http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 8 (7): 1054-1058, 2009
ISSN 1812-5638

Modeling Multimedia Synchronization using Petri Nets

[1]Wei Liu and [1,2]Yuyue Du
[1]College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266510, China
[2]State Key Laboratory of Computer Science, Chinese Academy of Sciences, China

Abstract: In this study, Logical Time Interaction Petri Nets (LTIPN) were designed to describe multimedia synchronization based on the previous models. In the model, we introduce logical expressions which are used to describe passing value indeterminacy in an logical time Petri net to model multimedia synchronization. And all multimedia synchronization events including multimedia objects are expressed by transitions of Petri nets, while the previous models mostly use places of Petri nets to express multimedia objects. This study provides users simple and intuitive modeling approaches. Basic temporal relations between multimedia objects, multimedia synchronization strategies and user interactive operations can be represented simply and explicitly by the LTIPN.

Key words: Multimedia synchronization, Petri nets, user interaction, logical expressions

INTRODUCTION

Petri nets (Du *et al*., 2008, 2009) are a powerful model and can model both the concurrency and parallelism in a natural way.

For the purpose of modeling timed-driven systems, the notion of time was introduced in Petri nets. They are known as Timed Petri Nets (TPN) (Du *et al*., 2007; Vicario, 2001), which allow all kinds of synchronization specifications.

There have been several models for describing multimedia synchronization based on TPN.

Object Composition Petri Nets (OCPN) (Little and Ghafoor, 1990) is based on a TPN and can be used for modeling synchronization requirements for multimedia objects. The OCPN augments Petri nets with values of time and resources used in the places of the net. The execution of the OCPN is similar to that of TPN where the transition firing is assumed to occur instantaneously and the places are assumed to have states.

Extended OCPN (XOCPN) model (Woo *et al*., 1994) takes into account the demands of isochronous data, requiring a rate-controlled transmission. It is used under a network environment where several configurations are possible for a distributed multimedia information system.

Using the OCPN and XOCPN model, it is impossible to describe modifications of the presentation sequence by a user. For instance, a user may wish to stop presentation, reverse it or skip a few frames. These operations cannot be described in the existing OCPN architecture.

Dynamic Timed Petri Nets (DTPN) (Prabhakaran and Raghavan,1993) allows user participation to preempt the execution sequence and modify the duration of time associated with the preempted net process. This structure can be adopted to model multimedia synchronization characteristics with dynamic user participation.

Time Stream Petri Net (TSPN) (Senac *et al*., 1996) introduces a unified formal model for the complete and accurate specification of both temporal and logical (i.e., link) synchronization within hypermedia distributed and weakly synchronous systems. This new model extends time Petri nets with hierarchical design capabilities and new firing rules.

However, firing rules of DTPN and TSPN must be modified. As a result, existing analysis methods of Petri nets theory cannot be used to analyze their properties.

A logical time Petri net (Du *et al*., 2007) is presented based on Petri nets. So, is a logical Petri net (Du and Guo, 2009). Through attaching logical expressions to some actions of a logical time Petri net model, it can model passing value indeterminacy and describe batch processing function of cooperative systems which the existing formal techniques cannot model.

There also exists passing value indeterminacy in multimedia synchronization when some unexpected situations occur such as data blocking, package losses and blocking timer timeout. In this case, The transitions will be fired forcibly even if data do not arrive.

In this study, logical expressions which are used to describe passing value indeterminacy in a logical time Petri net are introduced to model multimedia

Corresponding Author: Wei Liu, College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266510, China

synchronization. And by referring to the previous models such as OCPN, XOCPN, DTPN and TSPN, we present a multimedia synchronization model LTIPN which can simply and explicitly represent basic temporal relations between multimedia objects, multimedia synchronization and user interaction. This model provides users simple and intuitive modeling concepts. In present model, all multimedia synchronization events including multimedia objects are all expressed by transitions of Petri nets, while the previous models mostly use places of Petri nets to express multimedia objects.

LOGICAL TIME INTERACTION PETRI NET MODEL

Definition 1: A logical time interaction Petri net is a 10-tuple,

$$LTIPN = (P, T, F, Dt, M, St, I, Dm, E, M)$$

where, P is a set of places; $T = T_M \cup T_S \cup T_I \cup T_D$ is a set of transitions; T_M is a set of multimedia object transitions which denote multimedia objects, T_S is a set of logical synchronization transitions which denote multimedia synchronization, T_I is a set of user interaction transitions, T_D is a set of delay transitions; P, T_M, T_S, T_I and T_D are disjunct sets;

$F \subseteq (P\times(T_M\cup T_S\cup T_I\cup T_D))\cup((T_M\cup T_S\cup T_I\cup T_D)\times P)$ is a set of arcs;

Dt is a function such that $\forall t\in T_D$, $Dt(t) \in R$, which denotes the delay time of t;

M is a mapping function such that $\forall t \in T_M$, M(t) is a multimedia object and $M(t) \in \{m_1, m_2, ..., m_n\}$;

St is a mapping function such that $\forall t\in T_s$, St(t) is a logical expression f_s;

I is a mapping function such that $\forall t\in T_I$, I(t) is a user interactive operation and $I(t) \in$ {skip, back, replay, pause, resume};

Dm: $T_M \rightarrow (R^+ \cup \{0\})\times(R^+ \cup \{0\}) \times (R^+ \cup \{0\})$, $\forall t\in T_M$;

$Dm(t)\rightarrow(x, n, y)$, where the 3-tuples associated with multimedia object transitions represents its earliest, nominal and latest execution duration of time, respectively;

E: $T_M \rightarrow R$. $\forall t\in T_M$, $E(t)\rightarrow R$, where R represents the remaining execution duration of time;

M is a marking function.

Graphically, multimedia object transitions are drawn as bars; synchronization transitions are drawn as the rectangles in which the symbol S is embedded; user interactive transitions are drawn as the rectangles in which the symbol I is embedded; delay transitions are drawn as the rectangles in which the symbol D is embedded.

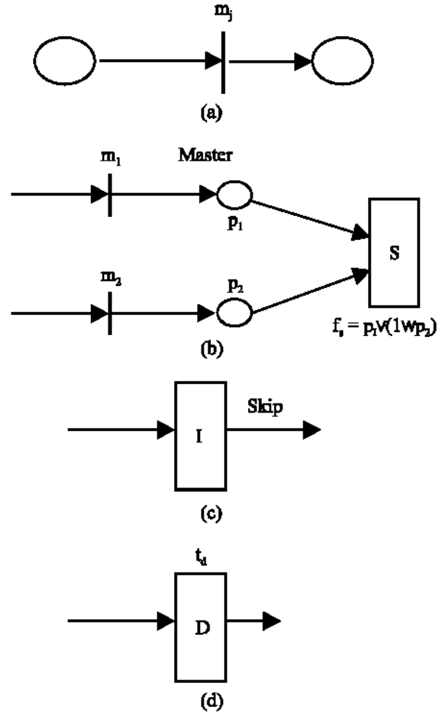

Fig. 1: The LTIPN representations of four classes of transitions, (a) the representation of a multimedia object transition, (b) the representation of a logical synchronization transition, (c) the representation of a user interactive transition and (d) the representation of a delay transition

For the clarity, (x, n, y) is not marked in Fig. 1a-d.

The LTIPN representations of multimedia object transitions, logical synchronization transitions, user interactive transitions and delay transitions are shown in Fig. 1.

Definition 2: Firing rules of the transitions in LTIPNs:

- $\forall t \in T_M$, if $\forall p\in \bullet t$, $M(p) = 1$, t is said to be enabled which represents the beginning of execution of a multimedia object. t is said to be firable if $E(t) = 0$ which represents the end of a multimedia object. $Dm(t)\rightarrow(x, n, y)$, in normal case, n represents the execution duration of time of t. Firing t generates a new marking M': $\forall p\in\bullet t$: $M'(p) = M(p)-1$; $\forall p\in t\bullet$: $M'(p) = M(p)+1$
- $\forall t \in T_s$, $S(t)=f_s$, t is said to be enabled if $f_s|_M = {}_\bullet T_\bullet$, i.e., all input places of t satisfy the logical input expression f_I at M; if t is enabled, it can fire and firing t generates a new marking M': $\forall p\in\bullet t$: $M'(p) = M(p)-1$; $\forall p\in t\bullet$: $M'(p) = M(p)+1$. In some cases, the backtracking algorithm needs to be executed

- $\forall t \in T_I$, t is said to be enabled if the user choose the operation I(t) and meanwhile $\forall p \in \bullet t$, M(p) = 1; if t is enabled, it can fire and generates a new marking M': $\forall p \in \bullet t$: M'(p)=M(p)-1; $\forall p \in t\bullet$: M'(p) = M(p)+1.
- $\forall t \in T_D$, Dt(t) = t_d, t is said to be enabled if $\forall p \in \bullet t$; M(p) = 1; t is said to be firable if its enabled duration of time is equal to t_d and firing t results in a new marking M': $\forall p \in \bullet t$: M'(p) = M(p)-1; $\forall p \in t\bullet$: M'(p) = M(p)+1

MODELING MULTIMEDIA REQUIREMENTS USING LTIPN

Modeling synchronization strategies: Three basic synchronization strategies (i.e., firing rules) that favor statically or dynamically defined synchronization units (Senac *et al.*, 1996) are proposed:

- A dynamic synchronization strategy called strong-or, driven by the earliest processing
- A dynamic synchronization strategy called weak-and driven by the latest processing
- A static synchronization strategy called master, driven by a selected processing

The three fundamental strategies entail nine firing rules obtained from a consistent and complete combination of the absolute temporal validity interval of the synchronization units associated with an inter-stream synchronization point (Senac *et al.*, 1996).

The LTIPN can simply, explicitly and directly express the three basic synchronization strategies by using logical expressions.

Master: The synchronization strategy called master, can be modeled as Fig. 2. m_1 is the master multimedia object such as audio and m_2 is the secondary multimedia object such as video. When transition m_1 representing the master multimedia object is fired, $p_1 \in t\bullet$ receives a token. The logical variable p_1 = .T. The logical expression $f_s = p_1 \wedge (1 \vee p_2)$ = .T. corresponding to the synchronization transition s. So, the synchronization transition s is fired whether p_2 = .T. or not.

When the synchronization transition s is fired, it can cause two problems. One is that some places before s still have a token after s is fired. The other is that some transitions have not been fired before s after s is fired. These two problems can be solved by a backtracking algorithm (Shan *et al.*, 2000).

Strong-or: The synchronization strategy called strong-or, can be modeled as Fig. 3. When either p_1 or p_2 receives a token, the logical expression $f_s = p_1 \vee p_2$ = .T. corresponding to the synchronization transition s. The synchronization transition s is fired. When the synchronization transition strong-or is fired, it can also cause the above two problems and can be solved by the backtracking algorithm like master.

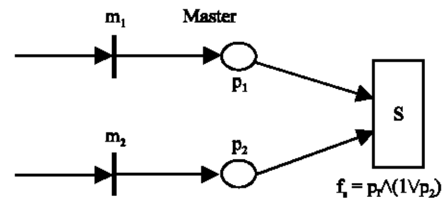

Fig. 2: The synchronization strategy called master

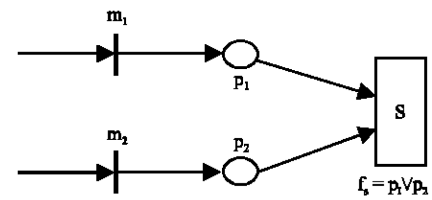

Fig. 3: The synchronization strategy called strong-or

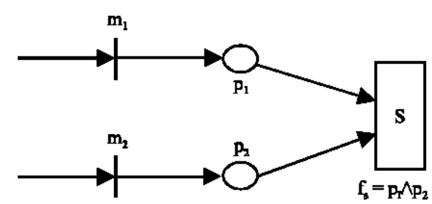

Fig. 4: The synchronization strategy called weak-and

Weak-and: The synchronization strategy called weak-and can be modeled as Fig. 4. When both p_1 and p_2 receive a token, the logical expression $f_s = p_1 \wedge p_2$ = .T. corresponding to the synchronization transition s. The synchronization transition s is fired. The other six synchronization strategies can be expressed by composing the above three basic synchronization strategies.

Modeling basic temporal relations: Given any two multimedia objects specified by temporal intervals, there exists a LTIPN representation for their relationship in time. LTIPN can simply, explicitly and directly describe seven basic temporal relations between two multimedia objects. There are totally thirteen temporal relations between two objects and the other six are inverse relations of the six basic temporal relations. Note that the equality relation has no inverse. The LTIPN representations of the seven basic temporal relations are modeled as Fig. 5a-g.

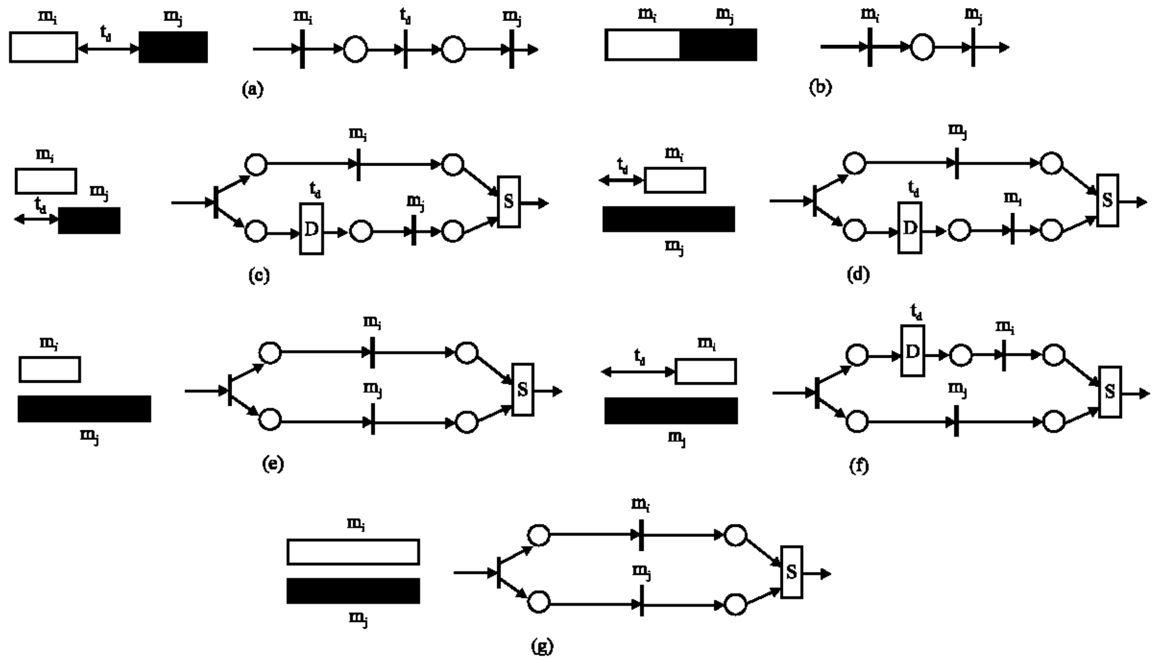

Fig. 5: The LTIPN representations of seven basic temporal relations, (a) m_i before m_j, (b) m_i meets m_j, (c) m_i overlaps m_j, (d) m_i during m_j, (e) m_i starts m_j, (f) m_i finishes m_j and (g) m_i equals m_j

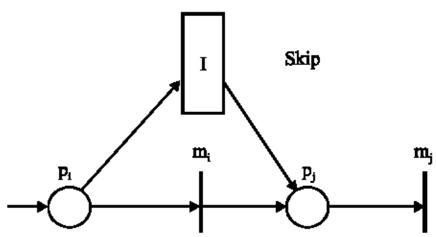

Fig. 6: The LTIPN representation of the skip operation

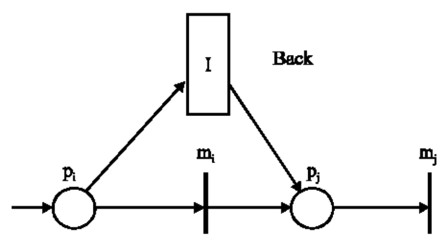

Fig. 7: The LTIPN representation of the back operation

Modeling user interaction: A user should be allowed to manipulate the presentation sequence in multimedia systems either through a program or by key-board/mouse input (Prabhakaran and Raghavan, 1993). The LTIPN can describe some multimedia synchronization with user participation such as skip, back, replay, pause and resume.

Skip: The skip operation can be modeled as Fig. 6. When a user chooses the skip operation and $p_i \in \bullet t$ has a token at the same time, the user interactive transition skip is fired. p_i removes a token and the remaining duration of time e associated with m_i is reset and $p_j \in t\bullet$ receives a token, the transition representing a multimedia object m_j is enabled and m_j is started.

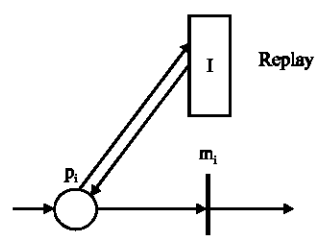

Fig. 8: The LTIPN representation of the replay operation

Back: The back operation can be modeled as Fig. 7. When a user chooses the back operation and $p_j \in \bullet t$ has a token at the same time, the user interactive transition back is fired. p_j removes a token and the remaining duration of time e associated with m_j is reset. $p_i \in t\bullet$ receives a token, the transition representing a multimedia object m_i is enabled and m_i is started.

Replay: The replay operation can be modeled as Fig. 8. When a user chooses the replay operation and $p_i \in \bullet t$ has

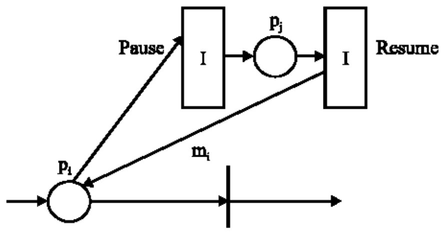

Fig. 9: The LTIPN representation of the pause and resume operation

a token at the same time, the user interactive transition replay is fired. p_i removes a token and the remaining duration of time e associated with m_i is reset. Then $p_i \in t\bullet$ receives a token again, the transition representing a multimedia object m_i is enabled and m_i is restarted.

Pause and resume: The pause and resume operation can be modeled as Fig. 9. When a user chooses the pause operation and $p_i \in \bullet t$ has a token at the same time, the user interactive transition pause is fired. p_i removes a token and the remaining duration of time e associated with m_i is saved. Then, $p_j \in t\bullet$ receives a token.

When a user chooses the resume operation and $p_j \in \bullet t$ has a token at the same time, the user interactive transition resume is fired. $p_i \in t\bullet$ receives a token again. The transition representing a multimedia object m_i is enabled again. However, because the remaining duration of time e associated with m_i is not the initial value and m_i only show the remaining duration of time e.

CONCLUSIONS

A multimedia synchronization model LITPN for describing multimedia synchronization is designed in this study. In this model, we classify transitions into four categories: multimedia object transitions, multimedia synchronization transitions with logical expressions, user interaction transitions and delay transitions. These transitions express multimedia synchronization in a natural and intuitive way.

Further study is to develop an algorithm for building the LTIPN reachability graph to perform some analysis, thus allowing to check the real time properties and the consistency of multimedia systems. Also, we intend to develop a tool for the modeling and analysis of LTIPN models.

ACKNOWLEDGMENTS

This study is supported in part by the National Natural Science Foundation of China under Grants 60773034, 90818023, 90718012 and 60803032; the Taishan Scholar Construction Project of Shandong Province, China; the Scientific and Technological Developing Program of Shandong Province of China under Grant 2008GG30001024 and the Open Project of the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences under Grant SYSKF0804; the Research Project of SUST Spring Bud in Shandong University of Science and Technology in 2009 and the Graduate Innovation Fund of Shandong University of Science and Technology.

REFERENCES

Du, Y.Y., C.J. Jiang and M.C. Zhou, 2007. Modeling and analysis of real-time cooperative systems using petri nets. IEEE Trans. Syst., Man Cybernetics-Part A: Syst. Hum., 37: 643-654.

Du, Y.Y., C.J. Jiang and M.C. Zhou, 2008. A petri nets based correctness analysis of internet stock trading systems. IEEE Trans. Syst. Man Cybern. C Applied Rev., 38: 93-99.

Du, Y.Y. and B.Q. Guo, 2009. Logic petri nets and equivalency. Inform. Technol. J., 8: 95-100.

Little, T.D.C. and A. Ghafoor, 1990. Synchronization and storage models for multimedia objects. IEEE J. Selected Areas Commun., 8: 413-426.

Prabhakaran, B. and S.V. Raghavan, 1993. Synchronization models for multimedia presentation with user participation. Proceedings of the 1st ACM international Conference on Multimedia, 1993, Anaheim, California, pp: 157-166.

Senac, P., M. Diaz and P. De-Saqui-Sannes, 1996. Modeling logical and temporal synchronization in hypermedia systems. IEEE J. Selected Area Commun., 14: 84-103.

Shan, Z.G., Y. Yang, R. Wang and X.Y. Yu, 2000. Extended time stream petri nets: A multimedia synchronization model based on Tspn. J. Comput. Res. Dev., 37: 223-227.

Vicario, E., 2001. Static analysis and dynamic steering of time-dependent systems. IEEE Trans. Software Eng., 27: 728-748.

Woo, M., N.U. Qazi and A. Ghafoor, 1994. A synchronization framework for communication of pre-orchestrated multimedia information. IEEE Network, 8: 52-61.