This is the published version (version of record) of:


Available from Deakin Research Online:

http://hdl.handle.net/10536/DRO/DU:30005715

©2005 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Copyright : 2005, IEEE
Supporting Adaptive Learning in Hypertext Environment: A High Level Timed Petri Net Based Approach

Shang Gao, Zili Zhang
School of Information Technology
Deakin University
Geelong, VIC 3217, Australia
{shang, zzhang}@deakin.edu.au

Igor Hawryszkiewycz
School of Computing Sciences
University of Technology, Sydney
PO Box 123 Broadway, NSW 2007, Australia
igorh@it.uts.edu.au

Abstract

One problem for hypertext-based learning application is to control learning paths for different learning activities. This paper first introduced related concepts of hypertext learning state space and Petri net, then proposed a high level timed Petri Net based approach to provide some kinds of adaptation for learning activities. Examples were given while explaining ways to realizing adaptive instructions. Possible future directions were also discussed at the end of this paper.

1 Introduction

Web-based hypertext is an ideal e-learning tool which reveals interconnections or complex interdependencies among ideas on net [1]. Educational hypertext is made up of a group of knowledge nodes related to a specific knowledge field and connected to each other with links. Too many links, however, not only add additional cognitive overhead, but also increase the chance of losing knowledge nodes.

We are currently working on solutions not only to support the flexibility of hyperlink, but also to provide adaptive control over learning activities. For instance, allowing students to access specific advanced knowledge nodes after they have already known all the required prerequisite knowledge. Inspired by ideas presented in [2][3][4] and [5], a model of hypertext learning state space was proposed [6]. By defining the learning state space and manipulating learning-state-transition thresholds, students’ learning paths are controlled, while at the same time flexibility of the hypertext is maintained.

Various methods can be applied to manipulate learning-state-transition threshold [6]. Given the similarity between learning state space and Petri Net model, in this paper, a high level Petri Net based approach is introduced to interpret the browsing semantics of learning state space.

The whole paper is organized as follows: an overview of related concepts of hypertext learning state space and high level Petri Net are given as background knowledge in the next section, followed by the description of the proposed high level timed Petri Net model and its various adaptive operations on browsing semantic structure of hypertext learning state space. Future research directions and possible extensions to this approach are also discussed at the end of this paper.

2 Background knowledge

2.1 Hypertext learning state space

Based on the knowledge space theory and Albert’s mathematical model of hypertext structure [2][3][4], a hypertext learning state space is proposed to reflect the characteristics of educational hypertext and students’ learning paths in hypertext. (For formal descriptions, refer to [6].)

In an educational hypertext, a basic information unit expresses certain didactical information, which is called a knowledge node. An outside link connects a source anchor of a learning state and a destination anchor of another learning state. An inside link connects a source anchor of a learning state and a destination anchor of the same state. A hypertext learning state consists of a subset of knowledge node, a subset of source anchor and a subset of destination anchor to which the subset of knowledge node is related.

state space, as visualized in Figure 1, each of which may include multiple interconnected nodes.

![Figure 1. A example of 18-hypertext learning state space](image)

A hypertext learning state space is a space that has one starting learning state, at least one ending learning state and no loop among its learning states, namely, a learning state cannot become its own prerequisite state. In the space shown in Figure 1, the starting state is p₁ “Introduction”, and p₁₈ “Conclusion and review” is the only ending state.

If a knowledge node has an outside link, it is a key knowledge node. Accessing key knowledge nodes is one of the prerequisites to trigger learning state transition.

A sequence of learning states forms a learning path in a hypertext learning state space, where any two adjacent learning states find each other by outside links that reside in key knowledge nodes. In Figure 1, after browsing three basic learning states p₁, p₂ and p₃, students are free to choose the next one from three parallel candidates p₄, p₈ and p₁₄, each of which also has optional learning paths suitable for demands at different levels.

As you may have noticed, in the above HTML tutorial scenario, students are not required to study all the “Client and server side scripting” nodes before entering state p₇, which implies outside links pointing at p₇ are OR links. p₁₃ and p₁₇ are the same as p₇. However, p₁₈ requires the access of all the three preceding states p₇, p₁₃ and p₁₇, which implies outside links pointing at p₁₈ are AND links. Distinguishing AND OR link relations is important for applying appropriate learning control mechanisms to hypertext learning state space.

2.2 Petri Nets and high level Petri Nets

Invented by Carl Adam Petri, a Petri Net (PN) [8] consists of two kinds of nodes called places (represented by circles) and transitions (represented by bars) where arcs connect them. Input arcs connect places with transitions, while output arcs start at a transition and end at a place.

Petri nets have also been extended over the years in many directions including time, data, and hierarchy. To describe the temporal behavior of a system, time can be associated with places (called P-timed), tokens (called age) and transitions (called T-timed). For instance, in a P-timed Petri Net (P-TPN), if one time attribute is associated with place, the firing rules are that a transition is enabled after tokens deposited in its input places take a fixed, finite amount of time delay. During that period, the tokens are not available. After the time delay, the transition becomes enabled. If fired, tokens are moved into the output places of that transition.

If two time attributes are adopted, one is defined as the minimum delay $d^{min}$ and the other as maximum delay $d^{max}$, the firing rules are that a transition is enabled after the minimum delay $d^{min}$, it remains enabled in $[d^{min}, d^{max}]$ interval; if after the maximum delay $d^{max}$, the enabled transition has not been fired, it is forced to do so, moving tokens from its input places to output places. If the transition can not fire, the token becomes unavailable. This “dead end” should be avoid by setting appropriate $(d^{min}, d^{max})$ and adjusting them dynamically.

3 P-timed Petri Net based adaptation model for hypertext learning state space

In a hypertext learning state space, each state includes attributes such as content, number and structure of knowledge nodes, recommended learning time, and link relation with other states, etc. Different students have different capabilities, which results in different learning paths and time. To provide personalized instructions, a good way could be adjusting state attributes to adapt to individual activities using personal learning agents, for instance, displaying further reading materials after assessment, or reducing advanced contents if students are found not doing very well. While providing dynamics, the underlying structure should be kept intact. In other words, students are only presented to personalized interface with the original underlying structure kept untouched for archival purpose.

Timed Petri Net can be used to model temporal events in a discrete system. Students’ learning activities in Web-based environment are series of discrete events, such as reading hypertext, typing words and clicking hyperlinks. These activities help students move in hyperspace. If the hypertext learning state and these activities, especially link-clicking, could be modeled with a timed Petri Net, we might be benefiting from dynamic executive semantics of the Petri Net, embedding path control information in Petri Net structure, and attempting realizing learning adaptation.

3.1 P-timed Petri Net based adaptation model

The first step is to build a P-timed Petri Net based model for hypertext learning state space. Mapping learning state onto place is quite straightforward. The complex process is how to convert outside link relation to transition in P-timed
Petri Net model, at the same time remaining the prerequisite constraint and browsing flexibility in learning state space. Extra care should be taken while mapping AND OR links to transitions. There are several typical structures to consider:

- **Sequence (Figure 2)**

![Figure 2. Modeling sequence structure to P-timed Petri Net](image)

- **Merging (Figure 3)**

![Figure 3. Modeling merge structure to P-timed Petri Net](image)

- **Forking (Figure 4)**

![Figure 4. Modeling fork structure to P-timed Petri Net](image)

Adding such a self-cyclic arc in Figure 4“OR forking” is due to the consideration of multiple paths available at one time, which should still be available when students attempt other parallel paths.

- **Multi-path (Figure 5)**

![Figure 5. Modeling multi-path structure to P-timed Petri Net](image)

Before minimum delay $d_{min}$, the outside link is not click-able or visible. Students are required to concentrate on current learning state and freely select knowledge nodes inside the state. A transition (which is an outside link) $t$ is enabled only when each of its input places $p_i$ has one token and is still in its $(d_{min}^{min}, d_{max}^{max})$ delay interval, where the $(d_{min}^{min}, d_{max}^{max})$ is measured relatively to the time the place receives a token. In the $(d_{min}^{min}, d_{max}^{max})$ interval, students are free to select visible links, transferring to other related states. If none of links is selected by $d_{max}$, an enabled transition is chosen to fire by the system automatically. Otherwise, current content becomes unavailable and the students go back and try other paths. The predefined firing priority of transitions usually reflect teaching preference.

For instance, in Figure 6, place $p_1$ is visible because of the token deposited in it. Its output transition $t_1$ is immediately click-able due to the $0$ value of $d_{min}^{min}$. Students can browse knowledge nodes of $p_1$ without any time constraint because $d_{max}^{max}$ is set to $\infty$. If $t_3$ is fired in sequence, $p_4$, $p_6$ and $p_{10}$ become visible simultaneously. Student can stay in $p_4$ for at most 50 time units before forced to leave for $p_5$, $p_6$ or $p_7$. Assuming $p_5$ is chosen because of its smallest $d_{min}^{min}$ value 6, students are required to spend at least 6 time units in it. After the 15 time units, $t_5$ is fired and students enter $p_7$.

Of course students can also choose $p_4$ or $p_6$ if the value of $d_{min}^{min}$ or $d_{max}^{max}$ is set to be the smallest one. If $d_{min}^{min}$ are
the same, then consider $d^{max}$.

To activate $t_{21}$, students should have accessed $p_7$, $p_{13}$ and $p_{17}$ in whatever order. Only when all the three input places $p_7$, $p_{13}$ and $p_{17}$ have tokens and are in their valid intervals, could transition $t_{21}$ be fired.

If delay pairs are set inappropriately and no overlapping intervals exist in related places, a “dead-end” could possibly be happening. This problem could be solved by agents which monitors students’ status and learning paths, adjusting delay pair values accordingly based on predefined teaching model.

### 3.3 Timing control in P-timed Petri Net model

Various timing control on place $p_i$ can be easily achieved by setting attributes pair $(d^{min}_i, d^{max}_i)$ to some particular values, as shown in Table 1.

Different delay settings result in different learning activities. Students are required to study for time defined by $d^{min}_i$ and leave before time defined by $d^{max}_i$. The interval provides flexibility of focusing on detail or choosing to leave.

<table>
<thead>
<tr>
<th>Timing requirement</th>
<th>Delay attribute pair setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal learning time</td>
<td>$d^{min}_i &lt; d^{max}_i$</td>
</tr>
<tr>
<td>No maximum constraint</td>
<td>$d^{min}_i = d^{max}_i$</td>
</tr>
<tr>
<td>No any time constraint</td>
<td>$d^{min}_i = 0, d^{max}_i = \infty$</td>
</tr>
<tr>
<td>Learning skip</td>
<td>$d^{min}_i = d^{max}_i = 0$</td>
</tr>
<tr>
<td>Strict learning time</td>
<td>$d^{min}_i = d^{max}_i \neq 0$</td>
</tr>
<tr>
<td>Learning blocked</td>
<td>$d^{min}_i = d^{max}_i = \infty$</td>
</tr>
</tbody>
</table>

Figure 7(a) illustrates a simply case where the target place $p_{10}$ has only one successor place $p_{11}$. $p_{10}$ is skipped by setting $(d^{min}_{10}, d^{max}_{10})$ to $(0, 0)$. For a more complex situation like (b), target place $p_8$ has several successors with different delay pairs. If $(d^{min}_8, d^{max}_8)$ is set to $(0, 0)$, a token from $p_8$ bypasses $p_8$ and flows into $p_{12}$ directly because it has the smallest $d^{max}_8$ value $50$. Which enabled transition to choose depends on the teaching model and the policies applied. For instance, if faster learning is preferred, the place with the smallest $d^{min}$ would be the first choice. $d^{max}$ is the next variable to consider if related $d^{min}$ values are the same, like $p_9$, $p_{10}$, $p_{12}$ and $p_{13}$ in Figure 7(b).

Place comes to life after changing $(0, 0)$ back to its original value.

### 4 Adaptive operations on P-timed Petri Net model

As discussed above, different learning paths can be produced by setting $(d^{min}, d^{max})$ pair. Several simple adaptive operations could also be accomplished by this means.

### 4.1 Content adaptation

Content adaptation corresponds to displaying or hiding places. For instance, in P-TPN based model, a place is skipped by setting its delay pair to $(0, 0)$, which is called learning skip. When a token flows into this place, no delay is permitted and it enters its successors instantaneously. If no such a successor exists, students try other paths, or accept agents’ advise. On all accounts, students are not able to see any contents contained in this place.

Figure 7(b) demonstrates how deletion of a transition (or link) can be accomplished. Case (a) is simple where the target transition $t_{12}$ has only one output place, $t_{12}$ and all its followed places are removed by setting $(d^{min}_{12}, d^{max}_{12})$ to $(\infty, \infty)$. For a more complex situation like (b), target transition $t_8$ has three parallel transition $t_{11}$, $t_{14}$ and $t_{10}$. $(\infty, \infty)$ for $p_8$ definitely makes $t_8$ never be chosen, however, $(81, 81)$ for $p_9$ also works given $d^{max}_{13}$ is 80.

A transition would recover after changing its output places’ $(d^{min}, d^{max})$ back to their original values.
4.3 Timing and priority adaptation

Timing adaptation could also be achieved by dynamically adjusting \( d^{\text{min}}, d^{\text{max}} \). When authoring educational hypertext, timing attributes are set by default. Examination timing usually is reasonable, however, predefined timing for some learning state might not be appropriate. For instance, students devote too much time to some details and do not finish browsing before recommended maximum delay. Under such kind of circumstances, adjustment is necessary and important for effective instruction.

Changing \( d^{\text{min}}, d^{\text{max}} \) affects the transition sequence (or learning path). A place \( p_i \) receives higher priority if \( d_i^{\text{min}} \) set to be the smallest among all the parallel places. Its priority degrades when granted a larger \( d_i^{\text{min}} \). Figure 9 illustrates this situation.

5 Discussion and conclusion

There are many situations in which this adaptation mechanism can be applied, for instance, temporal activity coordination in cooperative working environment and discrete event management in a time-sensitive system.

Besides of timed Petri Net, colored Petri Net is an ideal tool to describe multiuser behavior in an event-driven environment, where different users are granted different colored tokens. Transition Enabling and firing is decided by classified colors held by a place and the color consuming function defined for each arc. In our case, if students at different levels are represented by different colored tokens, their access control becomes a matter of color allocation and consumption.

In our future research, we are interested in applying colored timed Petri Net in the learning state space, also planning to introduce timed Petri Net based adaptation to collaborative work environment.

References


