

# Temporal Verification of Business Processes Using BPMN and CPN

Ryoya Shiraki \*   Yoshiyuki Shinkawa \*

## Abstract

Because of recent advances in information technologies, along with incessant changes in economic and political environments, today's business operations require more speed than ever before. In order to achieve the required speed in business, we have to describe and analyze the business accurately from temporal aspects, especially for business processes. This paper proposes a modeling and simulation based verification method for temporal aspects of business processes, integrating two contrastive modeling tools. One is the Business Process Modeling Notation (BPMN) which provides us with high capability for business process description, and the other is the Colored Petri Net (CPN) which provides us with high capability for rigorous system verification.

*Keywords:* Business Process Modeling, BPMN, Colored Petri Nets, Time Constraints.

## 1 Introduction

Recent advances in information and other technologies make the speed in business more important than ever before. For example, IoT (Internet of Things) makes the supply chain and logistics more efficient, and the relevant business operations require more speed. In another example, new transportation technologies like drones also affect the speed of business. In addition, in order to react properly to incessant and swift change in economic and political environments, rapid business transformation becomes imperative. For this speed, business processes (or BPs for short) are one of the most important critical success factors.

Many kinds of business process modeling tools have been proposed to describe, analyze, and evaluate business processes [1]. Among them, the BPMN (Business Process Modeling Notation) is one of the most widely used and standardized modeling tools, which can rigorously describe complicated task structure, organizations to perform the tasks, and resources to be managed in business processes.

In addition, by incorporating time constraints into them, we can reveal the requirements for the above speed in business. Although it has high descriptive capability using abundant elements, the resultant models become too complicated to be verified. Therefore, we need more analytic tools to be used complementarily with the BPMN. These tools must be

---

\* Graduate School of Science and Technology, Ryukoku University, Shiga, Japan

equipped with the capability to simulate and evaluate the temporal aspects of the business processes.

In this paper, we propose a method to verify the validity of BPMN models with time constraints through simulation, by converting them into Colored Petri Net (CPN) models. The paper is organized as follows. In section 2, we briefly discuss the temporal aspects in business processes, in conjunction with their expressions using BPMN. Section 3 introduces Colored Petri Net (CPN) including temporal functionality. Section 4 shows a model transformation method from BPMN to CPN. In section 5, we discuss how the transformed CPN models are verified, and finally in section 6, overall conclusions along with the future work are presented.

## **2 Temporal Aspects of Business Processes and Their Expression by BPMN**

In this section, we discuss the temporal aspects of business processes, along with their notation using BPMN.

### **2.1 Business Processes with Time Constraints**

Some elements and events in business processes are associated with time constraints or conditions, which include allowable task execution time, resource delivery time, and process initiation/termination deadlines. These constraints and conditions are recognized as the temporal aspects of a business process as a whole. Before discussing these temporal aspects, we briefly summarize a business process.

A business process represents a series of linked tasks, which collectively describe the procedure of how a business goal is achieved [2]. These tasks are performed serially or concurrently, following predefined business rules, legal regulations, resource availability, and other constraints.

Recently, time constraints and conditions become more crucial to make the business competitive. Therefore, these constraints have to be included in the business processes. The basic time constraints that exist in a business process are as follows.

1. The deadline of a task or process
2. The execution time of a task or process
3. Start of task or process at specified time
4. Task and process duration
5. Periodic execution of tasks and processes

In addition, there are complicated combinations of them, however they are composed of two basic constraints, that is, a specific point in time and a duration. These constraints are divided into two groups, namely requirements-based and fact-based, which are associated with the tasks and processes that appear in the business process model. In order to describe the model accurately, it is necessary to use appropriate notation and language. In this paper, we use BPMN for model description.

## 2.2 Business Process and BPMN

BPMN is one of the most widely used business process modeling tools because of its ease of use, intuitive understandability, and visualization capabilities. It provides us with many kinds of model elements to express business participants, activities, events, data, artifacts and control structure in the form of symbols. BPMN was announced in 2004 by industry association BPMI (Business Process Management Initiative). After BPMI is merged with OMG (Object Management Group), an industry organization that promotes the standardization of object-oriented technology, this organization is in charge of BPMN maintenance, and has standardized BPMN specifications in 2006 [3][4]. Figure 1 shows the basic element of BPMN.

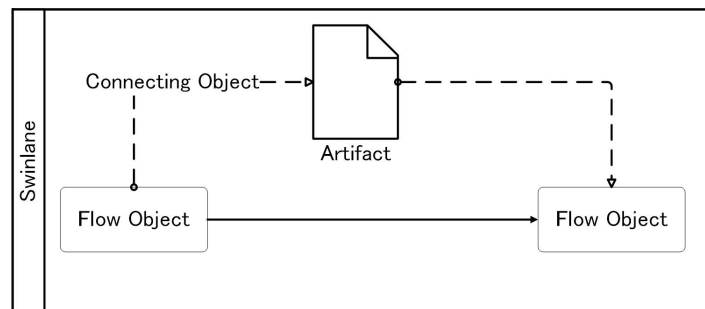


Figure 1: Basic elements of BPMN

As for the time constraints, BPMN uses “timer events” to specify time constraints in a business process. With these timer events, the following specifications become possible.

1. Delay before starting activity
2. End of activity deadline
3. Process deadline
4. Activities started after the specified period of time
5. Process start at specified time or at regular intervals

Figure 2 shows an example of a model using BPMN timer event. In this figure, the task “Sell Items” is active from 10 o’clock to 18 o’clock, and if all the items are sold out before 18 o’clock, “Post Sold Out” task is activated, while if some items remain unsold until 18 o’clock, the whole process terminates.

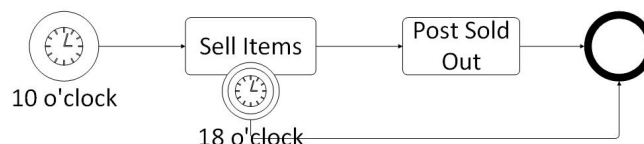


Figure 2: An example of the timer event

### 3 Colored Petri Net

A Petri net is a directed bipartite graph with two kinds of nodes “place” and “transition”, which represent a condition and an event respectively[3]. These nodes are connected by “arcs” in order to express the relationships between the conditions and events. The behavior of a Petri net is determined by the movement of “tokens” marked in the places. Figure 3 is the basic element of a Petri Net.

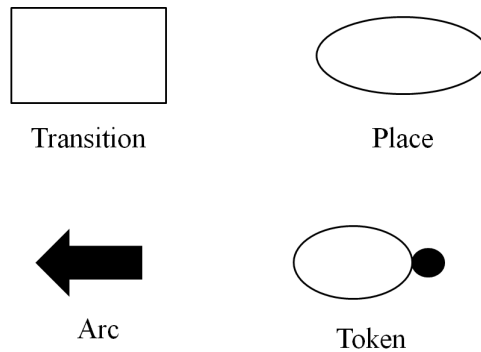


Figure 3: Basic elements of a Petri Net

Each transition becomes eligible to fire when all the input places contain the designated number of tokens. This simply structured Petri net is often referred to as a regular Petri net. While a regular Petri net can express the behavior of a system precisely, it seems difficult to represent the functionality of a system using it. In order to express the functional aspects of a system, Colored Petri Net (CPN) is introduced to make it possible to

1. assign a data type to each token which is referred to as a *color*
2. set the values to a token according to the color definition
3. assign a function to an arc to manipulate the values that are set to tokens
4. assign a function to a transition to control its firing, which is referred to as a *guard*
5. define variables for representing the tokens moving within a CPN model

Formally, the CPN is defined by the following nine items [5].

$$CPN = (P, T, A, \Sigma, V, C, G, E, I)$$

- P: A finite set of places

A place represents a storage location of data and information. Storing them in a place is called token marking.

- T: A finite set of transitions

A transition represents an event, and an event occurrence is referred to as transition firing.

- A: finite set of arcs

An arc represents the movement of tokens between places. In CPN, it can be associated with a function to transform the token data.

- $\Sigma$ : A finite set of non-empty color sets  
A color set represents a data type, and is used to assign an attribute to each token.
- $V$ : A finite set of typed variables.  
A typed variable is used to represent a token to be processed by a function written using CPN / ML.
- $C$ : Color function  
A color function associates a color with a place.
- $G$ : A set of guard functions  
A guard function indicates an ignition condition of a transition.
- $E$ : An arc expression function  $A \rightarrow \text{expression}$   
Each expression represents a token conversion rule due to transition firing, and is referred to as an arc function.
- $I$ : An initialization function :  $P \rightarrow \text{closed expression}$   
It defines the token marking of each place at the start point.

While the CPN is an extension of the original Petri net to express the functionality of a system, there is another extension for the temporal properties. This extension is referred to as “timed Petri nets” [6]. The CPN can be extended to the *timed* CPN by appending a *timed* property to each token through the color definition. This property gives a *clock timer* to a token, which can postpone the transaction firing until the timer expires. The clock values can be set or reset by arc functions or transaction firing. This simple mechanism makes it possible to express, analyze, and evaluate complicated systems including various temporal events. This firing delay is interpreted as one of the followings.

1. The corresponding event occurs after a specified time
2. The corresponding event lasts for a specified time
3. The token stays in the input place for a specified time

## 4 Model Transformation from BPMN to CPN

In order to evaluate the time constraints in BP using timed CPN, we first have to transform the BPMN models with time constraints into timed CPN models. The basic conversion rules for BPMN model elements are as follows.

1. Start and end events are transformed to CPN places
2. An activity is expressed as a CPN place
3. A connection object is expressed as an CPN arc
4. A decision at a gateway node is implemented as a guard in CPN

5. A time constraint is implemented as a combination of timed tokens and CPN /ML functions in CPN

Figure 4 shows an example of conversion from a BPMN model to a CPN model.

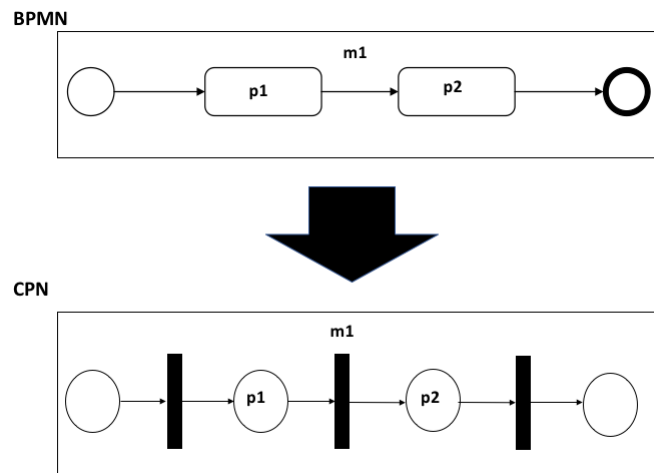


Figure 4: An example of conversion from BPMN diagram to CPN diagram

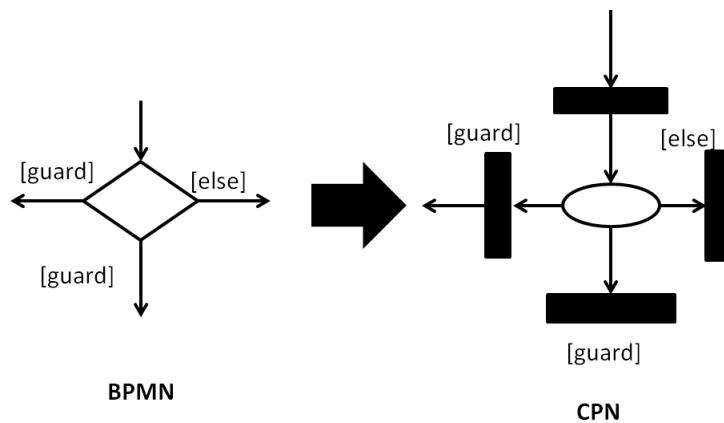


Figure 5: An example of select point conversion

In addition, in the section of a gateway in BPMN, the selection point evaluates the guard conditions one by one, and a firstly satisfied guard is used to select the path [7]. If the guards are specified inappropriately, no path could be selected and the process stops. In order to avoid this, the “else” condition is appended to the selection point, and transformed into CPN as shown in Figure 5.

## 5 Verification of Business Processes with Timed CPN

Once BPMN models are transformed into the equivalent CPN models, the next step is to verify the transformed models whether they satisfy the given temporal requirements reside in the original models. Since CPN is a simulation based tool, rather than analytical one, we need to set up the models for simulation.

In order to make the CPN models simulatable, which are converted from BPMN models including time constrains, we have to set an appropriate initial token marking. In this marking, we have to define the number of tokens and the places to which the tokens are marked, attribute values of each color token, and clock values assigned to the timed tokens[8]. In addition, we need to identify each delay time, its mean time and distribution, which occurs in the business process. These delay times with their distribution are processed in CPN / ML codes in order to reflect the time fluctuation as close to the real world as possible. After all the setup is completed, the CPN model can be simulated using cpn tools [9].

The proposed approach was evaluated using a simplified web-based travel reservation business process. This business process is composed of three major tasks that are performed randomly with their unique time constraints. The BPMN notation is shown in Figure 6, and the transformed CPN notation following our approach is shown in Figure 7. A part of CPN ML codes to define the color sets, guard functions, and arc functions, is shown in Figure 8. In Figure 6, eight time constraints are embedded in the BPMN model, which are marked by the timer elements (shown as the clock symbols). They are implemented as arc functions using the “+@” operator for the delay, and the “discrete” function for the fluctuation.

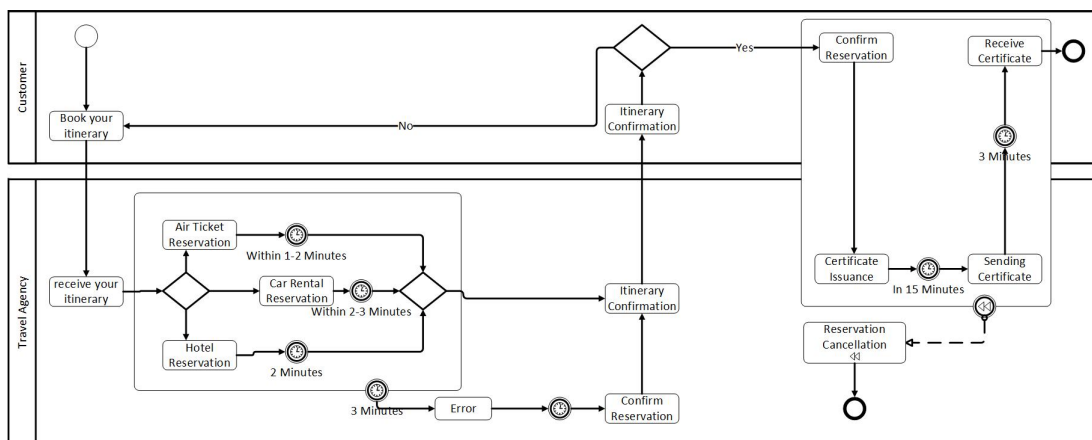


Figure 6: Example of BPMN

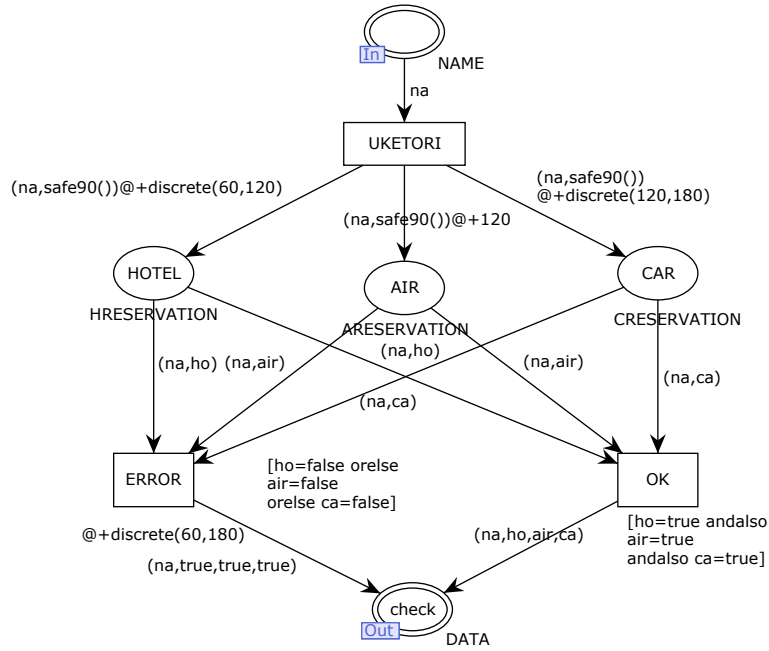


Figure 7: Example of conversion to a timed-CPN

```

colset NAME =string timed;
colset HOTEL =bool timed;
colset HRESERVATION=product NAME*HOTEL timed;
var na:NAME;
var hre:HRESERVATIO
fun safe90 ()=if (discrete (1,100) > 5) then true else false;

```

Figure 8: An example of CPN/ML

## 6 Conclusions and Future Work

Temporal evaluation of business processes becomes more and more crucial than ever before, because of recent advances in information and other technologies. In this paper, we propose a CPN based evaluation method for time constraints in business processes. The method assumes that all the business processes with time constraints are provided in the form of BPMN models. By using BPMN, we can express the business processes in a visually understandable way. These models are to be transformed into CPN models, so that we can express and simulate the models rigorously for the evaluation of time constraints.

Currently only task-related delay is taken into account for the temporal aspects of business processes. However, other kinds of delays, e.g. resource related, business rule related, and legal related delays also affect the temporal aspects. Therefore, these delays have to be included in business process models. In addition, possible waiting lines at each task should also be taken into account. Our future work focuses on these points, using the revised CPN model structure and the queuing theory. Another future work includes an automated model transformation process, since the model transformation from BPMN to CPN follows a set



of strict rules, and takes long time to complete. This automated transformation would be possible by exporting the BPMN models into XML documents, and converting them into the Prtri Net Markup Language (PMNL – ISO/IEC 15909-1:2004), which is expressed in the form of XML.

## References

- [1] M. Havey, *Essential Business Process Modeling*, Oreilly & Associates Inc, 2005.
- [2] M. Dumas, M. La Rosa, J. Mendling, and H.A. Reijers, *Fundamentals of Business Process Management*, Springer, 2013
- [3] Object Management Group, *Business Process Model and Notation (BPMN)*, 20th ed.: Object Management Group, 2011.
- [4] B. Silver, *BPMN Method and Style, 2nd Edition, with BPMN Implementer's Guide: A Structured Approach for Business Process Modeling and Implementation Using BPMN 2*, Cody-Cassidy Press, 2011
- [5] K. Jensen and L.M. Kristensen, *Coloured Petri Nets : Modelling and Validation of Concurrent Systems*:Springer-Verlag New York Inc, 2009.
- [6] J. Wang, *Timed Petri Nets: Theory and Application (The International Series on Discrete Event Dynamic Systems)*, Springer, Heidelberg, Germany, 1998.
- [7] M. Ramadan, H.G. Elmongui, and R. Hassan.,“BPMN Formalisation using Coloured Petri Nets”,*Proceedings of the 2nd GSTF Annual International Conference on Software Engineering Applications (SEA2011)*, 2011
- [8] W.M.P. van der Aalst and C. Stahl, *Modeling Business Processes: A Petri Net-Oriented Approach (Information Systems)*, The MIT Press, 2011.
- [9] K.Jensen, L. Kristensen, and L. Wells “Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems,” *International Journal on Software Tools for Technology Transfer (STTT)*, Vol. 9, Numbers 3–4, 2007, pp. 213–254