Integration of UML and Petri Net for the Process Modeling and Analysis in Workflow Applications

KWAN-HEE HAN*, SEOCK-KYU YOO**, BOHYUN KIM***

Department of Industrial & Systems Engineering, Gyeongsang National University, KOREA*, VMS Soutions Co., Ltd., Deajeon, KOREA**, Korea Institute of Industrial Technology, Ansan, KOREA***
hankh@gsnu.ac.kr*, skyoo@vms-solutions.com**, bhkim@kitech.re.kr***

Abstract: Recently, workflow management system has been widely accepted in most enterprises as one of enabling technologies for process innovation or BPR (Business Process Reengineering). It is quite essential to customer satisfaction and productivity enhancement that structural errors in workflow instances must be detected before their enactments. For a qualitative workflow analysis, this paper combines the strengths of UML activity diagram and Petri Net, proposes a mapping scheme from UML activity diagrams to Petri Nets, and finally demonstrates the transformation results by using two workflow scenarios. The transformed Petri Net is used for the reachability tree analysis for the detection of structural errors of workflows.

Key-Words: qualitative workflow analysis, UML activity diagram, Petri Net, reachability tree, workflow-net

1 Introduction

Most enterprises are struggling to change their existing business processes into agile, product-centric and customer-centric structures to survive the competitive, global business environment. Since a workflow is concerned with the automation of procedures where information and tasks are passed between participants according to a defined set of rules, workflow management systems are considered as a significant enabling technology for the successful implementation of the BPR (Business Process Reengineering) or process innovation projects for sustainable competitive advantage. To achieve a business goal by using a workflow management system in an enterprise, it is essential to rapidly execute correct workflow instances for their customers. Because the enactment of erroneous workflows causes the loss of customer’s royalties, the necessity of analyzing the potential errors of workflows in an efficient way prior to its execution is ever increasing.

Van der Aalst and van Hee differentiated between the analysis of the qualitative aspects and the quantitative aspects of workflows [9]. The former mainly concern the logical correctness of the defined process and can be analyzed typically by Petri Nets, whereas the quantitative aspects mainly concern the performance of the defined process, and can be analyzed typically by queuing theory or simulation method.

Widely used Petri Net techniques for qualitative workflow analysis are the reachability tree and the matrix equation. The reachability tree analysis method is quite suitable for business workflow because it can detect erroneous states of the workflow such as deadlock and livelock, and can trace the state transitions [1]. However, in spite of its analysis power, the Petri Net model is difficult to understand compared with other diagram models.

In recent days, UML (Unified Modeling Language) is being rapidly adopted as the de facto object-oriented modeling standard for information system development because of its graphical notation, which is readily understood, and a rich set of semantics for capturing key features of object-oriented systems. Moreover, it was presented that UML is also appropriate for business process modeling as well as information system modeling by virtue of its expressiveness, user-friendliness, and integration capability with information systems. Among the UML diagram, the activity diagram is most suitable for workflow modeling because of the following features: 1) it can describe the dynamic behaviors of business processes and events triggering the processes more easily. 2) It can represent various workflow routing types such as sequential routing, join, split, iteration, and parallel routing suggested by the WfMC (Workflow Management Coalition). However, the activity diagram has limited capability for analyzing a
business workflow because it suffers from a lack of precise semantics. The above mentioned characteristics of Petri Net and UML activity diagram indicates the possibility of a more useful workflow modeling and analysis tool by integrating the ease of modeling with analysis power.

The main objective of this paper is to propose an easy-to-use, qualitative workflow analysis method for enterprise practitioners by using a mapping scheme from UML activity diagrams to classical Petri Nets. The overall workflow analysis procedure in this paper proceeds as follows: 1) modeling of workflows by UML activity diagrams, 2) transforming workflows expressed in UML activity diagram into classical Petri Net by the proposed mapping scheme, 3) performing reachability tree analysis using the resulting Petri Net. By integrating the intuitive modeling capability of activity diagram and the mathematical formalism of Petri Net, workflow modelers in enterprises can model and analyze business processes more easily and accurately.

2. Related Works

UML Activity diagrams have been adopted as one of the major modeling tools for business processes. Dumas and ter Hofstede presented an evaluation of a UML activity diagram against a set of workflow patterns and provided insights into the relative strength and weakness of activity diagrams [2]. Eshuis and Wieringa pointed out that UML activity diagrams are not intended for workflow modeling [4]. For workflow modeling using activity diagrams, they defined semantics for activity diagram, and compared it with various Petri Net semantics. However, the elements of an activity diagram such as object, object flow, and swimlanes were not considered in the proposed semantics. Rodrigues pointed out that the activity diagram lacks a well-defined semantics for business processes, and proposed simple semantics by formalizing UML activity diagrams using finite state processes [8].

Among research about the transformation of an activity diagram into another format, Mantell proposed the mapping method of activity diagram into BPEL4WS (Business Process Execution Language for Web Services) format [7]. One of authors of this paper developed a business process modeling tool based on the transformation method of activity diagrams into XPDL (XML Process Definition Language) format file [5]. White compared UML activity diagrams to the BPMN Business Process Diagram (BPMI) for technical availability to represent the workflow patterns, as well as their readability [10]. Eshuis and Wieringa proposed a tool which translates an activity diagram into an input format for a model checker for the verification of workflow models [3]. Lopez-Grao et al. proposed the transformation procedure of an activity diagram into a LGSPN (Labeled Generalized Stochastic Petri Net) for the performance evaluation simulation of software [6]. They, however, excluded the modeling elements of activity diagram such as object, object flow, and swimlanes for the mapping from an activity diagram to a Petri Net. This transformation is not suitable for business workflows because business processes compared with software processes require more modeling semantics such as object flows and activity responsibilities.

3. The Mapping Scheme of a UML Activity Diagram to a Petri Net

For the complete workflow analysis, a workflow model must contain the core entities necessary for defining business processes. The entities of workflow process definition identified at the standard XPDL meta-model by WfMC (Workflow Management Coalition) and their corresponding elements of UML activity diagram are as follows:

1) Workflow process activity: work task performed in the course of workflow execution (UML: action state, signal receipt, signal sending), and an instantaneous activity that represents the initiation and termination of an activity (UML: start point, end point)
2) Activity set: sub-process that is called from another process, and which forms parts of the overall process (UML: sub-activity state)
3) Transition: flow from activity (or object) to activity (or object) (UML: transition, object flow) and routing control (UML: decision, sync state)
4) Workflow participant: resources that can act as the performer of the various activities (UML: swimlane)
5) Workflow relevant data & workflow applications: information object that is used in the execution of a specific task, and the invoked applications during the process execution (UML: object)

As well as UML activity diagram, Petri Net-based business process model must contain the entities identified at the XPDL meta-model. However, Petri Net primitives lack the semantics for representing business processes sufficiently. To provide the
semantics to Petri Net models, Petri Net building block structure called A2P block structure, which is a combination of Petri Net primitives, is proposed in this paper.

As depicted in Fig. 1, the A2P block structure is composed of the following elements:
- Block boundary: it represents the boundary of A2P block.
- Logic part: it consists of Petri Net primitives, and describes the behavior of a workflow activity. It starts with ‘transition’ and ends with ‘place’ (except the logic part of ‘start’ activity)
- Input and output ports: they are used for connecting one A2P block to other A2P blocks.

From the view of the Petri Net, the boundary and input/output port elements have no effect on the structure and properties of a Petri Net. Also, each A2P block has its own logic part according to its entity type of workflow definition.

For the mapping scheme, workflow process activity type is further classified to two groups by their properties: One is general work task type. The other type is an instantaneous activity that represents the initiation and termination of an activity (i.e. source/sink of workflow process). The general work task sub-type of UML activity diagram is represented by an A2P block having one input port and two output ports. One of the two output ports is for the transition of activity, and the other is for the flow of the resource availability token. The resource availability token is used for the explicit modeling of a swimlane element which shows the responsibility of a work task. The source/sink activity sub-type which require no responsibility for a work task, are mapped to an A2P block having one input port and one output port.

Since activity set type which groups several activities into a sub-process cannot be represented properly by a classical Petri Net, the sub-process symbol within the Petri Net structure is newly introduced. However, this symbol also has no influence upon the structure of Petri Net.

Transition type is classified to two groups: One represents the flow of activity or object. The other type represents routing control of workflow. The flow sub-type is transformed into the arc of a Petri Net. The routing control sub-type is further categorized to four groups for the modeling of various routing types suggested by WfMC. These routing sub-types are AND-join, OR-join, AND-split, and OR-split. Since each sub-type has a different flow control method, four A2P blocks corresponding to each routing control sub-types have been devised.

Workflow participant type, which represents resources that take the role of performer of the various activities, is mapped to Petri Net token(s) within a place. Workflow relevant data & applications type, which require no responsibility for a work task, are mapped to an A2P block having one input port and one output port. Fig. 2 shows a summary of the mapping scheme from UML activity diagram to A2P block for qualitative workflow analysis.
4. Transformation of Workflow Scenarios

To illustrate transformation results through the proposed mapping scheme, this section demonstrates the transformation results of two workflow scenarios. The case scenarios include various features in real business processes such as complex routings, the use of objects as input to or output from activity, responsibility of activity, and signal handling activity.

The first business process scenario which was presented in the paper of [9], describing the customer complaint handling workflow, is described in Fig. 3. The main features of the UML activity diagram for this case scenario, depicted in Fig. 4, are various routing types such as AND-split, AND-join, OR-split, and OR-join. Direct Petri Net modeling from the case scenario described in Fig. 3 is shown in Fig. 5. This Petri Net result represents ten atomic activities included in the case scenario as ten transitions (white symbols) and the addition of five transitions (gray symbols) for workflow routing control. Fig. 6 is the result of mapping the activity diagram of Fig. 4 to the Petri Net using the proposed mapping scheme. Since this case scenario doesn’t specify the responsibility of workflow participant, A2P building blocks of the resulting Petri Net model did not use the output port for the flow of the resource availability token. Compared with direct modeling, this result requires more transitions and places because of its automatic transformation nature, i.e., it uses ten transitions for atomic activities (white building blocks) and an additional ten A2P blocks (gray building blocks) for routing control.

Reachability tree analysis of two models results in the same workflow structure (reachability tree analysis is explained in more detail in Section 5). It reveals that the proposed mapping scheme is well-defined and applicable.

The second one, depicted in Fig. 7, is an order processing workflow which represents the work task sequences from order request to product delivery. The main features of this example are the object, which represents physical things or information, the object flow, and the swimlane showing the responsibility of each work task. The resulting Petri Net is depicted in Fig. 8, in which a swimlane is transformed into a token within a place, and an object is mapped to the workflow relevant data & applications type A2P block. Finally, the object flow is mapped to the arc of Petri Net.

5. Qualitative Workflow Analysis Using a Transformed Petri Net

For ensuring that a workflow will be successfully executed, a workflow which is defined in terms of a classical Petri Net should satisfy other requirements as well as the requirement of the WF-Net. These are: 1) for any case, a workflow will terminate eventually, and the moment a workflow terminates there is a token in place ‘end’ and all the other places are empty. Moreover, there should be no dead tasks; 2) it should be possible to execute an arbitrary task by following the appropriate route through the WF-Net. These two additional requirements correspond to the so-called “soundness” property. However, because these requirements don’t consider the swimlane element of an activity diagram (i.e. responsibility of an activity), another requirement is added newly in this paper that the number and the position of tokens representing the responsibility of the activity should not be changed before nor after the execution of the workflow.

Whether the transformed Petri Net is a WF-Net could be checked easily by examining the structure of the workflow. Additional requirements for the successful execution of a workflow could be checked by using the reachability tree analysis. The reachability tree analysis of the case scenario one (Fig. 6), depicted in Fig. 9, shows 64 total reachable nodes (i.e., states), and among these, there are three terminal nodes ( , , in Fig. 9).

The successful execution condition of a workflow needs only one terminal node ( ), in which there is only one token at ‘end’ place ‘p26’. Since this case has 3 terminal nodes, this workflow can be considered an erroneous workflow. Since the first terminal node ( ) has no token except ‘end’ place, it satisfies the soundness property. However, the fact that the second terminal node ( ) has no token except at two places (p11, p17) causes the conflict for the firing of transition ‘AJ1’ and leads to a deadlock situation. The last terminal node ( ) has no tokens except at two places (p15, p26). The
existence of a token at place ‘p26’ implies the successful execution of a workflow, but the additional token at place ‘p15’ violates the soundness property.

6. Conclusions and Future Work
It is quite essential to customer satisfaction and productivity enhancement that structural errors in workflow instances must be detected before their enactments. For analyzing workflows easily and rapidly, this paper integrates the ease of modeling of UML activity diagram with the mathematical formalism of Petri Net. For the mapping of an UML activity diagram into a Petri Net, the Petri Net building block called an A2P block is proposed. In accordance with the mapping scheme, twelve modeling elements of UML activity diagram are
1) First, the complaint is registered by executing the task ‘register’.
2) In parallel, a questionnaire is sent to the complainant (task ‘send_form’) and the complaint is evaluated (task ‘evaluate’).
3) If the customer returns the questionnaire within two weeks, the task ‘process_form’ is executed.
   If the questionnaire is not returned within two weeks, the result of the questionnaire is discarded (task ‘time_out’).
4) Based on the result of the evaluation, the complaint is processed or not.
   The actual processing of the complaint (task ‘process_complaint’) is delayed until the form has been processed.
5) The processing of the complaint is checked via task ‘check_process’. According to the result of ‘check_process’, re-processing may be done again.
6) Finally, task ‘archive’ is executed if the condition (the questionnaire has been processed or a time-out has occurred) and complaint (the complaint has been evaluated or process has been checked) are satisfied.

Fig. 3 Workflow scenario-1 for customer complaint handling

Fig. 4 UML activity diagram for workflow scenario-1

Fig. 5 Petri Net result by direct transformation from scenario-1

Fig. 6 Petri Net result by proposed mapping scheme

Fig. 7 UML activity diagram for workflow scenario-2
mapped into their corresponding A2P blocks based on their entity types. For the validation of the proposed mapping scheme, the transformation results have been demonstrated by using 2 workflow scenarios. The transformed Petri Net is used for the reachability tree analysis for the detection of the structural errors of workflows.

Whereas this paper emphasizes on the qualitative workflow analysis, further research for the quantitative workflow analysis such as the performance evaluation is still needed, in which a transformation scheme from UML activity diagram into high-level Petri Net should be provided, which can handle colored tokens, time and stochastic properties of workflows.

References

Acknowledgements
This research was conducted as a part of project for the development of digital based real-time adaptive manufacturing system platform sponsored by Ministry of Knowledge Economy and VMS Solutions co., ltd., and also was financially supported by Ministry of Education, Science and Technology and Korea Institute for
Advancement of Technology through the Human Resource Training Project for Regional Innovation