

A NOVEL MODELING METHODOLOGY FOR COLLABORATIVE ENTERPRISE PROCESSES

KWANGYEOL RYU¹, SUNHWA LEE¹, DUCKYOUNG KIM² AND YOUNG-JUN SON³

¹Department of Industrial Engineering
Pusan National University
San30, Jangjeon-dong, Geumjeong-gu, Busan 609-735, South Korea
{ kyryu; sh5362 }@pusan.ac.kr

²School of Design and Human Engineering
Ulsan National Institute of Science and Technology
UNIST-gil 50, Eonyang-eup, Uljin-gun, Ulsan 689-798, South Korea
dykim@unist.ac.kr

³Department of Systems and Industrial Engineering
The University of Arizona
1127 E. James E. Rogers Way, P. O. Box 210020, Tucson, AZ 85721-0020, USA
son@sie.arizona.edu

Received March 2011; revised July 2011

ABSTRACT. *This paper proposes an extended collaborative process modeling (exCPM), underpinned by the concepts of tokens in Petri Nets (PNs) and ICOM in IDEF0. exCPM allows us to represent collaborative processes in a more comprehensive manner than CPM. In addition, it provides a systematic model verification procedure by model transformation into PNs. In this paper, we first analyze pros and cons of currently available process modeling methods. We then discuss details of the proposed exCPM, and demonstrate its modeling and verification capabilities with case studies.*

Keywords: Collaboration, Process modeling, Model verification, Collaborative process modeling (CPM), Petri-nets (PNs)

1. **Introduction.** In manufacturing industry, collaboration is essential to cope with a rapidly changing market environment. This collaboration, however, makes business processes more complex and dynamic, and thus, it is necessary for manufacturing firms to employ systematic process modeling methods for effective and clear specification of collaboration processes. Currently, several modeling methods are available, such as Integration Definition Language 3 (IDEF3) [1], Unified Modeling Language (UML) [2-5], Petri-Nets (PNs) [6], ARchitecture of integrated Information Systems (ARIS) [7], and Business Process Modeling Notation (BPMN) [8]. However, these methods have limits in modeling of intricate collaborative processes. Responding to this need, Ryu and Enver [9] proposed Collaborative Process Modeling (CPM).

In this paper, we propose an extended CPM (exCPM). The original modeling method of CPM is incorporated with the concept of state and color tokens in PNs in order to visualize the real-time status of collaborative processes and highlight participants in collaboration processes. In addition, ICOM (Input, Control, Output, and Mechanism), a major concept of IDEF0 [1], is built into exCPM to explicitly represent transaction information between processes.

2. Analysis of Process Modeling Methods.

2.1. Pervasive process modeling methods. IDEF3 allows us to specify processes as ordered sequences of events or activities including relations between time-based behaviors [1]. While IDEF3 provides an easy way to describe static processes, it is not suitable for representing dynamic processes. Also, IDEF3 does not provide a systematic procedure for model verification. In terms of collaborative process modeling, the major concern of the paper, it is difficult to identify collaboration processes of multiple participants using IDEF3, since each activity in IDEF3 is generally presumed that it is performed by a single actor.

UML is a de facto standard modeling language developed by the Object Management Group (OMG) to specify, visualize, and document models of systems [3]. Although UML is syntactically rich, user friendly, and flexible, it lacks ‘formality’ which is essential to support simulation and subsequent analysis [10]. Many researchers, therefore, have tried to incorporate other modeling techniques into UML such as PNs. There also exist formalization efforts focused on static elements, leaving dynamic semantics almost untouched [10]. While a collaboration diagram in UML allows us to represent collaborative processes, it is another way of representing an activity diagram, which is not sufficient to fully describe collaborative processes [9].

BPMN is a standard notation for capturing business processes, especially at the level of domain analysis and high-level systems design [11]. The notation inherits and combines elements from a number of previously proposed notations for business process modeling, including XML process definition language (XPDL) [11] and the activity diagrams component of UML. Although collaboration can be represented by using notes or referents in BPMN, it is still primitive and incomplete.

ARIS is a technique for modeling business processes, and it provides event-driven process chain (EPC) method [7]. While it is possible to represent actors of each process by using the organization view of ARIS, which is necessary to indicate multiple participants of collaborative processes, there is no direct method to verify the consistency and completeness of model. To tackle this issue, van der Aalst [12] developed a method to transform EPC into PNs by mapping an event to a place and a function to a transition. However, their works do not support the transformation of collaborative processes because organization entities are not considered during the model transformation.

PNs have been widely used to design and analyze concurrent and dynamic processes [13]. By tracing the flow of tokens, we can diagnose or anticipate dynamically changing system states such as reachability, liveness, boundedness, conservation, and consistency [14]. However, it is still difficult to address collaboration. We can possibly represent all relevant participants by a single PNs model, or use colored tokens to differentiate individual participants. However, this type of approach is not scalable, because adding a new participant will severely increase the model complexity [9].

2.2. Collaborative process modeling (CPM). CPM allows us to model collaborative processes as well as normal processes among multiple actors with different affiliations [9]. CPM has the following characteristics [9]: 1) CPM is process-oriented, 2) CPM is based on UML notation and consists of eight elements including normal, intra-, and inter- collaboration process, decision, synchronization, process transition, resource, and reference note, 3) CPM models are easy to understand since three types of processes are represented with different symbols, 4) processes carried out by corresponding participants can be modeled in a single CPM model, and each participant is explicitly identified, and 5) CPM models can be transformed into marked graphs so that it is possible to use many proven analytical methods of PNs. To transform CPM models into PNs ones, marked

graph building blocks (MGBB) have been defined and applied following the pre-specified transformation rules. For more details on these transformation rules, refer to [9].

However, CPM is still conceptual, and no software tool is available to support it. Furthermore, CPM models are yet static so do not support real-time monitoring of dynamic processes and representation of transaction information among collaborative processes. These limitations have motivated us to extend CPM, which will be discussed in this paper.

2.3. Comparison of process modeling methods. In order to identify necessary features for collaborative process modeling, various pervasive modeling methods including CPM have been compared and analyzed. Comparison criteria have been developed under general modeling and collaborative process modeling as illustrated in Table 1. We consider four and six comparison criteria for the properties of general modeling and collaborative process modeling, respectively.

TABLE 1. Comparison of process modeling methods

Criteria	IDEF3	UML	BPMN	ARIS	PNs	CPM
¹⁾ Model simplicity	Very simple (but not available for very complex models)	Simple	Fairly simple	Simple	Fairly simple (but complex for large-sized models)	Fairly simple
¹⁾ Understandability	Very easy	Easy	Fairly easy	Easy	Mostly difficult	Fairly easy
¹⁾ Representability	Not very strong	Fairly strong (restricted to diagrams supported)	Strong	Fairly strong	Very strong	Strong
¹⁾ Standardization	Very strong	Fairly strong	Strong	Strong	Very weak (many versions)	Strong
²⁾ Representation of multiple actors in collaboration	Impossible	Limited (with collaboration diagram)	Limited (with definition)	Possible (with organizational symbols)	Possible (but models become complex)	Possible (with definition of <i>Actor-ID</i>)
²⁾ Representation of transaction information	Impossible	Limited	Limited	Limited	Impossible	Limited
²⁾ Model verification	Not directly (via PNs transformation)	Not directly (via PNs transformation for certain diagrams)	Possible (through BPEL)	Not directly (via PNs transformation)	Powerful (with formalism)	Not directly (via PNs transformation)
²⁾ Dynamics	Impossible (static)	Impossible (static)	Limited (through BPEL)	Impossible (static)	Powerful (with tokens)	Impossible (static)
²⁾ Real-time monitoring	Impossible	Impossible	Possible (through BPEL)	Impossible	Possible (with tokens)	Impossible
²⁾ Model simulation	Not directly (via WITNESS or PNs transformation)	Not directly (via PNs transformation)	Possible (through BPEL)	Not directly (via WITNESS, etc.)	Possible	Not directly (via PNs transformation)

*Note: ¹⁾ properties for general modeling, ²⁾ properties for collaborative process modeling.

On the one hand, regarding general modeling properties, most pervasive methods including IDEF3, UML, BPMN, ARIS reflect good results except for PNs. They are high-leveled and graphic-based methods so that users can easily develop process models.

On the other hand, the most important issue in representing collaborative processes is the capability of representing multiple participants in collaboration, which can be handled by ARIS and CPM. While PNs can also support it, the model usually becomes very complex. All methods considered in this paper have a limitation on the representation of transaction information, because they are focusing on business processes instead of data/information. To comprehensively represent processes and data together, IDEF*, which merges concepts of IDEF0 and IDEF3, has been proposed [15]. In terms of model

verification, PNs show a distinctive feature owing to their formalism. Therefore, coupling of PNs with other methods has been attempted in the literature to compensate for the weakness in verification of those methods and exploit the analysis capabilities of PNs [9,10,12,13].

In addition to model verification, PNs are more powerful to represent dynamic behavior than other methods. This facilitates real-time monitoring and simulation of processes. However, PNs become extremely complicated in modeling processes, which makes it difficult to understand large-sized PNs models at once.

The limitations of various modeling methods reveal that the following features are required in modeling methods for collaborative processes.

- 1) Multiple actors in collaboration should be clearly represented and easily recognized.
- 2) Transaction information among collaborative processes should be explicitly represented.
- 3) Model verification should be possible (directly or via model transformation).
- 4) Simulation capability to predict future status to cope with unexpected situations.
- 5) Real-time monitoring capability should be supported for tracking current status.

3. exCPM (extended CPM).

3.1. Definition of exCPM. As depicted in Table 2, exCPM inherits seven elements of CPM except for reference note, but instead, includes additional three elements such as state token, color token, and input/output/control.

The concept of state and color tokens of PNs has been applied to CPM. This addition enables us to monitor the real-time status of each process and to comprehend the diverse actors' behaviors in collaboration. The modified concept of ICOM in IDEF0 [1] is also added to represent transaction information. Note that ICOM concept denotes input, control, output, and mechanism. Since resources can be represented by the resource element already defined in CPM, 'mechanism' is omitted to avoid duplication of the same information.

As data and process flows do not usually match with each other, we need to indicate both flows. Process transitions and transaction data are expressed with bold and dotted arrows, respectively. Decision, resource and synchronization follow the same representation schemes as in CPM. The main features of exCPM can be summarized as follows:

- 1) It consists of 10 elements based on those available in CPM (refer to Table 2).
- 2) Different symbols for normal/intra-/inter-collaboration are used for clarity.
- 3) With state tokens of PNs, dynamic expression and process monitoring are possible.
- 4) With color tokens of PNs, multiple actors can be represented distinctively.
- 5) ICOM elements allow us to clearly represent transaction information among processes.

The proposed exCPM is defined with 10-tuple.

$$exCPM = \{P, CT, ST, PT, D, R, S, I, C, O\}$$

- (1) P is the set of processes including normal processes (P_{Normal}), intra-collaboration processes (P_{Intra}), and inter-collaboration processes (P_{Inter}).

$$P = P_{Normal} \cup P_{Intra} \cup P_{Inter} = \{p_k | k = 1, \dots, K\}$$











where, k = the serial number of each process, and K = the total number of processes.

- (2) CT is the set of color tokens indicating actors who are involved in the processes.

$$CT = \{ct_k^a | k = 1, \dots, K, a \geq 1\}$$

where, a = the number of color tokens in process k . Here, if $a = 1$ then it means the process k is a normal process. If $a \geq 2$ then the process k is a collaborative process.

TABLE 2. exCPM elements

Symbol	Description	Symbol	Description
	Normal process		Process transition
	Intra-collaboration process		Input/Output/Control
	Inter-collaboration process		Synchronization
	Decision		State token
	Resource		Color token

- (3) ST is the set of state tokens which indicate the current execution status of processes.

$$ST = \{st_k^u | u = 0 \text{ or } 1, k = 1, \dots, K\}$$

where, u = the number of state tokens in the process k .

- (4) D is the set of decisions, which determine process flows.

$$D = \{d_{(f,y,n)}^j | j = 1, \dots, J; f, y, n \rightarrow [1, k], f \neq y \neq n\}$$

where, j = the serial number of decisions, f = id of the process triggering the decision d^j , y = id of the process to be conducted when “yes” chosen, and n = id of the process to be conducted when “no” chosen.

- (5) S is the set of synchronizations, which synchronize processes.

$$S = \{s_{(L; M)}^x | x = 1, \dots, X\}$$

where, x = the serial number of synchronization, L = the set of pre-processes, and M = the set of post-processes and/or decision.

- (6) PT is the set of process transitions indicating the process flow.

$$PT = [pt_{(\alpha,\beta)}^e | e = 1, \dots, E; \alpha, \beta \in \{P \cup D \cup S\}]$$

where, e = the serial number of process transition, $\alpha = P, D$, or S where process transition starts (i.e., starting point of the pt^e), and $\beta = P, D$, or S where process transition ends (i.e., ending point of the pt^e).

- (7) R is the set of resources indicating the means used to perform a function, which is a similar concept to mechanism in IDEF0.

$$R = \{r_k^q | k = 1, \dots, K, q \geq 0\}$$

where, q = the serial number of each resource.

- (8) I is the set of inputs used to represent data or objects that will be transformed into outputs by the process k . Input arrows are connected to the left side of a process box.

$$I = \{i_k^v | k = 1, \dots, K, v \geq 0\}$$

where, v = the identification number of input of the process k .

- (9) C is the set of controls representing data or objects as constraints required to produce correct output. Control arrows are connected to the top side of a process box.

$$C = \{c_k^i | k = 1, \dots, K, i \geq 1\}$$

where, i = the identification number of control of the process k .

(10) O is the set of outputs representing data or objects produced by the process k . Output arrows are connected to the right side of a process box.

$$O = \{o_k^z | k = 1, \dots, K, z \geq 1\}$$

where, z = the identification number of output of the process k .

Figure 1 illustrates an inter-collaboration process among process notations of exCPM.

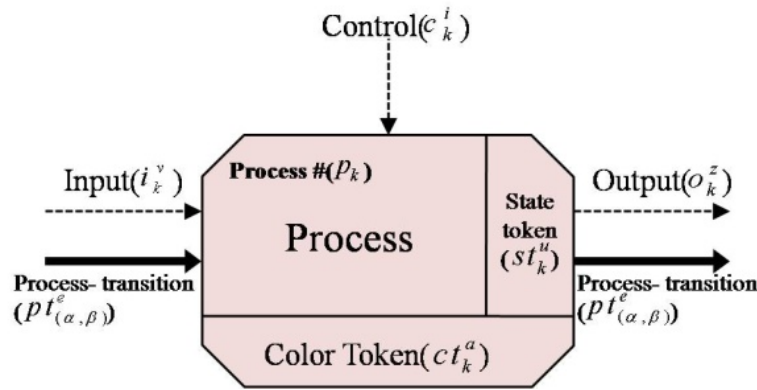


FIGURE 1. Inter-collaboration process

3.2. **Illustrative example of exCPM.** Similar to CPM, exCPM can decompose a model into three levels including company, department, and individual, to provide more detailed processes as well as information as it goes down to the lower levels. Figure 2 illustrates a simple exemplary exCPM model at the company level.

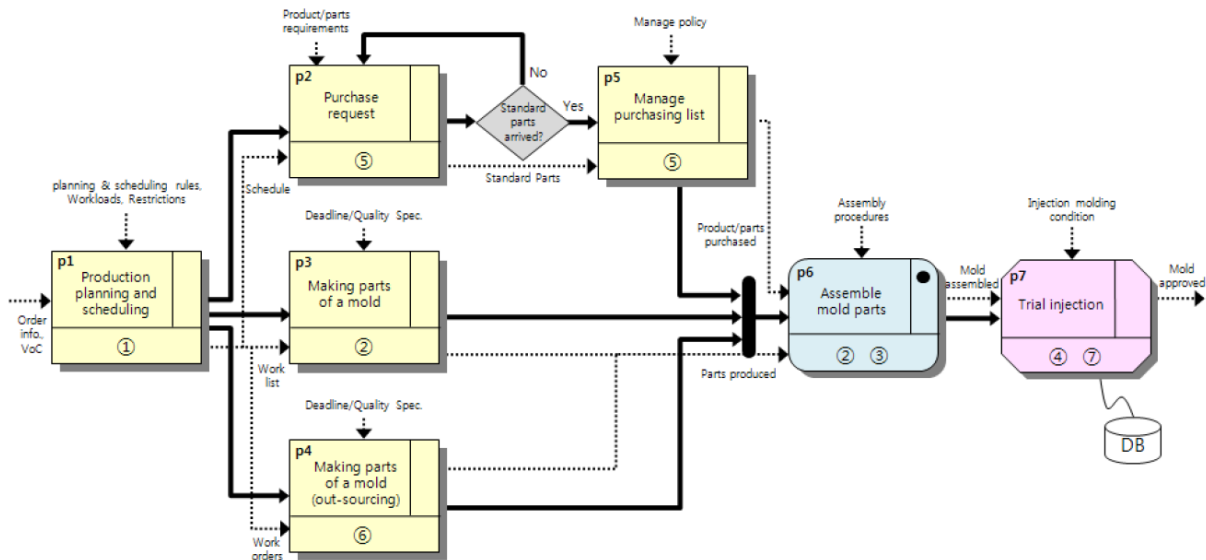


FIGURE 2. Examples of an exCPM model at the company level

In Figure 2, “p6” denotes “Assemble mold parts”. It is an intra-collaboration process (represented by a rounded rectangle), and in-progress process (indicated by existence of a state token). The color tokens ② and ③ reveal that the mold production and assembly teams of the company A are involved in the process (refer to Table 3). Note that the colors in each token are converted to numerical identifications in this paper to increase readability. In the process “p6”, the inputs of purchased (or produced) product/parts are

processed into the output ‘mold assembled’ under the control of ‘assembly procedures.’ As another example, the process “p7” is “Trial injection” activity. It is an inter- collaboration process (represented by an elongated octagon). In this process, the quality management team of company A (color token ④) and the trial injection team of company C (color token ⑦) are involved. As there is no state token, it is not in-progress status.

TABLE 3. Company and departments for the exCPM model of Figure 2

Company	Color token	Company/Department
Company A		Mold production company
	①	Marketing/strategy team
	②	Mold production team
	③	Mold assembly team
	④	Quality management team
Company B	⑤	Purchasing/procurement team
		Mold outsourcing company
Company C	⑥	Mold production team
		Trial injection company
	⑦	Trial injection team

When expressing process and data flows, dotted and bold arrows are different, as shown in Figure 2. For expressing process box, arrows can be connected to all four sides of a process symbol. On the other hand, for data flows, inputs, outputs, and controls are always connected to the left, right, and top side of a process box respectively, as defined in IDEF0.

3.3. Implementation of exCPM modeler. The exCPM modeler is developed based on “Microsoft .Net framework” (see Figure 3). The model verification module is still under development. Therefore, we have used the current exCPM modeler to make collaborative process models but conducted model verification only manually.

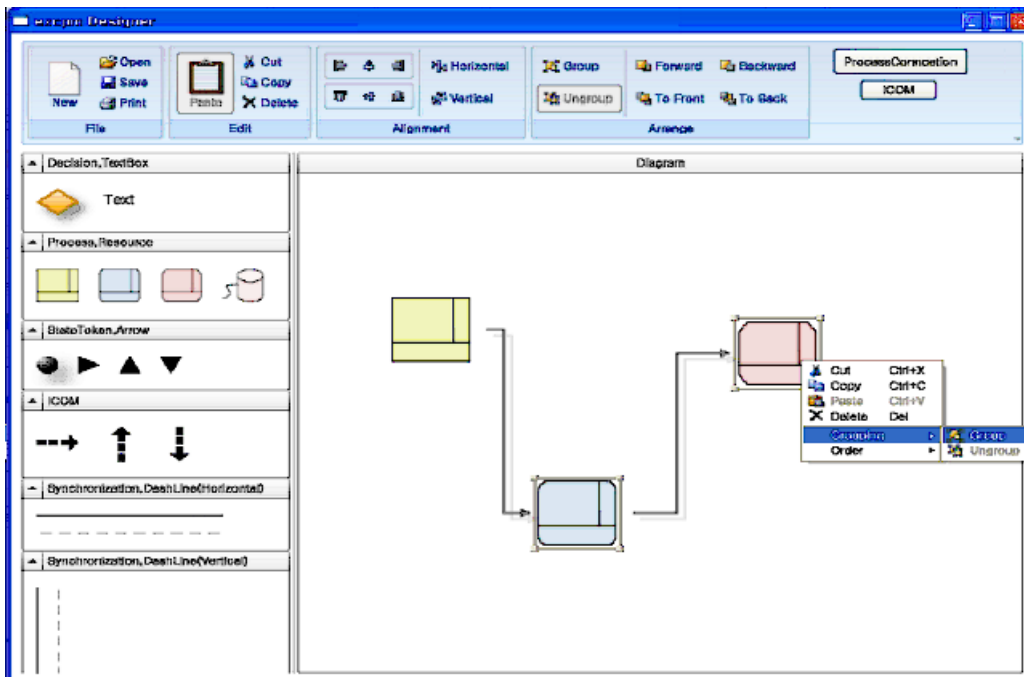


FIGURE 3. exCPM modeler

4. Verification of exCPM Models.

4.1. **Transformation rules.** In order to verify CPM models, it is necessary to convert a CPM model into a corresponding PN model. For this, Marked Graph Building Block (MGBB) was defined as transformation rules [9]. Since most exCPM elements are based on the existing CPM method, it is also possible to transform exCPM models into PN models using similar transformation rules. In order to take state tokens of exCPM into account, we need to modify the original transformation rules of CPM. Table 4 shows the modified MGBB for exCPM. Table 5 illustrates the rules to select a proper MGBB for each collaborative process, where PB0 ~ PB3 are the process blocks represented in Table 4. PB1 is the extended MGBB from PB0 by adding “Input for extension” to the input transition of PB0 and “Output for extension” to the output place of PB0. By doing so, PB1 can illustrate actors who belong to the organization represented in PB0. In a similar manner, PB2 and PB3 can be defined by extending PB1 and PB2, respectively.

When a department (or individual) is selected for the transformation level, it is possible to make a link between processes of different departments (or individuals) if and only if they belong to the same company. At this point, we have to insert an additional MGBB, i.e., “synchronization” (either split-type or join-type) between boundary processes of different department (or individual). However, note that “synchronization” is not necessary for the links to “decision” or “ending mark”.

TABLE 4. Marked graph building block (MGBB)

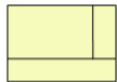

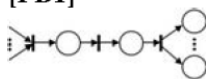






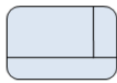
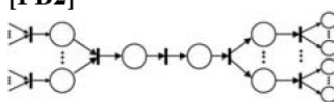
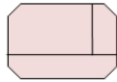
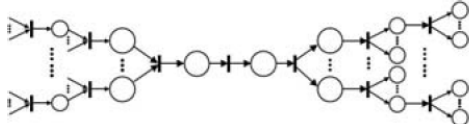






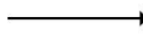



Symbol	Marked Graph Building Block				
 Normal process	<div style="display: flex; align-items: center;"> <div style="flex: 1;"> <p>[PB0]</p>  <p>[PB1]</p>  </div> <table border="1" style="margin-left: 20px;"> <tr> <td>Input for extension</td> <td>Output for extension</td> </tr> <tr> <td></td> <td></td> </tr> </table> </div>	Input for extension	Output for extension		
Input for extension	Output for extension				
					
 Intra-collaboration process	<p>[PB2]</p> 				
 Inter-collaboration process	<p>[PB3]</p> 				
 Synchronization	 (Split-type)  (Join-type)				
 Decision	(linked from a place) 				
 Process transition	 (link places to transitions)				
 Resource	 (link to transition)				
Ending mark (additional)	 (start) -link to the first transitions (end)				

TABLE 5. MGBB selection rules for transformation

Process \ Level	Company	Department	Individual
Normal Process	[PB0]		
Intra-collaboration Process	[PB0]	[PB1]	[PB2]
Inter-collaboration Process	[PB1]	[PB2]	[PB3]

The revised transformation rules for exCPM models are listed as follows:

- Rule 1:** Remove ICOM from exCPM models, since they are not defined as MGBB.
- Rule 2:** Choose an appropriate level of detail for PN models among company/department/individual levels.
- Rule 3:** Replace the exCPM elements with MGBB to generate PN models. A selection rule is used for a proper MGBB choice.
- Rule 4:** Add an “ending mark” between the transition(s) of the MGBB indicating the first process and the place(s) of the MGBB meaning the last process.
- Rule 5:** If any process has a state token, locate token(s) into the first place(s) of MGBB in the PNs model.

When we transform an exCPM model into a PN model, the PN model can have a different number of places according to the chosen transformation level. In order to keep consistency in assigning tokens into corresponding places, we have to use Rule 5. In short, if a process in the exCPM model has a state token in it, a token must be assigned to the first place of MGBB in the PN model.

4.2. **PN model.** Following the aforementioned transformation rules, we have transformed the exCPM model in Figure 2 to a PN model. The converted models are depicted in Figures 4 and 5, which represent PN models of the company level and the department level, respectively. A PN model of the individual level can be derived in a similar manner.

4.3. **Analysis of PN model.** In this section, we discuss reachability analysis for the dotted area in Figure 4 for illustration purposes. In the reachability tree in Figure 6, we define each state with a corresponding marking, M_n , and remain markings only when afterward states occur repeatedly. For example, the state M_{14} can be reached not only

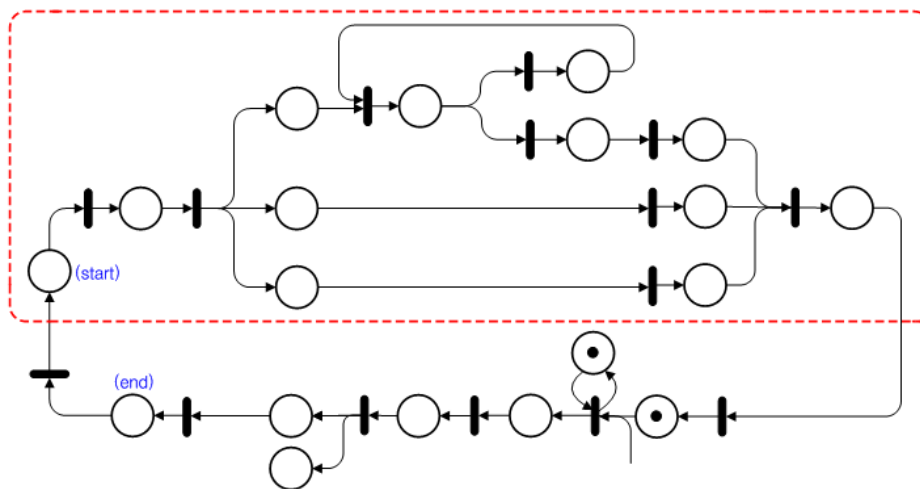


FIGURE 4. Company-level PN model by model transformation

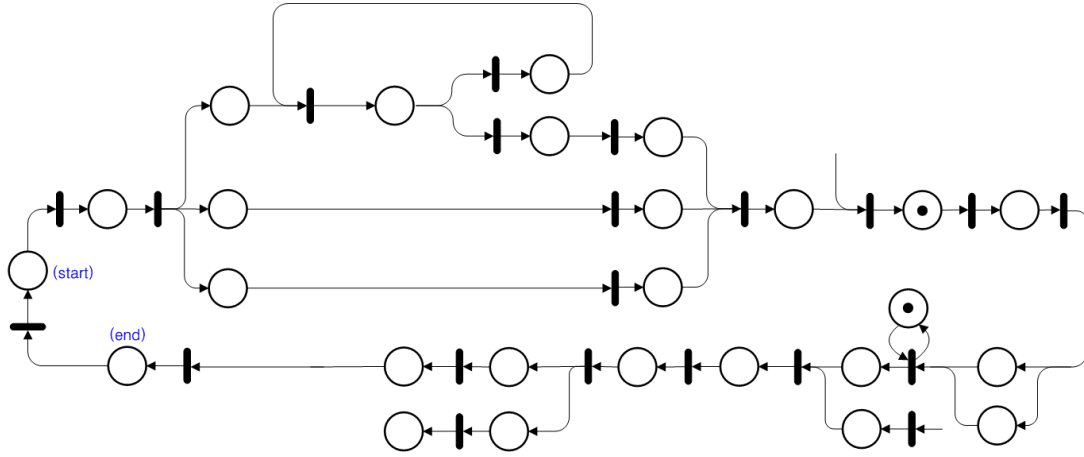


FIGURE 5. Department-level PN model by model transformation

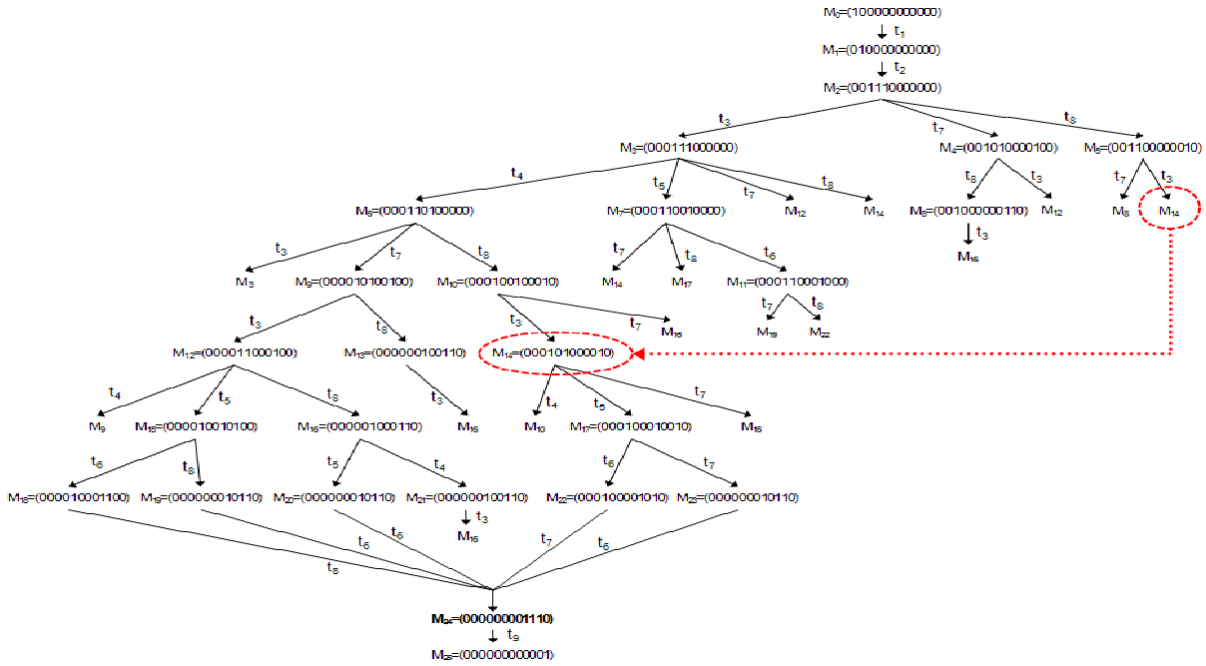


FIGURE 6. The reachability tree of the PN model in Figure 4

by firing $t_8 \rightarrow t_3$ from M_2 but also by firing $t_3 \rightarrow t_4 \rightarrow t_8 \rightarrow t_3$ from M_2 sequentially. Therefore, we omit expanding the tree from the former (M_{14}) since states after M_{14} will be the same after the latter, i.e., $M_{14} = (000101000010)$. From various states including $M_{18}, M_{19}, M_{20}, M_{22}, M_{23}$, the final state after firing t_8, t_6, t_6, t_7 , and t_6 respectively is M_{24} which indicates the state that all pre-requisite conditions are satisfied.

By investigating reachability tree or coverability tree, we can verify whether processes in the model are bounded or not. Normally, PNs are bounded iff for each place p there is a natural number k so that for every reachable state the number of tokens in p is less than k [12]. Especially, if the model is k -bounded and k equals 1 then it is called ‘safe’. As illustrated in Figure 6, we can verify k equals 1, therefore the model is bounded and safe.

As another analysis, PNs are called live iff for every reachable state M' and every transition t , there is a state M'' reachable from M' , which enables t [12]. Furthermore, if the initial state M_0 is reachable from all state with a corresponding marking M_n , then PNs are called reversible. As illustrated in Figure 6, we can verify that all transitions can be fired without any deadlock or blocking state. Even though the final state M_{25} in Figure 6 is not the last process of PNs in Figure 4, it is trivial that firing of remaining transitions will lead the process state to the initial state M_0 . These reasons reveal that PNs in Figure 4 are live and reversible. Furthermore, PNs are persistent and consistent because all states can be reachable to the initial state and there is no unexpected state during the process.

With the extracted reachability tree, the analysis results support the following conclusions: 1) a PN model, automatically transformed from an exCPM model, was well developed, 2) the proposed transformation rules work effectively, and finally 3) an initial exCPM model is proved to be well defined.

5. Concluding Remarks. This paper proposed exCPM which is capable of providing a visual snapshot of collaborative interactions, and enhanced modeling and analysis for collaborative processes. One of the remarkable characteristics of exCPM is its model verification capability via automatic transformation of exCPM models into PN ones. In this paper, we refined transformation rules of the original CPM and demonstrated efficiency of exCPM using an illustrative model. The results reveal that exCPM contributes to describing collaborative interactions in business domain. The proposed exCPM method will be useful to capture, visualize, and verify internal mechanisms of collaboration among cooperating partners so that the efficiency of business workflows will be substantially improved. While many of the functions proposed in this paper have been implemented in the developed exCPM modeler, further works are required to implement the other remaining functions such as model transformation and verification.

Acknowledgment. This work was supported by the 2010 Specialization Project Research Grant funded by the Pusan National University.

REFERENCES

- [1] R. J. Mayer, C. P. Menzel, M. K. Painter, P. S. deWitte, T. Blinn and B. Perakath, *Information Integration for Concurrent Engineering IDEF3 Process Description Capture Method Report*, KBS Inc., 1995.
- [2] G. Booch, J. Rumbaugh and I. Jacobson, *The Unified Modeling Language User Guide*, Addison-Wesley, 1999.
- [3] J. Rumbaugh, I. Jacobson and G. Booch, *The Unified Modeling Language Reference Manual*, Addison-Wesley, 1999.
- [4] R. Pooley and P. Stevens, *Using UML Software Engineering with Objects and Components*, Addison-Wesley, 1999.
- [5] OMG, *Unified Modeling Language (UML)*, <http://www.uml.org>.
- [6] C. A. Petri, *Communication with Automata*, DTIC Research Report AD0630125, 1966.
- [7] A. W. Scheer, *Business Process Engineering: Reference Models for Industrial Enterprises*, Springer, Berlin, 1994.
- [8] R. M. Dijkman, M. Dumas and C. Ouyang, Semantics and analysis of business process models in BPMN, *Information and Software Technology*, vol.50, no.12, pp.1281-1294, 2008.
- [9] K. Ryu and E. Yucesan, CPM: A collaborative process modeling for cooperative manufacturers, *Advanced Engineering Informatics*, vol.21, no.2, pp.231-239, 2007.
- [10] L. Baresi, Improving UML with Petri Nets, *Elect. Notes in Theoretical Computer Science*, vol.44, no.4, pp.1-13, 2001.
- [11] WfMC, *Business Process Modeling Notation*, <http://www.pxdl.org/nugen/p/bpmn/public.htm>.

- [12] W. M. P. van der Aalst, Formalization and verification of event-driven process chains, *Information and Software Technology*, vol.41, no.10, pp.639-650, 1999.
- [13] K. Santarek and I. M. Buseif, Modelling and design of flexible manufacturing systems using SADT and Petri nets tools, *Journal of Materials Processing Technology*, vol.76, no.1-3, pp.212-218, 1998.
- [14] G. Zhou, Y. He and P. Yu, Modeling workflow patterns based on P/T nets, *International Journal of Innovative Computing, Information and Control*, vol.1, no.4, pp.673-684, 2005.
- [15] C. L. Ang, L. P. Khoo and R. K. L. Gay, IDEF*: A comprehensive modelling methodology for the development of manufacturing enterprise systems, *Int. J. Prod. Res.*, vol.37, no.17, pp.3839-3858, 1998.