ A_{T}(t_{out}) = A_{F}(P_{2}.oip, t_{out}), \ A_{F}(t_{out}, p_{out}) \in L_{m}(A_{P}(p_{out})).$

Then CN is called the parallel assembling of CN_1 and CN_2 .

Theorem 2: Let $CN_1 = (P_1, T_1, F_1, S_1, A_{P_1}, A_{T_1}, A_{F_1})$ and $CN_2 = (P_2, T_2, F_2, S_2, A_{P_2}, A_{T_2}, A_{F_2})$ be two well-structured component nets and t_{in}, t_{out} be connectors. Then the parallel assembling component net $CN = (P, T, F, S, A_P, A_T, A_F)$ is also well-structured.



Figure2. the parallel assembling of EOOPN component net

Figure 3. the repetitive assembling of EOOPN component net

2.3. The repetitive assembling of EOOPN component net

Definition 3: $CN_1 = (P_1, T_1, F_1, S_1, A_{P_1}, A_{T_1}, A_{F_1})$ is a Component net, *t* is a connector, let $CN = (P, T, F, S, A_P, A_T, A_F)$, where:

- 1) $P = P_1$, and $CN.oip = CN_1.oip$;
- $2) T = T_1 \bigcup t ;$
- 3) $F = F_1 \bigcup (t, P_1.oip) \bigcup (P_1.oip, t);$

$$A_{P} = A_{P_{1}};$$

5)
$$A_T = A_{T_1} \bigcup A_T(t);$$

- 6) $A_F = A_{F_1} \bigcup A_F(t, P_1.oip) \bigcup A_F(P_1.oip, t);$
- 7) $A_F(P.oip,t) \in L_m(A_P(P.oip)), A_T(t) = A_F(P.oip,t), A_F(t, P.oip) \in L_m(A_P(P.oip)).$

Then CN is called the repetitive assembling of CN_1 .

Theorem 3: Let $CN_1 = (P_1, T_1, F_1, S_1, A_{P_1}, A_{F_1}, A_{F_1})$ be a well-structured component net and *t* be a connector. Then the repetitive assembling component net $CN = (P, T, F, S, A_P, A_T, A_F)$ is also well-structured.

3. Summary

Based on the EOOPN Component Net (EOOPN-CN) and their combination, three types of connectorbased component assembling approach, namely, parallel assembling, selective assembling and repetitive assembling were defined, analyzed and discussed under the general framework of EOOPN in this paper. It is through these types of component assembling techniques that EOOPN model obtains more capabilities of supporting component-based software evolution processes on a larger and broader scale.

4. Acknowledgment

This work has been supported by the National Science Foundation of China under Grant No. 61379032, 61364024, 61462091, 61462095, 61262024, 61462092, 61263017, by the Science Foundation of Yunnan Province, China under Grant No. 2015FA014, 2013FB008, 2014FD006, 2012FD005, by the Science Foundation of Yunnan Province Education Department of China under Grant No. 2015Y013, by the Science Foundation of Key Laboratory in Software Engineering of Yunnan Province under Grant No. 2012SE308, 2012SE401, 2012SE307, by the eighteenth batch of Yunnan Province in the young academic and technical leaders reserve personnel training project under Grant No. C6143002, by the Science and Technology plan of Applied Basic Research Programs Foundation of Yunnan province No.2014FB112 and the Training Program for the Key Young Teachers of Yunnan University.

5. References

- [1] Stanislav Chachkov, Didier Buchs, From Formal Specifications to Ready-to-Use Software Components: The Concurrent Object Oriented Petri Net Approach. In proceeding of ICACSD2001, 2001.
- [2] Jang-Eui Hong, Doo-Hwan Bae, Software modeling and analysis using a hierarchical object-oriented Petri net.Information Sciences.130 (2000) 133-164.
- [3] Na Zhao, The Description and Evolution of Components on the Basis of EOOPN[D]. PhD thesis, YunNan University, June 2011.
- [4] Na Zhao, Fei Dai, Yong Yu, Tong Li. An Extended Process Model Supporting Software Evolution[C]. In Proceedings of 2008 International Symposium on Intelligent Information Technology Application (IITA 2008). IEEE computer society. December 2008, Shanghai, China, 1013~1016.
- [5] Na Zhao, Tong Li, Ling ling Yang, Yong Yu, Fei Dai, Wei Zhang. The Resource Optimization of Software Evolution Processes[C]. In Proceedings of 2009 International Conference on Advanced Computer Control (ICACC 2009). January 2009, Singapore, 332~336.
- [6] Na Zhao, Jian Wang, Wei Du, Yunchun Zhang, Zuo Jiang, Jinzhuo Liu, Ye Qian, Using Fuzzy Theories to Evaluate the Component Reusability in the Process of Software Evolution, In Proceedings of 2012 International Conference on Uncertainty Reasoning and Knowledge Engineering, August 2012, Jakarta, 212-215.
- [7] Na Zhao, Jian Wang, Yong Yu, Fei Dai, Zhongwen Xie, Jianglong Qin, An Evaluation Algorithm of Component Retrieval Based on Fuzzy Theories, In Proceedings of 2012 International Conference on Uncertainty Reasoning and Knowledge Engineering, August 2012, Jakarta, 216-219.
- [8] Na Zhao, Jian Wang, Tong Li, Yong Yu, Fei Dai and Zhongwen Xie. ESDDM: a Software Evolution Process Model based on Evolution Behavior Interface[C]. Communications in Computer and Information Science, Vol 135, pp. 562–567. 2011. Springer-Verlag Berlin Heidelberg.
- [9] Zhao Na, Zhao Yang and Li Tong. A Method of Modelling and Performance Analysis for Concurrent Development Process of Software[C]. In Proceedings of the 11th Joint International Computer Conference. World Scientific: New Jersey, 2005, 803-809.