Position papers of the Federated Conference on Computer Science and Information Systems, pp. 161–168
DOI: 10.15439/2017F539
ISSN 2300-5963 ACSIS, Vol. 12

Temporal Evaluation of Business Processes Using Timed Colored Petri Nets

Yoshiyuki Shinkawa
Graduate School of Science and Technology
Ryukoku University
1-5 Seta Oe-cho Yokotani, Otsu, Shiga, Japan
Email : shinkawa@rins.ryukoku.ac.jp

Ryoya Shiraki
Graduate School of Science and Technology
Ryukoku University
1-5 Seta Oe-cho Yokotani, Otsu, Shiga, Japan
Email : t16m081@mail.ryukoku.ac.jp

***Abstract*—Time constraints become one of the most crucial requirements to be satisfied in business processes, in order to make the business competitive, and to achieve the business goal. This paper proposes a Colored Petri Net (CPN) based approach to modeling and evaluating business processes including various temporal constraints and requirements. The essential part of a business process and temporal aspects are separated in our approach to make the business process models comprehensive. More specifically, separately-defined modules to deal with time constraints are to be appended to the traditional non-temporal business process models. The created models can be simulated using "cpntools" to evaluate whether the given temporal requirements are satisfied.**

I. Introduction

THE evolution of information and communication technologies, along with the growth of the Internet, makes time constraints in business more important than what they were in the pre-Internet ages. This evolution also makes business processes [1] more complicated, since there are many alternatives available regarding information storage and retrieval, process automation, exception handling, and so on. In these circumstances, we need to include various information into business process models, such as data processing options, current and future availability for related resources, complicated decision rules to optimize the business, business and legal rules that regulate the process, and time constraints with statistical and probabilistic properties [2].

However, traditional business process modeling and specification languages like BPMN (Business Process Modeling Notation) [3] and BPEL (Business Process Execution Language) [4] mainly focus on task flows and their relationships within a business process. Even though several trials have been made to extend these languages from the temporal viewpoint, they do not provide us with enough capability to express the above information yet [5][6]. In order to express today's complicated business processes, we need a modeling tool which can reflect the following aspects of a business process.

1) Requirements and availability of material, human and financial resources
2) Business and legal rules regulating each task or the whole process
3) Task and process related data, along with their transformation rules
4) Temporal requirements to the process, such as response time and throughput
5) Temporal properties of each task and resource, such as mean execution time and mean resource retention time along with the variance.

Consequently, a modeling tool for business processes must provide us with the capability to depict the four aspects of a business process, namely, the structural, functional, behavioral, and temporal aspects. In addition, logical and probabilistic properties must be precisely expressed in the business process models.

Colored Petri Net (CPN) [7] is one of the comprehensive modeling tools satisfying the above requirements, since it can depict the structure of a system as a directed bipartite graph, the complicated regulation and transformation rules that reside in a system as functions written by CPN ML language, the data structure as color sets, the temporal requirements and facts as timed color sets, and probabilistic and statistical properties using the library functions provided as a part of CPN ML language.

This paper proposes a CPN based approach to modeling and evaluating time constraints in business processes. The paper is organized as follows. In section 2, we discuss various time constraints that reside in business processes, in conjunction with their probabilistic and statistical properties. Section 3 introduces Colored Petri Net (CPN) along with its temporal expression capability. Section 4 presents how complicated business processes including time constraints are modeled using CPN. Section 5 shows an evaluation of time constraints in business processes expressed in the form of CPN.

II. Time Constraints in Business Processes

In this section, we introduce possible time constraints that reside in business processes. Before discussing it, we first define the basic business process structure briefly.

A. The Essential Structure of a Business Process

There could be many viewpoints to a business process depending on the purpose of business process modeling, which include behavioral, functional, structural, financial, and temporal ones. In addition each business process modeling tool provides us with its unique model elements from its own

perspective to business. Therefore we need to build multiple business process models using multiple modeling tools reflecting the above multiple viewpoints. However, in order to define time constraints in a business process rigorously, it is desirable to build a unified model which reflects all the essential elements among those multiple models.

Generally speaking, the simplest form of a business process is a set of task flows, where a task represents an indecomposable unit of work that configures the business process. Upon this simplest model, the following information should be added to make its semantics clear.

1) The organizations and human resources that engage in the business process
2) The materials to be dealt with by the business process
3) The financial resources which are required to execute the business process
4) The business and legal rules to regulate the business process
5) The goals and strategies behind the business process

The representation and notation of the above information are shown in section 4.

B. Time Constraints in Business Processes

There are two contrastive aspects of time constraints in business processes. One is the time-related requirements to a business process or a task, which must be satisfied by the implemented business process. The other is the time-related properties of each task or resource, which restrict the behavior of them. The former are given in the form of *points in time* or *durations*. For example, the requirements such as "the task must start at 9:00AM" and "the whole process must end by 6:00PM" designate specific points in time, and are often referred to as *deadlines*, while "the task must end within one our" and "the whole process must end within three days" designate durations.

On the other hand, the latter are not usually given as specific values, instead we have to monitor and measure each task and resource in order to obtain the values. Unlike the physical events, business events are unstable and therefore most of the measured values are statistical. As a result, each value is given as a set of a mean value and variance with an appropriate distribution function. For example, the value is given in the form of "the mean execution time of the task is 5 minutes, the variance is 4, and follows the normal distribution", or "the mean term of validity for the resource is 5 days, the variance is 2, and follows the gamma distribution". In some cases, these values are given as fixed values without measurement, which include the values given by material specifications, business rules like employment regulation, or legal rules. For example, the expiration date of a material, the retirement date of an employee, and the warranty period of a product are the temporal properties given as fixed values.

The above temporal properties usually cause the delays in a business process, mainly at task initiation or during task execution. The delay mechanism is not so simple since there are several factors to cause the delays, and we have to take the combination of them into account. The first factor is the task execution time that can fluctuate following a distribution function that is associated with the task. The second is the resource waiting time, during which the related tasks are halted until the resources become available. These resources could be material, human, or financial ones. The third is the service waiting time, during which some serving mechanism is busy even though all the related resources are available. In this situation, tasks are waiting in a queue in terms of the *queuing theory* [8]. The last is the rule based waiting time, during which the tasks are halted because of some business or legal rules. This case includes financial audit, facility inspection, and complaint handling which might stop the task execution.

As discussed above, time constraints in a business process are so complicated that we cannot handle them using traditional task flow based business process models. We need more information-rich models which reflect not only the behavioral aspect of a business process, but also the functional, data, and temporal aspects along with various kinds of rules that regulate the business process. In this paper, we use Colored Petri Net (CPN) for this purpose. In the next section, we give a brief description of this technique.

III. Colored Petri Net and Time Constraints

The original Petri net, often referred to as a regular Petri net or a place-transition (PT) net, is a directed bipartite graph with two kinds of nodes called "*transition*" and "*place*", which are alternately connected by directed arcs [9] . In a model written by CPN, or a CPN model for short, a transition represents an event which could occur in a system, while its preceding places represent the pre-conditions for its occurrence, and its succeeding places represent its post-conditions. Each preceding place can be marked by *tokens* to represent the corresponding condition holds. If all the preceding places are marked by tokens, the transition becomes eligible to fire. The transition firing represents an occurrence of the event, and the tokens in the preceding places are transferred to the succeeding places.

Even though the regular Petri net can express the behavior of distributed concurrent systems rigorously, there are several aspects of a system difficult to be represented by it, such as functional (or data transformational) , temporal, probabilistic, statistical, and logical aspects. In order to relieve these difficulties, there have been many extensions to it proposed. Colored Petri net (CPN) is one of these extensions which makes 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

CPN is formally defined as a nine-tuple $CPN=(P, T, A, \Sigma, V, C, G, E, I)$, where

P : a finite set of places.
T : a finite set of transitions.
(a transition represents an event)
A : a finite set of arcs $P \cap T = P \cap A = T \cap A = \emptyset$.
Σ : a finite set of non-empty color sets.
(a color represents a data type)
V : a finite set of typed variables.
C : a color function $P \rightarrow \Sigma$.
G : a guard function $T \rightarrow$ expression.
(a guard controls the execution of a transition)
E : an arc expression function $A \rightarrow$ expression.
I : an initialization function : $P \rightarrow$ closed expression.

While the CPN is an extension of the original Petri net to express the functionality of a system, there is another extension to express the temporal properties. This extension is referred to as "timed Petri nets" [10]. 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. In the next section, we discuss how the business processes with time constraints are modeled using timed CPN.

IV. Modeling a Business Process by CPN

As discussed in section 2, business processes and time constraints are complicatedly interrelated and therefore the business process models including time constraints also become complicated. In order to relieve this model complexity, we build a model by combining three independent CPN models or modules, namely, the base module to represent the essential part of the business process without the time constraints, the delay module to represent the task and resource delays, and the queue module to represent the service waiting time in terms of the queuing theory.

A. The Base Module

This module represents a business process without time constraints, which shows its non-temporal logical aspects. There have been several researches to apply CPN to business processes [11][12][13]. Since the most essential part of a business process is a set of task flows, we focus on a task as a basic element to be modeled by CPN. As discussed previously, a task is an atomic unit of work in a business process, and four kinds of resources are required to perform it, namely, human, material, financial, and information (or data) ones. In CPN model, these resources are represented as *tokens*, and the kinds of tokens are distinguished by *colors*. Each color must include the following information as properties.

[Human Resource Color]
This color represents a person or personnel involved in the business process. The basic information that the color should include is the sort of occupation, own department, post and rank, and salary level.

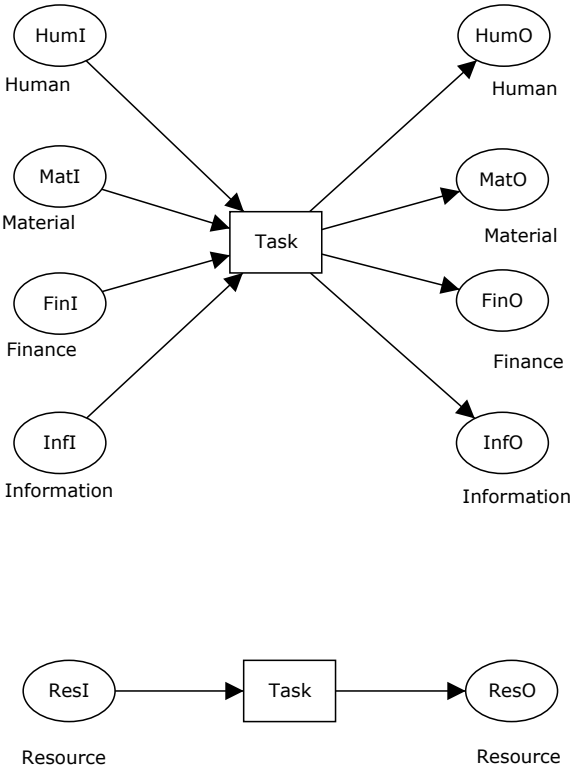

Fig. 1. Basic Task Unit

[Material Resource Color]
This color represents a material to be dealt with by each task. The properties that the color should include are the type of the material, price, quantity, and supplier.

[Financial Resource Color]
This resource represents the required operating capital to perform a task. The properties that the color should includes are the purpose, funding source, amount, and responsible department.

[Information Resource Color]
This color represents the information that is needed to perform a task. The information is provided mainly in the form of data, however in some cases, provided by documents, phones, or faxes. The color should include the type of contents, media of the information, source of the information, and the type of operations to be applied.

As the first step, we define each of the above properties as an integer type color, and the meaning of each value that a color takes is interpreted in CPN ML functions to be defined for arc and guard functions. These values could be unique for each business process. For example, the color set definition for the human resource in some business process is

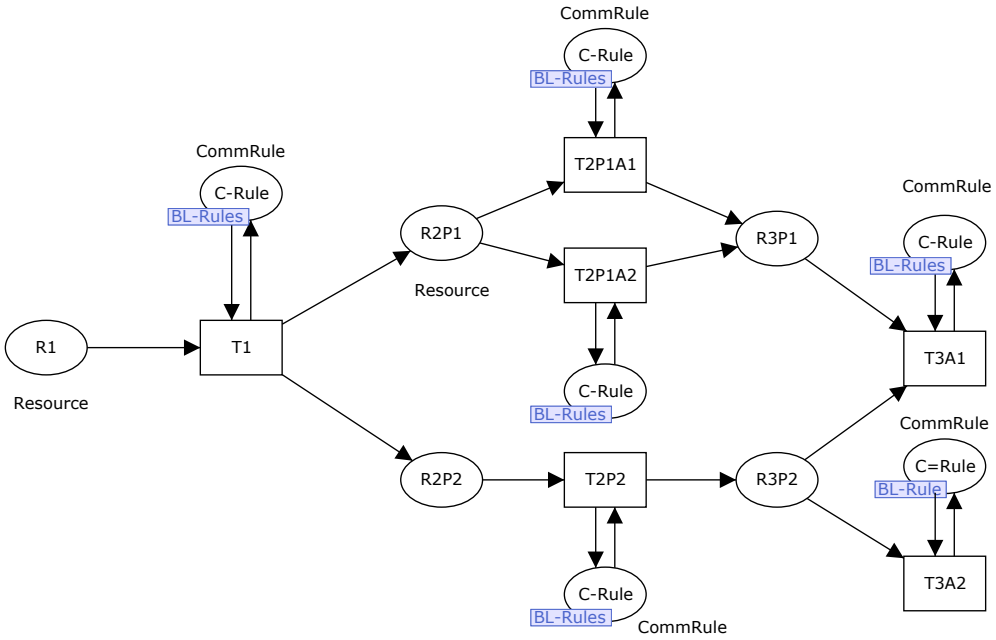

Fig. 2. Business Process Base Module

closet Human = product Occup * Dep * Pos * Rank * Salary;

where each element of the direct product is defined as "INT" type, and the meaning of each value is interpreted in the model, e.g. the value "1" of the color "Occup" means an *engineer*, value "2" means a *sales representative*, and so on.

Since each task manipulates the above four types of color sets, and transforms them as new tokens to be passed to the succeeding tasks, its structure in a CPN model is depicted as shown in the upper figure of Fig. 1. Even though it seems natural to assign the above "resource" colors to the tokens, it could increase the number of tokens, and make the guard and arc function complicated. In order to reduce the number of tokens, and make the functions simple, we use the list of tokens like

closet HumanList = list Human;

In addition, we can reduce the number of task input places by defining the tuple of these lists like

closet Resouce = product HumanList * MaterialList * FinancialList * InformationList;

This closet makes it possible to integrate the four input and output places into a pair of single places as shown in the lower figure of Fig. 1, and this CPN structure is to be used as a basic task unit in this paper. Once the basic unit of business process models is defined, the next step is to combine them into a whole process model that represents all the possible task flows. In order to make this model executable, we have to define guard functions and arc functions appropriately so that they reflect the business, legal and other kinds of rules regulating the business process. These rules are divided into two categories, namely, task unique (or local) rules and business process wide (or global) rules. Each task unique rule controls the transition firing using the information in the input token, and determines the information to be passed to the succeeding tasks through the output tokens. On the other hand, each business process wide rule affects possibly all the task executions using global common rules. These common rules, which include business and legal rules along with social and commercial practices, are represented in the form of CPN ML list, each element of which corresponds to a rule expressed as an integer list. Each guard or arc function is to be responsible for the interpretation and implementation of this integer list. As a result, only one token with the color defined as a list of integer list represents all the global rules, and consequently only one place is used to hold the global rules. This implementation makes the CPN structure simple, since only one arc is needed to access the global rules. Fig. 2 shows a simple implementation of a business process base module expressed in the form of CPN. We use the following naming convention to make the models readable.

1) A transition representing a task is labeled "$\mathrm{T}_i\mathrm{P}_j\mathrm{A}_k$", where "$\mathrm{T}_i$" represents the i-th task, "P_j" represents the task is to be performed in parallel as the jth occurrence, and A_k represents the k-th alternative in a selective task execution.
2) A place representing a resource is labeled "$\mathrm{R}_i\mathrm{P}_j$", which represents the resource is used for the task "$\mathrm{T}_i\mathrm{P}_j\mathrm{A}_k$". Since only one place is needed for aselective task execution, no "A_k" is used.

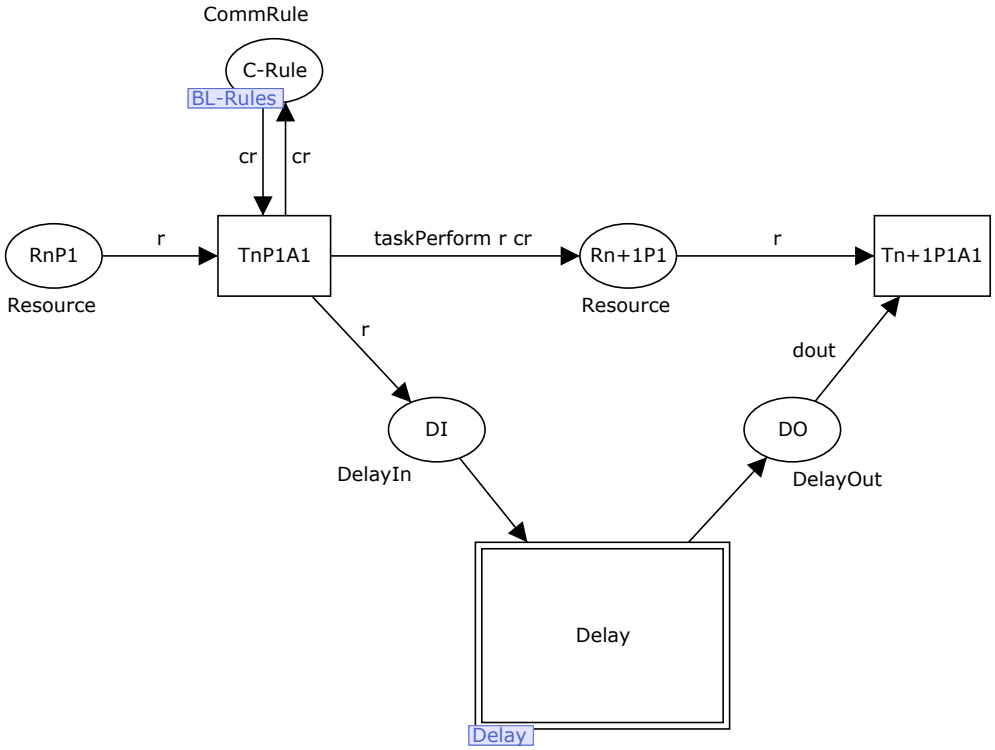

Fig. 3. Independent Delay Module

3) The place for the common rules is labeled "$\mathrm{C-Rule}$" with the *fusion* tag "$\mathrm{BL-Rule}$". A *fusion* place in CPN represents that the different places in a model designate the same place.

By using the above fusion place, we can avoid to use long arcs to access the common rules.

B. Delay Module

The purpose of this module is to implement temporal delays caused by task execution, resource waiting, or business or legal rules. A natural implementation of the delays in a CPN model is to increment a timestamp of each related token at a transition firing or by an arc function to postpone the next transition firing. However, this approach makes the CPN model complicated, since the temporal information spreads over the model. In addition, this information divergence makes the model maintainability worse. In order to avoid this difficulty, we concentrate the temporal information in an independent module connected to the transitions that the delays are expected. Fig. 3 briefly shows this approach. In this figure, the independent module "Delay" is inserted between the task transition "$\mathrm{T}_n\mathrm{P}_1\mathrm{A}_1$" and "$\mathrm{T}_{n+1}\mathrm{P}_1\mathrm{A}_1$" through the two places "DI" and "DO". This module with these places can be inserted between any two consecutive task transitions, and it prevents the divergence of the delay mechanism over the model. Fig. 3 shows the interrelation between the base module and delay module, and composed in the following way.

1) Prepare two additional places "DI" and "DO" to the original CPN model, where "DI" is used as the input to the delay module, while "DO" is the output place from it.
2) Draw an arc from the transition that delays are expected to "DI". This transition is referred to as "deferred transition".
3) Assign a variable (the variable "*r*" in Fig. 3) representing the resource for the delay transition to this arc.
4) Draw an arc from the delay module to the above new output place "DO". The color associated with this new place is an integer with a timestamp. A token with this color is referred to as "delayed token", and the integer value represents the id of a task.
5) Draw an arc from this new output place to the succeeding transitions from the delayed transition.

The firing of the succeeding transition, which represents the next task execution, is postponed arbitrarily by setting the timestamp of the "delayed token". Since the delay of each task transition is determined by the resource status and the global rules, one of the simplest implementations of the "delay module" is as shown in Fig. 4. In this figure, the transition "Analyze" is responsible to examine the resource status to compose the parameter to be passed to the delay calculation function "genDelay". On the other hand, the transition "DCalc" determine the actual delay time using the above parameter and the global rules.

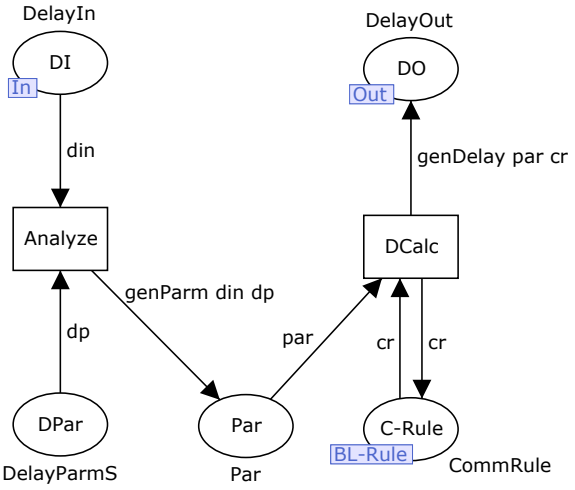

Fig. 4. An Implementation of Delay Module

C. Queue Module

In addition to the above explicitly defined delays, there is a different type of delays called a "wait". The concept of the wait is introduced in the queuing theory [8], which represents the delay caused by server unavailability. In business processes, this server corresponds to one of the person, organization, or facility, which is responsible for a task. In order to make the model simple, we build an independent module that realizes this type of delay, in the similar way we did for the delay module. This module is shown in Fig. 5, and composed as follows.

1) Add the "Queue" module with two additional places "Count" and "Arrival" to the original CPN model including the "Delay" module.
2) Draw a bi-directional arc between the input place "DI" and the queue module.
3) Draw a bi-directional arc between the "Count" place and the queue module, and between the "Count" place and the succeeding transition $\mathrm{T}_{n+1}\mathrm{P}_1\mathrm{A}_1$ in Fig. 5.
4) Draw a uni-directional arc from the queue module to the "Arrival" place.
5) Draw an uni-directional arc from the "Arrival" place to the delay module.

The queue module works as follows.

1) The place "Count" holds an integer token representing the number of concurrently available servers.
2) When the task "$\mathrm{T}_n\mathrm{P}_1\mathrm{A}_1$" ends, a token is passed to the place "DI" to determine the delay.
3) The guard of the queue module makes the module fireable if the token value in the "Count" place is positive. This guard examines the server availability.
4) The above firing passes a token to the "Arrival" place to make the delay module fireable.

The above mechanism implements the queuing model for a single task execution, and is to be deployed over the CPN model where the temporal delays are expected. The internal structure of the queue module is quite simple as shown in Fig. 6.

V. Evaluation of Time Constraints

The CPN models discussed in the previous section reflect temporal properties and constraints associated with the resources and tasks that occur in the business processes. In order to evaluate the business processes from temporal viewpoints, we need to execute the models using the cpntools [14]. For this simulation, the following data are required and must be set in appropriate tokens.

1) Initial resource status for the business process to be simulated, including human, material, and financial resources. These data are set in the resource list tokens which are marked in the input place of each task transition.
2) Business and legal rules that are to be applied to the business process. These data are set in the rule list token which is marked in the global rule place.
3) The number of concurrent ececutions for each task. These data are set in the tokens which are marked in the "Count" place.

After the simulation, we have to examine the timestamp of each timed token to determine whether the given temporal requirements are satisfied. For this purpose, we need to distinguish each business process instance in the simulation result. However, each token in our CPN model does not have enough information to identify to which business instance it belongs. Therefore, we add a sequence number to all the task-related tokens, which represents the process instance identification. This mechanism is implemented by modifying the CPN models as shown in Fig. 7 and is composed in the following way[1].

1) Add a place (the "PISeq" place in Fig. 7) to control the sequence number, which is marked by a timed INT type token initialized as 1.
2) Each business process initiation task transition, that is, a task transition having no preceding task, obtains the above token to get the sequence number of the process instance to be initiated, and pass it back incrementing the value by 1.
3) Add a new place "Trace" to hold the temporal events such as task execution, initiation, or termination in the form of a list. This list is referred to as a *trace list*.
4) Each process initiation task transition appends the above token including the sequence number to the trace list.
5) Each time a task transition other than the above ones fires, it appends a token in "DO" place to the trace list.
6) After a simulation ends, the *trace list* includes task execution history with timestamps.

By analyzing the above trace list, we can evaluate whether the temporal requirements are satisfied. This analysis can be

[1]In Fig. 7, the "Delay" module and "Queue" module are integrated into a single module "Delay and Queue" in order to make the figure concise.

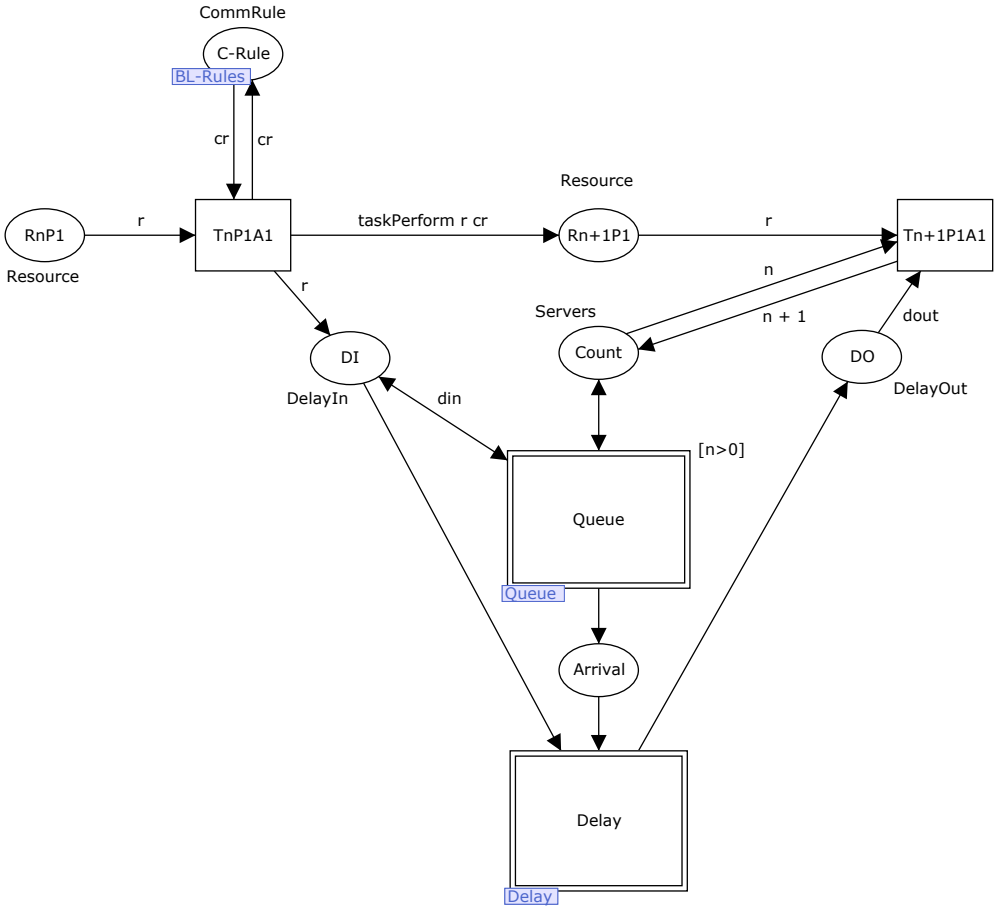

Fig. 5. Independent Queue Module

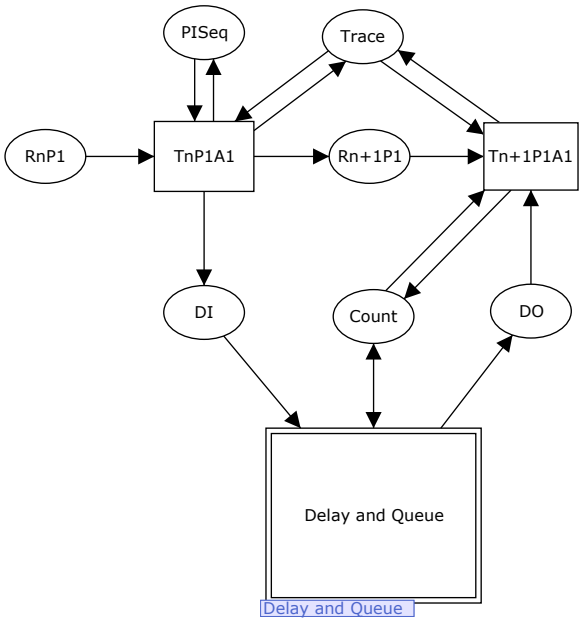

Fig. 7. Evaluation of Time Constraints

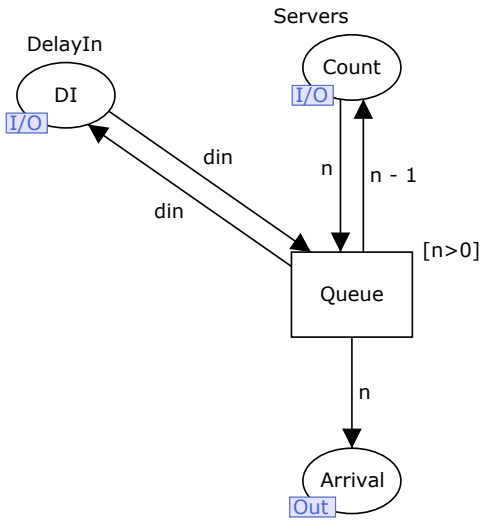

Fig. 6. An Implementation of Queue Module

performed within the model, adding CPN ML codes to the model.

VI. Conclusions

Business processes become more and more complicated for various reasons, e.g. intensified competition, deregulations, new regulations, business and technology innovation, and business globalization. These factors require business process models to deal with temporal properties more precisely. However traditional business process modeling tools are furnished with few capabilities to deal with them.

In this paper, we propose a Color Petri net (CPN) based approach to modeling and evaluating the business processes from temporal viewpoints. We first model the business processes without time constraints, including various business and legal regulations along with resource based constraints. Succeedingly, we added the temporal modules to them with minimal modification of the original models. Finally, we added the evaluation mechanism to the modified models. Our approach needs many CPN ML codes and functions, and currently they must be built for each individual model, which lowers the evaluation efficiency. The next step of this research is to manage and reuse these codes and functions.

References

[1] M. Dumas, M. La Rosa and J. Mendling, *Fundamentals of Business Process Management,* Springer, Heidelberg, Germany; 2013.

[2] S. Cheikhrouhou, S. Kallel, N. Guermouche, and M. Jmaiel, "The temporal perspective in business process modeling: a survey and research challenges," *Service Oriented Computing and Applications* vol. 9, issue. 1, 2015, pp. 75–85. [Online]. Available: http://dx.doi.org/10.1007/s11761-014-0170-x

[3] 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, Altadena, CA; 2011.

[4] M. B. Juric and D. Weerasiri, *WS-BPEL 2.0 Beginner's Guide,* Packt Publishing, Birmingham, UK; 2014.

[5] D. Gagne and A. Trudel, "Time-BPMN," *Proc. of 2009 IEEE Conference on Commerce and Enterprise Computing*, pp. 361–367, 2009. [Online]. Available: http://dx.doi.org/10.1109/CEC.2009.71

[6] H. Banati, P. Bedi and P. Marwaha, "Extending BPEL for WSDL-Temporal based Web services," *Proc. of 12th International Conference on Hybrid Intelligent Systems (HIS)*, pp. 484–489, 2012. [Online]. Available: http://dx.doi.org/10.1109/HIS.2012.6421382

[7] K. Jensen and L. Kristensen, *Coloured Petri Nets: Modelling and Validation of Concurrent Systems,* Springer, Heidelberg, Germany; 2009.

[8] V. V. Kalashnikov, *Mathematical Methods in Queuing Theory (Mathematics and Its Applications),* Springer, Heidelberg, Germany; 2010.

[9] M. Diaz, *Petri Nets: Fundamental Models, Verification and Applications,* Wiley-ISTE, Hoboken, NJ; 2009.

[10] J. Wang, *Timed Petri Nets: Theory and Application (The International Series on Discrete Event Dynamic Systems),* Springer, Heidelberg, Germany; 1998.

[11] M. Werner, "Colored Petri Nets for Integrating the Data Perspective in Process Audits," *Lecture Notes in Computer Science* vol. 8217, 2013, pp. 387–394. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-41924-9_31

[12] K. van Hee, O. Oanea, and N. Sidorova, "Colored Petri Nets to Verify Extended Event-Driven Process Chains," *in the 2005 Confederated International Conference on On the Move to Meaningful Internet Systems,* Agia Napa, Cyprus, 2005, pp. 183–201. [Online]. Available: http://dx.doi.org/10.1007/11575771_14

[13] W. P. M. Aalst and C. Stahl, *Modeling Business Processes: A Petri Net-Oriented Approach,* The MIT Press, Heidelberg, Cambridge, MA; 2011.

[14] 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. [Online]. Available: http://dx.doi.org/10.1007/s10009-007-0038-x