IE Interfaces
Vol. 20, No. 1, pp. 79-86, March 2007.

Petri Net-based Process Modeling for B2B e-Commerce

Sunho Kim†

Department of Industrial & Management Engineering, Myongji University

기업간 전자거래를 위한 Petri Net 기반 프로세스 모델링

김선호

명지대학교 산업경영공학과

In B2B e-commerce environments, many initiatives for process modeling have made efforts to design business processes correctly. Especially, Petri nets have been widely used as a good theory to design and verify process models. Therefore, we propose the process modeling method for the B2B e-commerce based on Petri nets. First of all, a B2B process model based on BPSS is represented by the UML activity diagram. Second, the activity diagram is transformed to a Petri net model. For the transformation, well-behaved building blocks/control structures and the modeling rules for inter-organizational workflow processes are proposed. Third, the process is partitioned into sharable processes for individual business partners. Finally, according to needs of individual business partners, the sharable processes are modified by well-structured refinement rules. The whole procedure is explained with the purchase process of an e-bookstore.

Keyword: BPM, workflow, BPSS, Petri net, B2B, e-commerce, business process

1. Introduction

In order to improve the performance of e-commerce, many initiatives have made efforts to design business processes correctly. Especially in B2B e-commerce, not only public processes exchanged among business partners but private processes progressed within a business partner's organization are of interest for correct process modeling. Today's B2B standards such as RosettaNet and ebXML, describe how to establish a trading partnership with specific commercial semantics and how to execute business processes between business partners. However, these standards define specifications only for public process, but not for private process or integrated processes between the two. For example, BPSS (Business Process Specification Schema) in ebXML specifies business transactions among business partners for public process.

For the seamless implementation of B2B processes, a modeling method which integrates public processes with private processes is inevitable. However, when public processes are integrated with private processes, modeling errors such as dead lock and live lock may occur (Aalst, 1999). Furthermore, there may be a wrong case that the moment the whole process terminates some activities within the process are still working. Therefore, it is desirable to verify that the B2B processes designed have no errors before full implementation. Though modeling languages/tools such as BPML (Business Process Modeling Language) (BPMI.org, 2002) and UML (Unified Modeling Language) can be used for the integrated process design, they do not have any function or logic to check these errors. However, Petri nets provide good theories to check modeling errors such as soundness, free-choice,

†Corresponding author : Professor Sunho Kim, Department of Industrial & Management Engineering, Myongji University, San 38-2, Namdong, Cheoingu, Yongin, Kyunggido, 449-728, Korea, Phone : +82-31-330-6451, E-mail : shk@mju.ac.kr
Received August 2006, accepted January 2007 after 2 revisions.

well-structuredness, IO-soundness, 1-consistency, etc. not only for individual private processes but also for the whole processes that public and private processes are integrated (Aalst, 1998; 1999). Furthermore, the Petri-net-based modeling uses graphical notations and has a high expression capability which facilitate modeling, analysis, or simulation (Murata, 1989). Woflan is one of the software tools to check these modeling errors base on Petri nets (Hauschildt, 1997).

Furthermore, Petri nets are very useful for the software design and the performance evaluation of computer-supported collaborative works. In references (Bernardi, 2002; Hu, 2004; King, 2000; Shishkov, 2002), class, sequence and state diagrams written in UML are transformed to Petri nets for the analysis or simulation of dynamic behavior. (Garrido, 2002) proposed transformation method from activity diagrams to Petri nets, using the example of collaboration between emergency coordination centers in Sweden and USA. (Lin, 2004) also utilized similar conversion method to analyze structural errors of CAD design processes. (Yan, 2001) suggested a method to transform class and activity diagrams to colored Petri net based on the SET protocol which provides a secure solution for credit card payment. (Gou, 2000) proposed the method to map UML diagrams to Petri nets for model analysis and simulation of business processes in virtual enterprises. In this method, operations in class diagrams are transformed to places, message events in sequence diagrams to transitions, and activities/state transitions in activity and statechart diagrams to arcs between places and transitions. However, the transformation method is not described in detail.

In this paper, we propose the method to generate private processes from B2B public processes through the transformation of UML activity diagrams to Petri nets. First of all, a B2B process model represented by the UML activity diagram. In this processes, public processes are defined based on BPSS developed by UN/CEFACT. Second, the activity diagram is transformed to a Petri net model. For the transformation, well-behaved building blocks/control structures and the modeling rules for inter-organizational workflow processes are proposed. Third, the process is partitioned into private processes what is called "sharable processes" for individual business partners. Finally, according to needs of individual business partners, the sharable processes are modified by well-structured refinement rules. By these rules, the sharable process does not lose the logical correctness of the public processes. The whole procedure is explained with the purchase processes of an e-bookstore.

2. Public Processes Based on BPSS

UN/CEFACT has developed BPSS as the standard specification for B2B public processes between business partners (UN/CEFACT, 2003). In this specification, a business activity is represented by a BT (Business Transaction). Therefore, in this section, characteristics of a BT such as success/failure states and responses in message exchange are briefly described and an example of BT is described since these properties are taken into account when transformation to Petri nets. Then, an example of B2B public processes which is defined by BPSS and depicted by UML activity diagram is given for the explanation of the proposed methodology.

2.1 Characteristics of a Business Transaction

A BT is an atomic unit of work in a trading arrangement between two business partners. The schematic of core BT semantics is depicted in <Figure 1>. A BT consists of a Requesting Business Activity, a Responding Business Activity, and one or two document flows between them. A BT may support one or more Business Signals that govern the use and meaning of acknowledgements. These acknowledgment signals are application level documents that signal the current state of the BT. The Receipt acknowledgement business signal, if used, signals that a message (Resquest or Response) has been properly received by the BSI (Business Service Interface) software component. The Acceptance Acknowledgement business signal, if used, signals that the message received (Request or Response) has been accepted for business processing by the receiving application, or a receiving business application proxy. This is the case if the contents of the message's business documents and document envelope have passed a business rule validity check.

In order for a BT activity to achieve a success state, it must complete with both a protocol and a business success. Protocol and business success conditions are as follows:

<protocol success>

no timeout would have occurred (signals or response)

no signal can have a negative content

<business success>

the response document sent to the requestor must be marked as isPositiveResponse = 'true' in the ebXML BPSS instance that specifies the business collaboration.

If either a protocol or business failure occurs, the BT activity will be put into a failure state.

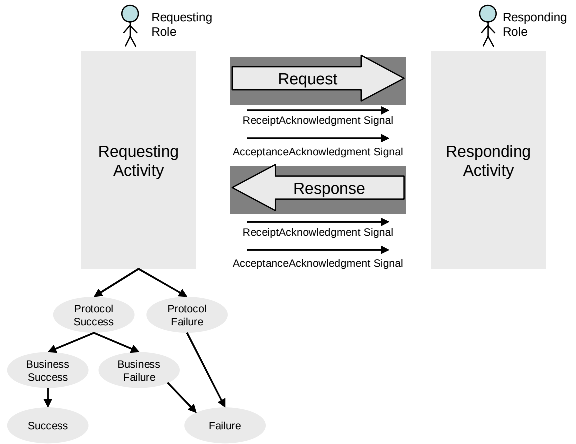


Figure 1. Schematic of core BT semantics (UN/CEFACT, 2003)

As an example of message exchanges in BPSS, consider a transaction example of PurchaseOrder(PO) shown in the <Figure 2>. The PurchaseOrder involves two possible responses to SendPO which issues a purchase order: AcceptPO and RejectPO which imply acceptance and denial of the purchase order, respectively. Note that in the actual execution of the transaction PurchaseOrder, only one of the defined possible responses will be sent. A seller receives the purchase order and responds with either acceptance or denial, including availability of the goods and the credit of the buyer. Obviously, the decision processes are opaque because they are internal, however, the fact of the decision must be reflected as behavioral alternatives in the public business process. In other words, the public process requires the selection of a branch, but the selection is non-deterministic from the perspective of the public process. In this situation, such a non-determinism can be modeled by allowing the assignment of a non-deterministic or opaque value to a process variable, typically from an enumerated set of possibilities. In this way, the process variable can be used to define the conditional behavior that captures behavioral alternatives without revealing actual decision processes.

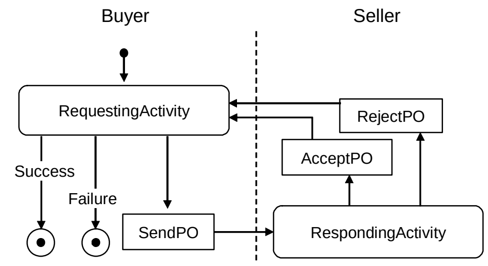


Figure 2. An example of the transaction PurchaseOrder (PO)

2.2 An Example of Public Process Based on BT Activities

In the following, we illustrate the method of modeling the process of B2B e-commerce with an example of the e-bookstore that has no books in stock. First of all, we design public processes over the business partners involved for the e-bookstore. <Figure 3> represents a simple B2B business process that consist of BT activities for a typical purchase process. This is depicted by the UML activity diagram.

This case has four roles, i.e., customer, bookstore, publisher, and shipper. The customer places an order that is represented by the BT activity PlaceOrder (BTA1). This customer order is sent to the bookstore. The bookstore then asks for the inventory status to an appropriate publisher by QueryInventory (BTA2). In this figure we use the attribute "onInitiation" that is defined in BPSS for a nested business transaction activity. The inventory query by the bookstore is then evaluated by the publisher. The publisher in turn informs the bookstore of the availability of the book. If the customer receives a negative answer, the workflow terminates. In this figure, the upper synchronization bar allows us to evaluate conditions on the context of the PlaceOrder and decide whether PlaceOrder is accepted or not. If the book is available, the customer is informed of a positive answer and the bookstore continues processing the customer order. The bookstore sends a request to the shipper by RequestShipping (BTA3), and informs the shipper of the shipping plan by NotifyShipping (BTA4). Then, the publisher sends the book to the shipper. The shipper notifies the bookstore of the delivery schedule by NotifyDelivery (BTA5). The bookstore in turn sends the delivery schedule and the bill along with the book to the customer by NotifyDelivery (BTA6) and SendBill (BTA7). After receiving both the book and the bill, the customer makes a payment by PayBill (BTA8), and the whole business process terminates.

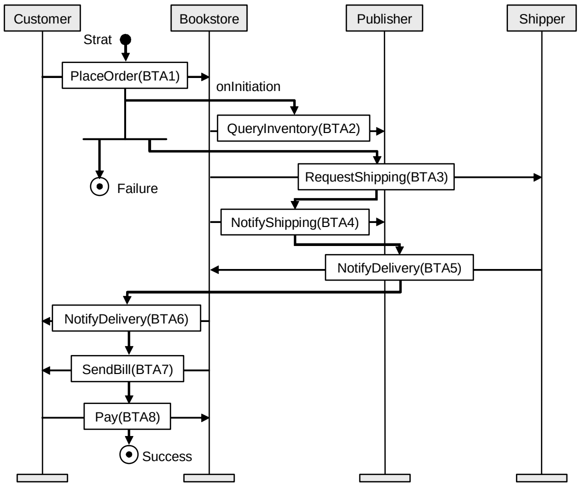


Figure 3. A B2B business process model for purchase order

3. Petri net based Process Modeling for B2B e-Commerce

We propose the method to convert the UML activity diagram based on BPSS to a Petri-net process. First of all, well-behaved building blocks and control structures are defined as transformation patterns. Based on these patterns, the method to convert a BT activity to a Petri net and the modeling rules to convert the whole activity diagram to Petri net processes are suggested. The activity diagram given in <Figure 3> is converted for illustration.

3.1 Well-behaved Building Blocks and Control Structures

WfMC has defined six well-behaved building blocks that can be used to build well-structured process models (WfMC, 2002). They can be represented by Petri nets as six well-behaved constructs shown in <Figure 4>. All constructs consist of places (circle), transitions (box), and arcs (arrow). Note that the AND-join and the AND-split are modeled by a transition with multiple arcs while the OR-join and the OR-split by a place with multiple arcs.

Similar to structured programming, we have modeled six well-behaved control structures, that is, parallel, selection, sequence, loop, begin and end structures, as shown in <Figure 5>. They are created by the combination of well-behaved building blocks proposed in <Figure 4>. Note that in the parallel structure the process splits by AND-split and joins by AND-join, and in the selection structure the process splits by OR-split and joins by OR-join. In <Figure 5>, there are two types of transitions, that is, control transitions (represented by C_{ij}) that take a role of routing controls, and task transitions (represented by T_{ij}) that perform business functions.

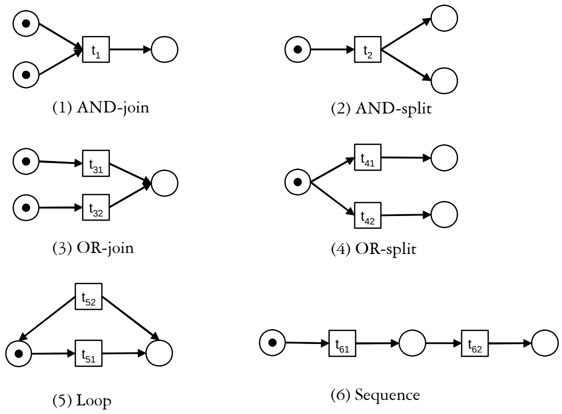


Figure 4. Petri-net representations of six well-behaved building blocks

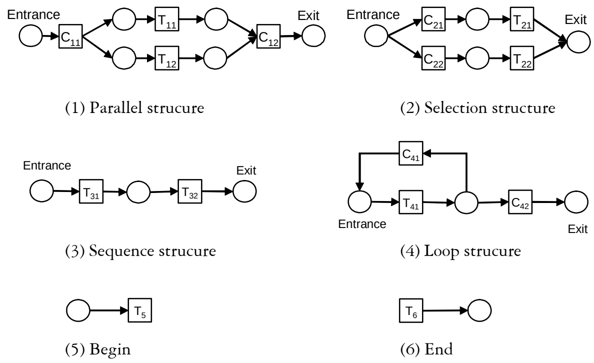


Figure 5. Petri-net representations of six well-behaved control structures

3.2 The Petri net Model for a BT

When a BT is represented by a Petri net, Request, Response, ReceiptAcknowledgement Signal, and AcceptanceAcknowledgement Signal can be considered as representation objects. Request and Response are Requesting Business Activity and Responding Business Activity, respectively, which may be accompanied with documents. Two Signals correspond to protocols which are automatically checked and sent by BSI or business applications. The BT terminates or rolls back in general when protocol failures occur due to false or null Signals. In the model proposed, business success/failure which makes the business process continued is of interest since protocol success/failure by signals is internally handled by software systems and protocol failure is directly related to the termination of the process or the system.

When Request and Response are considered in the modeling, there are two possible cases: 1) without Response and 2) with Response. The case without Response is the one when a one-way notification is sent to the responding partner. The case with Response is the one when the responding partner sends a message in return for the Request. The case without Response can be represented as in <Figure 6>. When Request sends a message, Receive may return back two signals, ReceiptAcknowledgement Signal and AcceptanceAcknowledgement Signal. At this time Request terminates successfully when two signal are all true. As Request and Receive communicates following the communication protocol., these two transitions are *synchronized*. In order to indicate the synchronized relationship, places and arcs between the two transitions are represented by dotted lines.

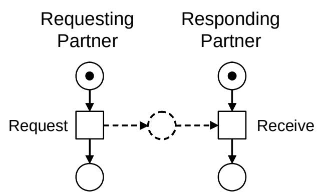
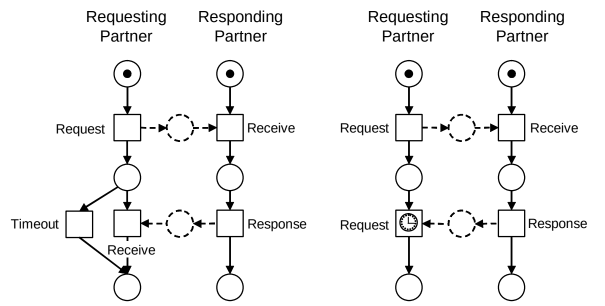


Figure 6. The representation of a BT to a Petri net without Response

On the other hand, the case with Response can be represented as in <Figure 7>. In this figure, when the requesting partner send the Request message, the responding partner receives it and responds the result through the internal decision making. The requesting partner then receives the message back and send thetwo Signals if required. If the requesting partner does not receive any message from the responding partner in a given time period, the transaction terminates in business failure. In a typical Petri net, the waiting task can be represented by the transition which works as a timer. The Timeout transition is depicted in <Figure 7(1)>. In the figure, Timeout and Receive are connected to one of the well-behaved building block, OR-split. This means the implicit OR-split that the transition which operates first uses the token in the input place (Aalst, 1998). However, this representation method makes the process model complicated since most of transitions have a time limit in response waiting. In order to simplify the process representation, a transition which has a timeout constraint is proposed. The transition is marked with ⏲ as in <Figure 7(2)>.

For the illustration of this representation, PurchaseOrder in <Figure 1> can be transformed by this method. In this example, Requesting Partner and Responding Partner correspond to Buyer and Seller. Request, Receive (Responding Partner), Response, and Receive (Requesting Partner) correspond to SendPO, ReceivePO, SendRT, and ReceiveRT, respectively (PO: Purchase Order, RT: Result. Refer to BTA1 in <Figure 8>). Seller's SendRT implies AccceptPO or RejectPO depending on the document attached, and Buyer's ReceiveRT implies AccceptPO, RejectPO, or Timeout.



(1) with the Timeout transition (2) with the transition which includes the Time-out function

Figure 7. The representation of a BT to a Petri net with Response

3.3 Modeling Rules for the Petri net-Based B2B Processes

Based on the transformation proposed in the previous section, we present modeling rules to design Petri net-based B2B processes. The modeling rules are as follows.

Rule1. Transform BT activities to Petri net processes. When encountering a nested business transaction represented by the attribute "*onInitiation*," the abstract transition can be used, which represents business transactions with coarse granularity. An abstract transaction may be detailed as a sub-Petri net representing a series of business transactions.

Rule 2. If any Petri net process ends with the transition ReceiveRT or SendRT, create multiple transitions from the output place of the transition by OR-split depending on the result received or sent.

Rule 3. The process of each partner must start with a single input place and a single output place for soundness (Aalst, 1998). Therefore, if there is a transition to terminate while modeling, mark its output place with "Exit." When the whole process comes to the end, all output places marked with Exit within a partner's process are integrated into one output place.

For the illustration of the modeling rules, we employ an example mentioned in <Figure 3>. The business process of the e-bookstore is transformed into a Petri net-based process as shown in <Figure 8>. The details of the modeling of this example are explained as follows.

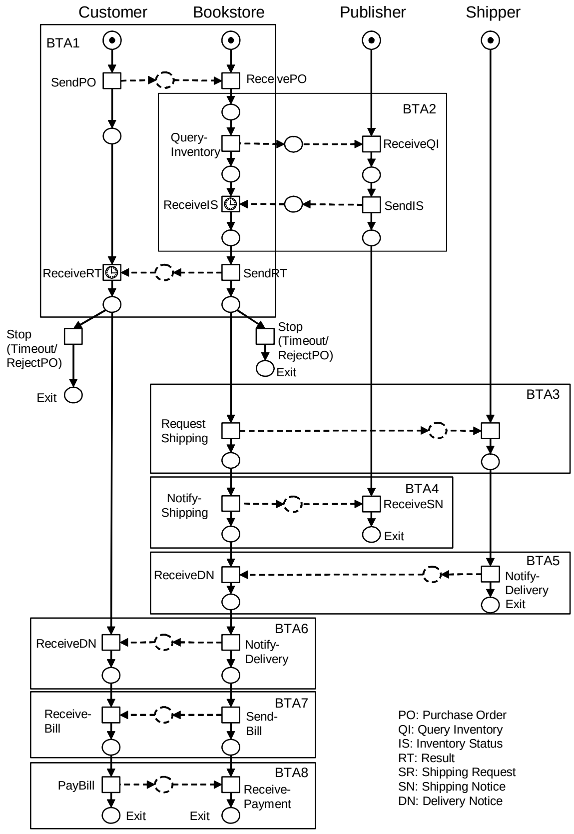


Figure 8. A Petri net-based B2B process for e-bookstore

First of all, by Rule 1, BTA1 and BTA2 are transformed to Petri net processes which have Response. Since BTA2 is the sub-BT activity of BTA1, Receive PO is directly connected to the input place of Query-Inventory, and the output place of ReceiveRT to SendRT. As Customer's process ends with ReceiveRT, the result can be separated to three cases: 1) time out, 2) RejectPO, and 3) AcceptPO. In the first two cases, the process stops (Stop in <Figure 9>), and in the third case, the process continues to the next BTA. Note that the output place of Stop is marked with "Exit" by Rule 3. Bookstore's process is extended in the same way since the process ends with SendRT. Other BTAs are all transformed to Petri nets which don't have Response. Note that the processes of Customer and Bookstore, respectively, have two output places marked with "Exit." Therefore, by Rule 3, the two output places are integrated into one output place. For the illustration of Rule 3, the process after integration is not shown in <Figure 8> (The integrated processes are shown in <Figure 9(1)> and <Figure 9(2)>).

4. Generation of Sharable Processes

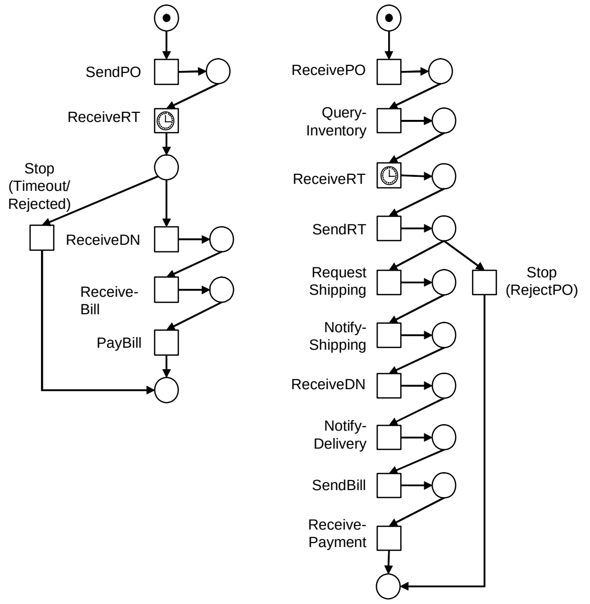
The process suggested in the previous section is a kind of 'skeleton' B2B business process which identifies all key tasks, message exchanges, and control and data dependencies between arbitrary two business partners. However, individual business partners need their own processes which include not only the private process but shared parts of public processes. The process of each business partner is called the *sharable process*. The sharable process can be generated from the Petri net-based B2B process. The method takes two steps: partition and modification.

First of all, the Petri net-based B2B process is partitioned into private processes of individual business partners. In the example of the e-bookstore, the whole process of the e-bookstore can be partitioned over four business partners. <Figure 9> shows sharable processes of the customer and the bookstore.

After sharable processes are generated, business partners can modify their own sharable process to incorporate new behaviors of their internal private business processes. In this modification, the *well-structured refinement rules* (*WSRR*) can be applied so that the correctness property of the overall B2B process be preserved (Aalst, 2002). There are four types of WSRR :

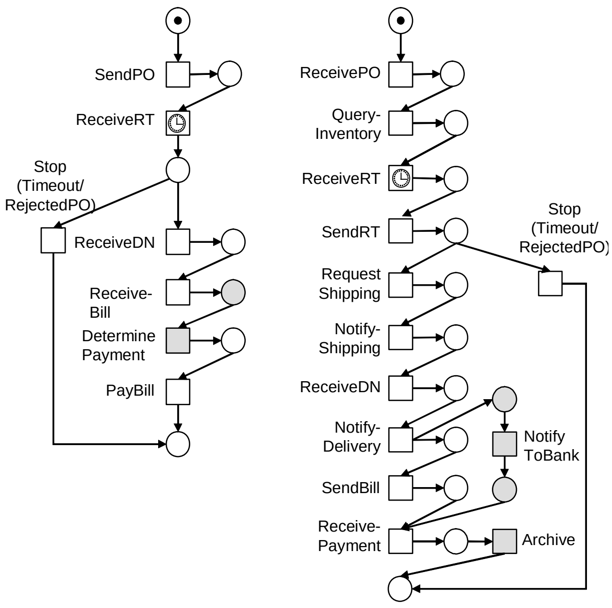
Rule 1: Inserting transitions between existing transitions;
Rule 2: Putting new transitions in parallel with existing transitions;
Rule 3: Putting new selective transitions between existing transitions;
Rule 4: Adding a loop.

Applying WSRR to sharable processes for Customer and Bookstore, we obtain the modified sharable processes as in <Figure 10>. <Figure 10(1)> shows Customer's sharable process with the transition DeterminePayment added by Rule 1. On the other hand, <Figure 10(2)> shows Bookstore's sharable process with two transitions added. The transition Archive is added by Rule 1 to save the business result, and the transition NotifyToBank is added by Rule 2 to open a bank account.



(1) Customer (2) Bookstore

Figure 9. Sharable Processes of Customer and Bookstore



(1) Customer (2) Bookstore

Figure 10. The modified sharable processes for business partners

5. Conclusion

In this research, we have proposed a method to design inter-organizational workflow processes especially for B2B e-commerce. Compared to existing approaches, this method has noble aspects as follows:

First, although our method has been designed based on the concept of references (Aalst, 1998; 1999), it is different from these references in that our process model has proposed the conversion method of UML-based business processes to Petri nets. Especially, business processes in this paper deals with messages exchanges of business transactions defined by BPSS. Messages within a business transaction has a relationship of request and response. Depending upon the relationship response results may be different, and timer transitions may be required to wait for responses from partners. However, other references including (Aalst, 1999) considered not the relationship of request and response in a transaction but the sequence of independent messages.

Second, although existing approaches convert UML diagrams to Petri nets, analyze the models, and feedback the results to UML diagrams, our method suggests the procedure to generate private processes needed for business partners from public processes by the transformation of UML activity diagrams to Petri nets. The private processes named sharable processes can be modified without losing information of public processes sharable between partners.

Most of researches have utilized the Petri net methodology to analyze the models designed by UML. However, they have not verified whether the Petri net models converted have any logical errors. Our modeling method also needs to verify whether the inter-organizational workflow process and its sharable processes are all modeled well. While individual private processes are relatively easy to verify the correctness, inter-organizational processes are more complex to verify because of synchronization relationships between private processes. For example, there exist cases where private processes are sound but the inter-organizational workflow process is not sound, and vice versa. For the verification of these processes, further researches are required: 1) verification of the soundness of the inter-organizational workflow process with synchronization relationships, and 2) the effect of sharable process modifications on the soundness of the inter-organizational workflow process.

References

Bernardi, S., Donatelli, S. and Merseguer, J. (2002), From UML Sequence Diagrams and Statecharts to Analysable Petri Net Models, International workshop on software and performance, 35-45, ACM

BPMI.org (2002), Business Process Modeling Language (BPML), Version 1.0. *The Seventh Working Draft of BPML*.

Garrido, J. L. and Gea, M. (2002), A Coloured Petri Net Formalisation for a UML-based Notation Applied to Cooperative System Modelling, *Lecture Notes in Computer Science*, **2545**, 16-28.

Gou, H., Huang, B. and Ren, S. (2000), A UML and Petri net integrated modeling method for business processes in virtual enterprises, *Proceedings of the 2000 AAAI Symposium-Bringing Knowledge to Business Processes*, 142-144.

Hauschildt, D., Verbeek, H. M. W. and van der Aalst, W. M. P. (1997), Woflan: a Petri-net-based Workflow Analyzer*, Computing Schience Reports 97/12*, Eindhoven University of Technology, Eindhoven.

Hu, Z. and Shatz, S. M. (2004), Mapping UML Diagrams to a Petri Net Notation for System Simulation, *Software engineering and knowledge engineering*, 213-219, Knowledge Systems Institute Graduate School.

King, P. and Pooley, R. (2000), Derivation of Petri Net Performance Models from UML Specification of Communications Software, *Lecture Notes in Computer Science*, **1786**, 262-276.

Lin, C. P., Lin, Y. P. and Jeng, M. D. (2004), Design of Intelligent Manufacturing Systems by Using UML and Petri Net, *Proceedings of the 2004 International Conference n Networking, Sensing & Control*, 501-506, IEEE.

Murata, T. (1989), Petri Nets: Properties, Analysis, and Applications, *Proceedings of the IEEE*, 77(4), 541-580.

Shishkov, B. and Barjis, J. (2002), Modeling of e-Business Brokerage Systems Using UML and Petri Net, *IFIP world computer congress; Information systems the e-business challenge*, 61-76, Kluwer Academic.

UN/CEFACT (2003), ebXML Business Process Specification Schema, Version 1.10 Business Process Project Team of UN/CEFACT, October .

Van der Aalst, W. M. P. (1998), The Application of Petri Nets to Workflow Management, *Journal of Circuits, Systems and Computers*, **8**(1), 21-66.

Van der Aalst, W. M. P. (1999), Inter-organizational W-orkflows An approach based on message sequence charts and Petri nets, *Systems Analysis-Modeling-Simulation*, **34** (3), 335-367.

Van der Aalst, W. M. P. (2002), Inheritance of Inter-organizational Workflows: How to agree to disagree without loosing control? *Electronic Commerce Research*, 2(3), 195-231.

WfMC (2002), Workflow Process Definition Interface - XML Process Definition Language, Version 1.0. Document Number WFMC - TC 1025.

Yan, Z. and Gan, R. (2001), Modeling of SET Protocol Based on UML and Colored Petri Net, *Proceedings of the 2001 International Conference on Information Technology and Information Network*, **6**, 124-129, IEEE.

Sunho Kim

Professor, Department of Industrial & Management Engineering, Myongji University

Vice-Chair, e-Catalog TC, Integrated Forum for e-Commerce

Ph.D and MS, Pennsylvania State University

BS, Seoul National University

Research Topics: BPM, e/u-business standards