Petri Net-Based Modelling and Analysis of Context Aware System Considering Context Timeliness

Hong-li JIANG, Hui-fang LI and Bai-hai ZHANG
School of Automation, Beijing Institute of Technology, China

Keywords: Context awareness, Petri net, Time constraints, Model merging.

Abstract. Context-aware applications become very popular due to its providing intelligent and customized services to the front-end users. However, most of them focus on context management flow and don’t consider time constraints of context and business process. To make full use of context information in business environment and provide more accurate services, this paper extends CPN with context and proposes a merging modeling approach considering business process, context and its timeliness. Then to verify whether the time constraint of merged model is satisfied, the available time allowance algorithm is presented. Eventually, a case of intelligent care exemplifies the effectiveness of the model and algorithm.

Introduction

Many researches have conducted the context management and modeling of context aware system. They usually think the context is always valid and can successfully trigger the corresponding services. However, context is dynamic and has timeliness. If the context is invalid in the business process execution, the process will be interrupted. By introducing the business process to modeling of context aware system, we can judge whether the context is valid at given time and provide precise services. So how to model and analyze context timeliness in business process is challenging.

The modeling of context aware system should include the context management and the description of business process. As a graphical and mathematical modeling language, Petri net is recognized as a good way to model business process. Meanwhile, Petri net has the function of dynamic property analysis, which can also be used to describe dynamic context information. So when modeling the context aware systems, CPN can be utilized to model both the context information and business process. But how to merge the business process with the context in context aware systems is challenging.

To solve the problems above, this paper focuses on how to model and analyze the context timeliness of context aware systems. The main contributions of this paper are as follows. (1) It extends the CPN net to describe the business process and context. Context is merged into the business process modeling and its timeliness is expressed as extension elements of Petri nets. (2) It explains the merging procedures of business process and context. (3) It introduces context timeliness to time constraints of business process and proposes an available allowance algorithm, which improves the accuracy of context aware services.

Related Work

Context-aware systems need to deal with lots of complex context information to provide intelligent services and satisfy personalized demands. So the suitable modeling method should have good theoretical support and can help manage the context information. Petri net [1] is a theoretical modeling tool with precisely mathematical expression and intuitionally graphical presentation for context aware systems.

Some researchers have studied the modeling and analysis of context-aware systems based on Petri net model. Lu et al. [2] presented a scenario-based method to design context-aware service using Petri net. Han S et al. [3] proposed to use Petri net to model context relationship and dependencies. These
works focused on context description, paying no attention to adapt context to the business process. Some works [4, 5] introduced context as constraints to business process modelling and ignore context timeliness. Lin X. et al. [6] proposed context reasoning real-time scheduling algorithm based on freshness. They explained the context time constraint but didn’t consider the relationship between context and business process.

While some works consider both business processes and context, they just use very limited context information and lack the required context reasoning application. In order to make full use of context information in business environment and provide more intelligent services for users, this paper extends CPN with context and proposes a merging modeling approach with not only business process, and context information as well as its timeliness into consideration. Then to verify whether the time constraint of merged model is satisfied, the available time allowance algorithm is presented.

Approach and Methodology

To model and verify context timeliness, based on traditional CPN [1], this paper proposes an extension model called Context Aware Time Petri Nets (CATPN) which contains the context information and its timeliness in the model. Moreover, the paper proposes a merging modeling approach and available time allowance algorithm to verify whether the time constraint of merged model is satisfied.

Formal Definition of Extended Petri Nets

In order to introduce context to business process when modelling context aware systems and analyze whether the context is valid in the business process execution, the paper extends CPN with context and context timeliness. The extended CPN is expressed as follows:

\[ CATPN = (P, T, A, \Sigma, C, E, G, TC, TK, CF, TF, M) \]

where

- \( P = P_M \cup P_C \), \( P_M \) and \( P_C \) represent the finite sets of places, and satisfies \( P_C = P_{cin} \cup P_{cref} \), \( P_{cin} \) and \( P_{cref} \) is respectively the finite sets of initial context places and reasoning context places;
- \( T = T_M \cup T_C \), \( T_M \) and \( T_C \) represent the finite sets of activity transitions and context operation transitions in the context model, denote \( type(T) \) as related activities or operations of transitions;
- \( A = A_M \cup A_C \), \( A_M \) and \( A_C \) represent the finite sets of directed arcs;
- \( \Sigma = \Sigma_M \cup \Sigma_C \), \( \Sigma_M \) and \( \Sigma_C \) respectively represent the type set in business process and a finite set of non-empty context, and satisfies \( \Sigma_C = \{ \text{Context}_i | i = 1, 2, \cdots, n \} \), \( \text{Context}_i \) represents the \( i \)-th context type;
- \( TC : (P \rightarrow Z \times Z) \vee (T \rightarrow Z \times Z) \), represents the time mapping function for places and transitions, mapping the places to time interval \([TC_{min}(p), TC_{max}(p)]\) and mapping the transitions to \([TC_{min}(t), TC_{max}(t), TC_{d}(t)]\), where \( TC_{d}(t) \) represents the duration time.
- \( TK : P \rightarrow \{(p, \text{Context}_i, f_{start}, f) | p \in P, \text{Context}_i \in C(p)_{init}, f_{start}, f \in \mathbb{Z}^+ \} \), represents the possible color mappings of Places, \( f_{start} \) represents the creation time of colored token, \( f \) represents token freshness, denotes \( TK(p, \text{Context}_i) \) as token that its color is \( \text{Context}_i \), and its freshness is \( f_{TK}(p, \text{Context}_i) \);
- \( CF = \{f_1, \cdots, f_i, \cdots, f_k\} \), represents the set of context freshness, and \( f_i \in (0, \infty) \);
- \( TF : T_C \cup P_C \rightarrow L(CF) \), represents the non-negative linear function of freshness of context token that \( T_C \) produces or \( P_C \) owns, \( L(CF) = a_1 f_1 + a_2 f_2 + \cdots + a_j f_j + \cdots + a_k f_k, i = 1, \cdots, n, a_i \in \{0,1\} \), corresponding to \( \text{Context}_i \), \( a_j = 0 \) means places don’t have the token \( \text{Context}_i \), \( j \) represents the \( j \)-th context attribute;
Merging Modeling Approach

In order to analyze the context timeliness in business process, we merge the business process with context models. As expressed above, we model a business process with $M_p = (P_p, T_p, A_p, \Sigma_p, C_p, E_p, G_p, \Sigma_p, M_p)$, and describe a context model corresponding to activity $t_i$ by $N_c = (P_c, T_c, A_c, \Sigma_c, C_c, E_c, G_c, \Sigma_c, M_c)$. $N_c^j$ represents the $j$-th context model of the $i$-th activity.

The equivalent transition is to simplify models.

Definition 1: Given two context models $N_c^{j1}$ and $N_c^{j2}$, $t_1 \in T_c^{j1}, t_2 \in T_c^{j2}$, and satisfies $\forall p_i \in (t_1), p_j \in (t_2)$, and $type(t_1) = type(t_2)$, we have $type(p_i) = type(p_j), E(p_i, t_1) = E(p_j, t_2)$, and $G(t_1) = G(t_2)$, then $t_1$ and $t_2$ are equivalent. We can simplify the two context models.

The model merging is divided into three steps: firstly, the business process and context model are merged through shared transitions; secondly, equivalent transitions are chosen and combined; thirdly, context models and context places in newly combined model are merged based on same context type.

Time Performance Analysis

Traditional schedulability analysis [7,8] focuses on verifying time constraints of process dimension and neglects the valid time of instance. That means the instance is always valid if the business process doesn’t end. However, in context aware system, context may change over time. Context instance has limited available time. So when analyzing the schedulability of context aware systems, we consider the context timeliness. We define the schedulability in the context aware system as verifying whether both the valid time of context instance and process instance can satisfy time constraints of business process.

The paper proposes schedulability analysis method considering context timeliness. Firstly, enable time of transition is adjusted based on context timeliness. Secondly, context start time is updated according to the time constraints of output of transition and places at the same level. When the time constraints are violated, both context start time and the time constraints of activity in business process can be adjusted.

$$
EN_{\text{max}}(p_i) = TK_{\text{max}}(p_i) + TC_{\text{max}}(p_i), p_i \in P_c
$$

$$
EN_{\text{max}}(p_i) = TK_{\text{max}}(p_i) + \min\{\min_{\forall p_i \in (t_1), p_j \in (t_2), \text{type}(t_1) = \text{type}(t_2)} (E(p_i, t_1) + E(p_j, t_2))\}
$$

$$
EFBT(t_i) = \max\left\{\max_{i,j}\left\{TK_{\text{max}}(p_i) + TC_{\text{max}}(p_j)\right\}, \max\{EN_{\text{max}}(p_i)\}\right\} + TC_{\text{max}}(t_i)
$$

$$
LFET(t_i) = \min\left\{\min_{i,j}\left\{TK_{\text{max}}(p_i) + TC_{\text{max}}(p_j)\right\}, \text{EFBT}(t_i) + TC_{\text{max}}(t_i) - \min\{EN_{\text{max}}(p_i)\}\right\}
$$

$$
where \quad p_i \in (t_1), p_j \in (t_2), \text{type}(t_1) = \text{type}(t_2)
$$

$$
LFBT(t_i) = LFET(t_i) - TC_{\text{max}}(t_i)
$$

Eq.1 is used to calculate the enable time of context places. Eq.2 is for schedulability analysis of transitions. Eq.3 is for $LFBT$. To analyze the schedulability of transitions and calculate the allowed delay time of activities, we propose the available time allowance (ATA) algorithm as follows:
ATA algorithm

Input: Combined model $N_{\text{complete}}$, start time of process $t_0$, context relative time $\theta_i$, time constraints of $t_i, p_i$, initial mark $m_0$;
Output: Available time allowance of each business process activities;
1. While (T not null){
2. Find the enable transition $t_i$ under marking $m_0$;
3. Compute the start time of context token in place $p_i$, initial start time is $T_0 + \theta_i$;
4. Compute the enable time of context places according to Eq.1;
5. Make sure the enable time of $t_i$ under the constraints of process places and context places, if not satisfied adjust the start time of context token, change $\theta_i$;
6. Compute the $EFTB, LFET$ of $t_i$ according to Eq.2 , if $t_i$ is non-schedulable, adjust the time constraints of transition or precursor places or the start time of context tokens, repeat step 6;
7. compute the $LFET$ according to Eq.3;
8. update the start time of context token according to $TK_{\text{arr}}(p_i) = \min \{FR_{\text{arr}}(t_i)\}$;
9. Compute the ATA of each transition by $\text{ATA} = LFET(t_i) - EFTB(t_i)$;
}

Intelligent Care Case Study: Modeling and Analysis

An emergency assistance process in intelligent care service is modeled and analyzed to illustrate the concept and method of proposed model. The case is intended to handle a patient fall process in an adaptive way to the execution context and verify our method to adjust patient’s rescue time.

Description of Business Process and Context

In the patient falls handling process, we consider six sub activities. The business process is shown in Figure 1. The context models are shown in Figure 2. In this case, five types context are included. $ContextType = \{\text{PhysioContext}, \text{StaticContext}, \text{UserContext}, \text{LocationContext}, \text{ResourceContext}\}$, $\text{PhysioContext} = \{<\text{pressure}, 1h>, <\text{rate}, 30\text{min}>, <\text{temp}, 15\text{min}>>$, $\text{StaticContext} = \{<\text{pressure}, \infty>, <\text{rate}, \infty>, <\text{temp}, \infty>>$, $\text{UserContext} = \{\text{fallen}, 8\text{min}\}$, $\text{LocationContext} = \{\text{room}, 5\text{min}\}$, $\text{ResourceContext} = \{\text{doctor}, 30\text{min}>, <\text{nurse}, 2h>, <\text{caregiver}, 8h}\}$.

![Image](https://via.placeholder.com/150)

Figure 1. The fall handling process model.
Merging Modeling

The business process contains six activities. Five activities have related context models. According to the proposed merging steps in merging modeling approach. The complete model is shown in Figure 3.

Time Performance Analysis

Suppose, the start time of business process is $T_0$, initial time of context is $\theta_1$, then abnormal context is perceived at $T_0 + \theta_1$. The colored tokens are $TK_{p_1} = (\theta_1,\theta_1,8)$ and $TK_{p_2} = (\theta_2,\theta_2,\theta_1,5)$, the valid time of valid token in $p_1$ is expressed as $[\theta_1, \theta_1 + 8]$, so the enable time is $[\theta_1 + 2, \theta_1 + 8]$. Similarly, the enable time of other context places for different activities can calculate. According to ATA algorithm, activities and their computing results are shown in Table 1.

### Table 1. Time constraints of activities transitions.

<table>
<thead>
<tr>
<th>Transitions</th>
<th>Valid Freshness</th>
<th>Context Enable Time [min] (relative to $T_0 + \theta_1$)</th>
<th>EFBT [min]</th>
<th>LFBT [min]</th>
<th>ATA [min]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>$(p_{1,1},\theta_1,8),(p_{1,2},\theta_1,5)$</td>
<td>[2,5]</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>$t_2$</td>
<td>no contexts</td>
<td>no contexts</td>
<td>5</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>$t_3$</td>
<td>$(p_{1,1},\theta_1+4,8),(p_{1,2},\theta_1+9,15),(p_{1,2},\theta_1,\infty)$</td>
<td>[9,24]</td>
<td>9</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>$t_4$</td>
<td>$(p_{1,2},\theta_1+4,5)$</td>
<td>[4,9]</td>
<td>7</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>$t_5$</td>
<td>$(p_{2,2},\theta_1+9,30)$</td>
<td>[9,39]</td>
<td>19</td>
<td>19</td>
<td>0</td>
</tr>
<tr>
<td>$t_6$</td>
<td>$(p_{1,2},\theta_1+39,8),(p_{1,2},\theta_1+44,15),(p_{1,2},\theta_1+44,\infty)$</td>
<td>[44,59]</td>
<td>45</td>
<td>58</td>
<td>13</td>
</tr>
</tbody>
</table>

In Table 3, the second column represents the start time and freshness of tokens in context place. The last four columns respectively represent the valid time of context place, the earliest fire beginning time, the latest fire beginning time and the available time allowance. They all are relative to $T_0 + \theta_1$, which means the real values should add $T_0 + \theta_1$. The activity $t_4$ adjusts $TC_{\min}(t_4)$ from 3 to 0. The activity $t_4$ adjust $C_{\min}(t_4)$ from 1 to 0 and The activity $t_6$ adjust the start time of tokens from $\theta_1 + 14$ to $\theta_1 + 44$. The whole process will 46 min at least. The activities $t_1$, $t_2$, $t_3$ and $t_6$ can have delay time.
while $t_4$ and $t_5$ need to be executed immediately after being fired. So, by using ATA algorithm, we can analyze the schedulability of the activities, adjust the unsuitable time constraints and confirm the start time of context. If time constraints are not right, we adapt the process time constraints and change the start time of context.

Conclusion

By using the emerging networking technology and sensor technology, context aware systems can provide intelligent and customized services to the front-end users. However, context is not always valid. A method of verifying context timeliness is needed. By introducing context information to business process when modelling context aware systems and analyzing the validity of context in the business process execution, the context aware systems can judge whether the context is valid at given time and then satisfy the precise requirements of the provided services.

The proposed Context-aware Time Petri net can solve the problems of how to model and analyze context timeliness in business process. Firstly, this paper extends CPN with context and proposes a merging modeling approach with not only business process, and context information as well as its timeliness into consideration. Secondly, to verify whether the time constraint of merged model is satisfied, the available time allowance algorithm is presented. A case of patient fall handling exemplifies the proposed model and algorithms. The extended CPN and proposed algorithm can be used to model and analyze context aware systems.

References


