Decomposition of Connected Trusted Component Based on Or-transition Colored Petri Net

Yong YU¹, Yong-jun LUO¹, Jian WANG², Yu-ru MA¹, Qiang LIU¹ and Na ZHAO¹*

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

*Corresponding author

Keywords: Trusted component, Component net, Petri net, Component decomposition.

Abstract. Trusted software is one of the most challenging and valuable research field in software engineering domain, among which are the wide ranging developments of component-based software engineering. On the basis of authors’ previous work - the description and modeling of trusted component based on Or-transition Petri net, the architectural evolution of trusted software is analyzed and the decomposition algorithm of connected trusted component based on Or-transition Colored Petri Net is proposed in this paper.

Introduction

Driven by a wide array of application requirements, the scale and the complexity of modern software are ever escalating at an alarming speed. The software credibility has becoming one of the most important properties to consider when evaluating a software system due to the sheer complexity of applications in the industries of finance, national defense, government and telecommunication.

As early as 1991, Laprie proposed the concept of software dependability from the angle of system security[1]. In 1997, the National Science and Technology Council of the United States explicitly proposed the concept of high credibility in document “the research challenge of high credibility system”[2]. Chinese scholars, Chen Huowang and Wang Ji, believe that high credibility software should provide a series of key properties[3] such as security, real-time, reliability, robust, and confidentiality when providing system services. As one of the most challenging and valuable research topics in the field of software, trusted software has attracted the attention of scholars at home and abroad. In modern software engineering technology, system componentize has become one of the major trends[4] and component based software development technology has been widely developed[5].

According to the relevant definition and principles of trusted software architecture and component, we extended Petri net to OR-transition colored Petri net, thus to effectively use the transitions in OR-transition colored Petri net to describe the operations in trusted component. In author’s previous work[6], the evolution of trusted software architecture on the basis of the characteristics of trusted component is analyzed, and the cohesion-based trusted component decomposition algorithm is proposed. This paper proposes decomposition algorithm of connected trusted component based on OR-transition Petri Net in our previous works.

Relative Works

At present, as a hot research area in the field of computer software, the research of trusted software is in the ascendancy. Governments, academics and industry experts and scholars from all over the world have conducted a comprehensive and multi-level study.

Taking software reliability attributes as the center, document[7] reviewed the classification framework and software architecture in many aspects, such as reliability modeling, reliability
growth model, reliability process simulation, reliability evaluation, and PCM-based (Palladio Component Model) reliability technology. With respect to the credibility during software execution (dynamic credibility), literature[8] proposed a credibility evaluation model based on the grading attributes of checkpoints. Based on the behavioral profile obtained during the software execution, a reliability evaluation model featured in behavior analysis is proposed by literature[9]. Literature[10,11,12,13] launches research work around the theme of component evolution. Literature[14] integrates the theory and method of credibility on the basis of ICEMDA architecture, namely a model driven construction method for trusted software, trying to ensure the credibility of management software throughout the entire evolutionary cycle. Literature[15] analyzes the problems of component discovery, selection and classification in network environment. According to the literature, a comprehensive reliability evaluation and processing system needs to be studied and designed concerning different influence factors, including network node delay, bandwidth and processing capability, thus providing faster speed and more comprehensive functions. Based on the trust chain model in area of trusted computing, a reliable intelligent entity model of internetwork is proposed in literature[16].

**Component Connectivity and Its Relevant Definition**

We first provide the component connectivity and its relevant definition.

**Definition 1:** For trusted component $C_t = \langle p_t, t_t, f_t, s_t, A_{P_t}, A_{T_t}, A_{F_t}, IP_t, OP_t \rangle$, for random $x, y \in P_t \cup T_t$, $x(F_R \cup F_R^{-1})^y$ is always true, $x(F_R \cup F_R^{-1})^y$ is the converse to $F_R$, namely $yF_R^{-1}x$ exists for $xF_R$.

**Definition 2:** For the sub trusted component $C'_t=\langle p'_t, t'_t, f'_t, s'_t, A_{P'_t}, A_{T'_t}, A_{F'_t}, IP'_t, OP'_t \rangle$ of trusted component $C_t = \langle p_t, t_t, f_t, s_t, A_{P_t}, A_{T_t}, A_{F_t}, IP_t, OP_t \rangle$, if $C'_t$ is connected, then $C'_t$ is called the connected sub trusted component of $C_t$.

**Definition 3:** Sub trusted component $C'_t=\langle p'_t, t'_t, f'_t, s'_t, A_{P'_t}, A_{T'_t}, A_{F'_t}, IP'_t, OP'_t \rangle$ is the connected sub connected trusted component of trusted component $C_t = \langle p_t, t_t, f_t, s_t, A_{P_t}, A_{T_t}, A_{F_t}, IP_t, OP_t \rangle$. If after adding a random arc which is in $C_t$ but not in $C'_t$ and its relevant status and operations in $C_t$, $C'_t$ is unconnected, then $C'_t$ is called the maximum connected sub trusted component of $C_t$; and the amount of all the maximum connected sub trusted components in $C_t$ is remarked as $\delta(C_t)$.

**Definition 4:** After deleting a random arc $\langle x, y \rangle \in F_t$ in trusted component $C_t = \langle p_t, t_t, f_t, s_t, A_{P_t}, A_{T_t}, A_{F_t}, IP_t, OP_t \rangle$, the trusted component changes into $C'_t=\langle p'_t, t'_t, f'_t, s'_t, A_{P'_t}, A_{T'_t}, A_{F'_t}, IP'_t, OP'_t \rangle$, where:

1. $P'_t = P_t \setminus \{p \}$, $T'_t = T_t \setminus \{t \}$, $F'_t = F_t \setminus \{(x, y)\}$, $S'_t = S_t$;
2. $\forall p \in P'_t$, then $A_{p'_t}(p) = A_{p_t}(p)$;
3. $\forall t \in T'_t$, then $A_{t'_t}(t) = A_{t_t}(t)$;
4. $\forall f \in F'_t$, then $A_{f'_t}(f) = A_{f_t}(f)$;
5. $IP'_t = IP_t; OP'_t = OP_t$.

**Definition 5:** In trusted component $C_t = \langle p_t, t_t, f_t, s_t, A_{P_t}, A_{T_t}, A_{F_t}, IP_t, OP_t \rangle$, the obtained trusted component after deleting an arc $f=\langle x, y \rangle \in F_t$ in trusted component $C'_t=\langle p'_t, t'_t, f'_t, s'_t, A_{P'_t}, A_{T'_t}, A_{F'_t}, IP'_t, OP'_t \rangle$. If $\delta(C'_t) = \delta(C_t) + 1$, then arc $f \in F_t$ is called the cutting arc of $C_t$.

**Definition 6:** In trusted component $C_t = \langle p_t, t_t, f_t, s_t, A_{P_t}, A_{T_t}, A_{F_t}, IP_t, OP_t \rangle$, the projection sub trusted component of $C_t$ based on set $N(N \subseteq P_t \cup T_t)$, then and only when:

1. $P'_t = P_t \cap N; T'_t = T_t \cap N; F'_t = (P'_t \times T'_t \cup T'_t \times P'_t) \cap F_t$;
2. $IP'_t = IP_t \cap N; OP'_t = OP_t \cap N$;
3. $S'_t = S_t \cap N$ (where $S_t$ is the sum of the corresponding types of all the elements in $N$);
4. $\forall p \in P'_t$, then $A_{p'_t}(p) = A_{p_t}(p)$; $\forall t \in T'_t$, then $A_{t'_t}(t) = A_{t_t}(t)$; $\forall f \in F'_t$, then $A_{f'_t}(f) = A_{f_t}(f)$.

the projection sub trusted component of trusted component $C'_t=\langle p'_t, t'_t, f'_t, s'_t, A_{P'_t}, A_{T'_t}, A_{F'_t}, IP'_t, OP'_t \rangle$ based on set $N(N \subseteq P_t \cup T_t)$ can be expressed as $C'_t \cap N$, namely $C'_t = C_t \cap N$. 

560
Decomposition of Connected Trusted Component in Trusted Software Architecture

For the convenience of the decomposition of connected trusted component in trusted software architecture, we can find a suitable cutting arc in the trusted component to be decomposed, and decompose trusted component on this base.

The algorithm for finding the cutting arc set in connected trusted component $C_t = \langle P_t, T_t, F_t, s_t, A_{P_t}, A_{T_t}, A_{F_t}, IP_t, OP_t \rangle$ is as algorithm 2 shows:

**Algorithm 2** The algorithm for finding the cutting arc set in trusted component $Arc-cut(C_t)$:

Input: connected trusted component $C_t = \langle P_t, T_t, F_t, s_t, A_{P_t}, A_{T_t}, A_{F_t}, IP_t, OP_t \rangle$;

Output: the cutting arc set $arc-cut-set(C_t)$ in trusted component $C_t$.

BEGIN

$arc-cut-set(C_t) = \emptyset$;

FOR every arc $f = \langle x, y \rangle \in F_t$, in trusted component $C_t$, DO

BEGIN

$F_t = F_t \setminus \{ \langle x, y \rangle \}$ /*delete arc $f$ from $C_t$, and get trusted component $C_t^*$*/

$N = \{ x \}$; $M = \{ y \}$;

FOR every element $r \in P_t \cup T_t$ DO

IF $r \in F_t \setminus x \setminus x \setminus r$ THEN $N = N \cup \{ r \}$;

END;

FOR every element $r \in P_t \cup T_t$ DO

IF $r \in F_t \setminus y \setminus r$ THEN $M = M \cup \{ r \}$;

END;

IF $N \cap M = \emptyset$ THEN $arc-cut-set(C_t) = arc-cut-set(C_t) \cup \{ \langle s_t, t \rangle \}$

END;

END.

For the cutting arc $\langle x, y \rangle \in F_t$, in connected trusted component $C_t = \langle P_t, T_t, F_t, s_t, A_{P_t}, A_{T_t}, A_{F_t}, IP_t, OP_t \rangle$, the non-connected trusted component $C_t' = \langle P'_{t_1}, T'_{t_1}, F'_t, s'_t, A_{P'_{t_1}}, A_{T'_{t_1}}, A_{F'_t}, IP'_{t_1}, OP'_{t_1} \rangle$ can be obtained after deleting cutting arc $\langle x, y \rangle \in F_t$ from trusted component $C_t$. Therefore, according to the decomposition method of non-connected trusted component, the dependency graphs corresponding to trusted component $C_t'$ can be grouped, and the specific method is as mentioned above. Assume that trusted component $C_t' = \langle P'_{t_1}, T'_{t_1}, F'_t, s'_t, A_{P'_{t_1}}, A_{T'_{t_1}}, A_{F'_t}, IP'_{t_1}, OP'_{t_1} \rangle$ and trusted component $C_t' = \langle P'_{t_2}, T'_{t_2}, F'_t, s'_t, A_{P'_{t_2}}, A_{T'_{t_2}}, A_{F'_t}, IP'_{t_2}, OP'_{t_2} \rangle$ are two maximum connected sub trusted components of trusted component $C_t'$, and the cohesion of both $C_{t_1}$ and $C_{t_2}$ is higher than that of $C_t$. Then it’s deemed that cutting arc $\langle x, y \rangle \in F_t$ is the most suitable as the decomposition basic for trusted component $C_t$, where $\mid P'_{t_1} \cup T'_{t_1} \mid$ and $\mid P'_{t_2} \cup T'_{t_2} \mid$ represent the amounts of elements in set $P'_{t_1} \cup T'_{t_1}$ and $P'_{t_2} \cup T'_{t_2}$.

The following two conditions can be considered when decomposing the connected trusted component $C_t = \langle P_t, T_t, F_t, s_t, A_{P_t}, A_{T_t}, A_{F_t}, IP_t, OP_t \rangle$ on the basis of cutting arc $\langle x, y \rangle$:

Condition 1: if cutting arc $\langle x, y \rangle \in P_t \times T_t$ does not lose its generality, assuming $x \in P'_{t_1}$, $y \in T'_{t_2}$, then in trusted software architecture $SA = \langle CN, LN, D_t, G_t, A_{L_t}, A_{G_t} \rangle$, after decomposing trusted component $C_t = \langle P_t, T_t, F_t, s_t, A_{P_t}, A_{T_t}, A_{F_t}, IP_t, OP_t \rangle$ on the basis of cutting arc $\langle x, y \rangle$, trusted component $C_t$ turns into two trusted components $C_{t_1}$ and $C_{t_2}$, where:

In trusted component $C_{t_1}$, the original internal status $x$ changes into the output interface of trusted component $C_{t_1}$, namely $C_{t_1} = \langle P'_{t_1}, T'_{t_1}, F'_t, s'_t, A_{P'_{t_1}}, A_{T'_{t_1}}, A_{F'_t}, IP'_{t_1}, OP'_{t_1} \rangle \cup \{ x \}$;

In trusted component $C_{t_2}$, an input interface $p$ and the arc $\langle p, y \rangle$ from $p$ to operation $y$ are added, and $A_{F'_{t_2}}(\langle s_t, t \rangle) = A_{F_t}$ (\langle x, y \rangle), namely, $C_{t_2} = \langle P'_{t_2} \cup p, T'_{t_2}, F'_t \cup p, x \rangle$.

Meanwhile, a connector $L_t$ needs to be added into the original trusted software architecture $SA_t$ to connect the output interface $x$ of trusted component $C_{t_1}$ and the input interface $p$ of $C_{t_2}$.

Condition 2: if cutting arc $\langle x, y \rangle \in T_t \times P_t$ does not lose its generality, assuming $x \in T'_{t_1}$, $y \in P'_{t_2}$, then in architecture $SA = \langle CN, LN, D_t, G_t, A_{L_t}, A_{G_t} \rangle$, after decomposing trusted component $C_t = \langle P_t, T_t, F_t, s_t, A_{P_t}, A_{T_t}, A_{F_t}, IP_t, OP_t \rangle$ on the basis of cutting arc $\langle x, y \rangle$, trusted component $C_t$ turns into two trusted components $C_{t_1}$ and $C_{t_2}$, where:

In trusted component $C_{t_1}$, an output interface $p$ and the arc $\langle x, p \rangle$ from operation $x$ to $p$ are added,
and $A_{Fi1}(x, p) = A_{Fi2}(x, y)$, namely, $C_{ii1} = P_{i1} \cup p$, $T_{ii1}$, $F_{i1} \cup \{x, p\}$, $S_{i1}$, $AP_{i1}$, $AT_{i1}$, $A_{Fi1}$, $IP_{i1}$, $OP_{i1}$).

In trusted component $C_{i2}$, the original internal status $y$ changes in to the output interface of trusted component $C_{i2}$, namely, $C_{ii2} = P_{i2} \cup T_{ii2}$, $F_{i2} \cup \{x, p\}$, $S_{i2}$, $AP_{i2}$, $A_{Fi2}$, $IP_{i2}$, $OP_{i2}$.

Meanwhile, a connector $L_i$ needs to be added to the original software architecture $SA_i$ to connect the output interface $p$ of trusted component $C_{ii1}$ and the input interface $y$ of $C_{i2}$.

**Algorithm 3** Connectivity-based trusted component decomposition algorithm (Decomposition-2 ($SA_i$, $C_{ii}$))

Input: trusted software architecture $SA_i = <CN_i, LN_i, D_i, G_i, A_{Li}, A_{Gi}>$ and its connected trusted component $C_{ii}= P_{ii} \cup T_{ii}$, $F_{ii} \cup \{x, p\}$, $S_{ii}$, $AP_{ii}$, $A_{Fi2}$, $IP_{ii}$, $OP_{ii}$.

Output: trusted software architecture $SA_i' = <CN_i', LN_i', D_i', G_i', A_{Li}', A_{Gi}'>$ obtained after decomposing trusted component $C_{ii}$.

BEGIN

Call the algorithm of the cutting arc set in connected trusted component to find the cutting arc set in trusted component $C_{ii}$;

IF cutting arc set is empty, THEN RETURN (empty cutting arc set);

IF suitable cutting arc does not exists THEN RETURN (no suitable cutting arc);

Assume the suitable cutting arc that has been found is $<x, y>$, then

$a = A_{Fi}(x, y)$; $F_{ii} = F_{ii} \setminus \{x, y\}$ /* delete cutting $<x, y>$ from component $C_{ii}$/

$N_{i1} = \{x\}; N_{i2} = \{y\}$;

FOR element $x \in P_{ii} \cup T_{ii}$ DO

IF $x \in F_{ii}$ THEN $N_{i1} = N_{i1} \cup \{x\}$$; ELSE $N_{i2} = N_{i2} \cup \{x\}$;

$C_{ii1} = C_{ii1}^{N1}; C_{ii2} = C_{ii2}^{N2}$ /* construct the two project sub components $C_{ii1}$ and $C_{ii2}$ of $C_{ii}$/

$CN_i = CN_i \cup C_{ii1} \cup C_{ii2}$;

IF $<x, y> \in P_{ii} \times T_{ii}$ THEN

BEGIN

$C_{ii1}.OP = C_{ii1}.OP \cup \{x\}$ /* internal status $x$ changes into the output interface of $C_{ii1}$/

$C_{ii2}.P = C_{ii2}.P \cup \{p\}$; $C_{ii2}.IP = C_{ii2}.IP \cup \{p\}$ /* add an input interface $p$ to $C_{ii2}$/

$C_{ii2}.A_{Pi} (p) = C_{ii1}.A_{Pi} (x)$; $C_{ii2}.F_i = C_{ii2}.F_i \cup \{x, p\}$ /* add arc $<x, p>$ to $C_{ii2}$/

$C_{ii2}.A_{Fi}(<ip, y>) = a$; $D_i = D_i \cup C_{ii1}.A_{Pi}(x)$;

$LN_i = LN_i \cup \{L_i\}$ /* add a connector $L_i$ in $LN_i$ */

$A_{L_i}(L_i) = C_{ii1}.A_{Pi}(x)$;

$G_i = G_i \cup \{<x, L_i>\} \cup \{<L_i, p>\}$ /* add arc $<x, L_i>$ to $<L_i, p>$ to architecture $SA_i$/

$AG_i (x, L_i) = a; AG_i (L_i, p) = a$

END

ELSE /* if $<x, y> \in T_{ii} \times P_{ii}$/

BEGIN

$C_{ii2}.IP = C_{ii2}.IP \cup \{y\}; C_{ii1}.P = C_{ii1}.P \cup \{p\}$;

$C_{ii1}.OP = C_{ii1}.OP \cup \{p\}$; /* add an output interface $p$ to trusted component $C_{ii1}$/

$C_{ii1}.A_{Pi}(p) = C_{ii2}.A_{Pi}(y)$; $C_{ii1}.F_i = C_{ii1}.F_i \cup \{x, p\}$ /* add arc $<x, p>$ to $C_{ii1}$/

$C_{ii1}.A_{Fi}(x, op) = a$; $D_i = D_i \cup C_{ii1}.A_{Pi}(p)$; $LN_i = LN_i \cup \{L_i\}$ /* add a connector $L_i$ in $LN_i$ */

$A_{L_i}(L_i) = C_{ii2}.A_{Pi}(y)$; $G_i = G_i \cup \{<p, L_i>\} \cup \{<L_i, y>\}$;

$AG_i (p, L_i) = a; AG_i (L_i, y) = a$

END;

FOR every connector $L_i$ ($L_i \in LN_i$) in trusted software architecture $SA_i$ DO

FOR every input interface $ip$ of trusted component $C_{ii}$ DO

BEGIN

IF $p \in N_{i1}$, THEN

BEGIN

$G_i = G_i \cup \{<L_i, C_{ii1}.ip>\}$ /* add arc $<L_i, C_{ii1}.ip>$ to $G_i$/

$AG_i (L_i, C_{ii1}.ip) = AG_i (L_i, C_{ii1}.ip)$;

END;
\[ \begin{align*}
G_t &= G_t \setminus \{ \langle L_t, C_{ii}, ip \rangle \} \quad \text{/* delete arc } \langle L_t, C_{ii}, ip \rangle \text{ from } G_t */ \\
\text{END;} \\
\text{ELSE} \\
\text{BEGIN} \\
G_t &= G_t \cup \{ \langle L_t, C_{ii2}, ip \rangle \} \quad \text{/* add arc } \langle L_t, C_{ii2}, ip \rangle \text{ to } G_t */ \\
A_{G_t} (\langle L_t, C_{ii2}, ip \rangle) &= A_{G_t} (\langle L_t, C_{ii}, ip \rangle); \\
G_i &= G_i \setminus \{ \langle L_t, C_{ii}, ip \rangle \} \quad \text{/* delete arc } \langle L_t, C_{ii}, ip \rangle \text{ from } G_i */ \\
\text{END;} \\
\text{END;} \quad \text{/* end of FOR */} \\
\text{FOR for every output interface } op \text{ of trusted component } C_{ii} \text{ DO} \\
\text{BEGIN} \\
\text{IF } op \in N_1, \text{ THEN} \\
\text{BEGIN} \\
G_t &= G_t \cup \{ \langle C_{i1}, op, L_t \rangle \}; \\
A_{G_t} (\langle C_{i1}, op, L_t \rangle) &= A_{G_t} (\langle C_{ii}, op, L_t \rangle); \\
G_i &= G_i \setminus \{ \langle C_{ii}, op, L_t \rangle \} \quad \text{/* delete arc } \langle C_{ii}, op, L_t \rangle \text{ from } G_i; \\
\text{END;} \\
\text{ELSE} \\
\text{BEGIN} \\
G_t &= G_t \cup \{ \langle C_{i2}, op, L_t \rangle \}; \\
A_{G_t} (\langle C_{i2}, op, L_t \rangle) &= A_{G_t} (\langle C_{ii}, op, L_t \rangle); \\
G_i &= G_i \setminus \{ \langle C_{ii}, op, L_t \rangle \} \quad \text{/* delete arc } \langle C_{ii}, op, L_t \rangle \text{ from } G_i */ \\
\text{END;} \\
\text{END;} \quad \text{/* end of IF */} \\
\text{END;} \quad \text{/* end of FOR */} \\
CN_t &= CN_t \setminus \{ C_{ii} \} \quad \text{/* delete isolate trusted component } C_{ii} */ \\
\text{END.}
\end{align*} \]

**Conclusion**

As one of the most valuable and challenging core topics in the computer software research field, trusted software has attracted enormous attention from government organizations, science field and industry field at home and abroad.

In authors’ previous work, a trusted component model based on Or-transition Colored Petri Net is proposed. The characteristics of the trusted component and the influence of cohesion and coupling on the trusted software architecture are analyzed. Furthermore, a trusted component decomposition algorithm based on software cohesion is given. Based on previously proposed component description and modeling methodologies of Or-transition Colored Petri Net, this paper proposes a Decomposition algorithm of Connected Trusted Component, so as to ensure the credibility of overall software system evolution.

**Acknowledgments**

This work has been supported by the National Science Foundation of China under Grant No. 61462091, by the Data Driven Software Engineering innovation team of Yunnan province No. 2017HC012, by the Science Foundation of Key Laboratory in Software Engineering of Yunnan Province under Grant No. 2017SE205, by the eighteenth batch of Yunnan Province in the young academic and technical leaders reserve personnel training project under Grant No. C614300.

**References**


