The Sequential Assembling of EOOPN Component Net

Na ZHAO\textsuperscript{2,a}, Jian WANG\textsuperscript{1,b*}, Ying LIN\textsuperscript{2}, Qian YU\textsuperscript{2}, Zhen-Di YANG\textsuperscript{2}, Xue SUN\textsuperscript{2}, Lei ZHANG\textsuperscript{2}

\textsuperscript{1}College of Information Engineering and Automation, Kunming University of Science and Technology, China
\textsuperscript{2}School of Software, Key Laboratory in Software Engineering of Yunnan Province, Yunnan University, Kunming, China

\textsuperscript{a}zhaonayx@126.com, \textsuperscript{b}obcs2002@163.com

\textsuperscript{*}Corresponding author

Keywords: Sequential Assembling, Component Net, Base Net, EOOPN, Component Evolution.

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 based on the EOOPN subnet combination. The definition of sequential assembling, a type of connector-based EOOPN-CN assembling approach, is given and subsequently its properties and proof is detailed. This paper’s work laid foundations for the further presentational and analytical aspects of component evolution.

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 developed 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\cite{1,2}. In order to more effectively carry out the formalized descriptions and analysis of components, the Petri Nets is extended to EOOPN \cite{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 three types of connector-based component assembling approach, namely,
parallel assembling, selective assembling and repetitive assembling, by giving their definitions, and subsequently their properties.

**Base Net of EOOPN[3-9]**

**Definition 1**

The base net of EOOPN is a 7-tuple $N = (P,T,F,S,AP,AT,AF)$, where:

1. $P$ is a finite set of places where $oip$ (object identify place) is a special place which can be defined as a 3-tuple $oip = (id,p,ip)$, where
   - $id$ is the unique variable name of the class in EOOPN;
   - $p$ is the interface place for sending and receiving messages of objects;
   - $ip$ is the variable name of the instance of a class;
2. $T$ is a finite set of transitions, $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$;
3. $F \subseteq (P \times T) \cup (T \times P)$ a set of directed arcs;
4. $S$ is a finite non-empty set, which is called as the colored set or type set;
5. $AP : P \rightarrow (PLS)\cup (PLS)\cup (PLS)$ is the color function of place. $PLS$ is a linear function with non-negative integer coefficients, $L(S)$ is the $PLS$ with coefficients are not all zero;
   - $PLS$ represents the type and number of a color that $p$ receives;
6. $AT : T \rightarrow L_r(S)$, $AT$ is a guard function of $T$, where $L_r(S)$, meets the condition: $\forall t \in T, AT(t) fire t$;
7. $AF : F \rightarrow L_r(S)$, $AF$ is the function of arc. Let $\forall f \in F, AF(f)$ represents the combination of colors passed through arc $f$;

**The Component Connector of EOOPN**

**Definition 2**

The component connector of EOOPN is a 7-tuple $LN = (P,T,F,S,AP,AT,AF)$, where:

1. $P$ is a finite set of places. $P$ represented the inner state of connector $LN$ while $P,oip$ denotes the input and output places are excluded from $P$;
2. $T$ is a finite set of transitions, $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$; $T$ represents the various operations in connector $LN$ where: $t_i$ and $t_o$ ($t_i, t_o \in T$), respectively, stands for message receiving and sending transitions .
3. $F \subseteq (P \times T) \cup (T \times P)$ is a finite set of arcs which describing the restraint relationship between places and transitions in a $LN$;
4. $S$ is a finite non-empty set, used to define the data types transmitted by $LN$;
5. $AP : P \rightarrow L_r(S)$, $AP$ is the color function of place. $L_r(S)$ is a linear function with non-negative integer coefficients and $L_r(S)$ is the $L(S)$ that the coefficients are not all zero; $\forall p \in P, AP(p)$ represents the type and number of a color that $p$ receives;
6. $AT : T \rightarrow L_r(S)$, $AT$ is a guard function of $T$, where $L_r(S)$, meets the condition: $\forall t \in T, AT(t) fire t$;
7. $AF : F \rightarrow L_r(S)$, $AF$ is the function of arc. Let $\forall f \in F, AF(f)$ represents the combination of colors passed through arc $f$;
The Component Properties of EOOPN

Definition 3

In component net $CN = (P, T, F, S, A_P, A_T, A_F)$, for $\forall p \in P$ and $\forall t \in T$, if $(p, t) \in F$, have $A_F(p, t) \in L_m(A_F(p))$, or $\forall p \in P, \forall t \in T$; if $(t, p) \in F$, have $A_F(t, p) \in L_m(A_F(p))$, then the place $p$ and transition $t$ in component $C$ is called typed-matched.

Definition 4

For any interface of a component, denoted as $C_{i.oip}$, and input and output transitions of connector $L_N$ where $L_{i.oip} \subseteq L_N$, if $A_F(C_{i.oip}, L_{i.oip}) \in L_m(A_F(C_{i.oip}))$, $A_F(C_{i.oip}, L_{i.oip}) \in L_m(A_F(C_{i.oip}))$, then $C_{i.oip}$ and $L_N$ is called type-matched.

Definition 5

In component net $CN = (P, T, F, S, A_P, A_T, A_F)$, component $C$ is well-structured if the following conditions are met:

1) $|CN.oip| = 1$;
2) for $\forall p \in P$ and $\forall t \in T$, if $(p, t) \in F$ or $(t, p) \in F$, then the place $p$ and transition $t$ in component $C$ is type-matched;
3) for $\forall t \in T$ and $\forall p_i \in \{i = 1, 2, \ldots, n\}$, have $\sum_{i=1}^{n} A_F(p_i, t) = A_F(t)$.

The Sequential 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 sequential assembling will be discussed under the general frame of EOOPN in this paper.

Definition 6

$CN_i = (P_i, T_i, F_i, S_i, A_{P_i}, A_{T_i}), CN_2 = (P_2, T_2, F_2, S_2, A_{P_2}, A_{T_2})$ are two Component net, and connector $t$, let $CN = (P, T, F, S, A_P, A_T, A_F)$, where:

1) $P = P_1 \cup P_2$, where $CN.oip$ is composed of $CN_{1.oip}$ and $CN_{2.oip}$;
2) $T = T_1 \cup T_2 \cup t$;
3) $F = F_1 \cup F_2 \cup (P_{1.oip}, t) \cup (t, P_{2.oip})$;
4) $A_F = A_{F_1} \cup A_{F_2}$;
5) $A_F = A_{P_1} \cup A_{T_1} \cup A_{P_2}$;
6) $A_F = A_{F_1} \cup A_{F_2} \cup A_{P_2}(P_{1.oip}, t) \cup A_F(t, P_{2.oip})$;
7) $A_F(P_{1.oip}, t) \in L_m(A_F(P_{1.oip})), A_F(t, P_{2.oip}) \in L_m(A_F(P_{2.oip})), A_F(t) = A_F(P_{1.oip}, t)$;

Then $CN$ is called the sequential assembling of $CN_i$ and $CN_2$.

Figure 1. The Sequential Assembling of EOOPN Component Net.
Theorem 1

Let \( CN_1 = (P_1, T_1, F_1, S_1, A_1, A_1, A_1) \) and \( CN_2 = (P_2, T_2, F_2, S_2, A_2, A_2, A_2) \) be two well-structured component nets and \( t \) be a connector. Then the sequential assembling component net \( CN = (P, T, F, S, A_p, A_T, A_F) \) is also well-structured.

Proof

According to the well-structured properties of component net \( CN_1, \ CN_2 \), it’s given that \( |CN_1,oip| = |CN_2,oip| = 1 \). Since \( CN_1,oip \) is composed by \( CN_1,oip \) and \( CN_2,oip \), thus component net \( CN \) also have single entry and single exit property, meaning \( |CN,oip| = 1 \). So \( CN \) meets the well-structured property (1).

Since \( CN_1 = (P_1, T_1, F_1, S_1, A_1, A_1, A_1) \), \( CN_2 = (P_2, T_2, F_2, S_2, A_2, A_2, A_2) \) are two well-structured component net, according to well-structured component property (2), for \( \forall p \in P_1 \cup P_2 \) and \( \forall t \in T_1 \cup T_2 \), if \((p', t') \in F_1 \cup F_2 \) or \((t', p) \in F_1 \cup F_2 \), then place \( p \) and transition \( t \) is type-matched. According to sequential assembling definition, the properties of \( A_i(P,oip,t) \in L_o(A_i(P,oip)) \) and \( A_i(t,P,oip) \in L_o(A_i(P,oip)) \) ensured the fact that \( t \) and \( P_oip \) is type-matched. Therefore, for \( \forall p \in P \) and \( \forall t \in T \), if \((t,p) \in F \) or \((p,t) \in F \), then the \( p \) and \( t \) in \( CN \) is type matched. This proves \( CN \) satisfies property (2) of well-structured component.

Since \( CN_1 = (P_1, T_1, F_1, S_1, A_1, A_1, A_1) \) and \( CN_2 = (P_2, T_2, F_2, S_2, A_2, A_2, A_2) \) are two well-structured component net, According to property (3), for \( \forall t' \in T_1 \cup T_2 \) and \( \forall p_i \in i \in 1, 2, \ldots, n \), one will have \( \sum_{i=1}^{n} A_i(p_i, t) = A_i(t) \). According to sequential assembling definition, the properties of \( A_i(t) = A_i(P_i,oip, t) \) ensured that, for \( \forall t \in T \) and \( \forall p_i \in i \in 1, 2, \ldots, n \), one will have \( \sum_{i=1}^{n} A_i(p_i, t) = A_i(t) \). This proves \( CN \) satisfies property (3) of well-structured component.

Summary

Based on the Extended Object-Oriented Petri Net, and the component evolution processes that has been proposed in our previous research, the sequential assembling technique of component net was discussed under the general framework of EOOPN in this paper. It is through component assembling that EOOPN is more capable of supporting component-based software evolution on a larger scale. The formal definition and detailed analysis on some other types of connector-based assembling approaches, like parallel assembling, selective assembling and repetitive assembling, to name a few, needs to be studied as the focus of next step.

Acknowledgement

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.
References


