Mapping Workflow Management Concepts onto Petri nets for the Modelling of the Process Dimension

Kamalakshi N[#1], H Naganna[*2]

[#1,2]*Dept of Computer Science Sapthagiri College of Engg Bangalore, India*

[1] `kamalnags@yahoo.co.in,` [2]`naganna_h@hotmail.com`

***Abstract*— Most work flow management systems and methodologies to support workflow management separate the modeling of the workflow process from the modeling of the structure of the organization and the resources within the organization. The distinction between these two aspects is illustrated, which relates the process dimension. There are many reasons for decoupling these two dimensions when specifying a workflow. The complexity is reduced, reuse is stimulated, and it is possible to modify a process without changing the organizational model (and vice-versa). In this paper we show that Petri nets are suitable for the modeling of the process dimension.**

***Keywords*— Iteration, sequential, process dimension, workflow , Conditional routing**

I. Workflow Process Definition

Workflow management mapping meets these objectives by showing how the “stuff” of any organization’s work, whether tangibles liked manufactured materials, or intangibles like information or changes in patients’ states of health, moves through that organization. In today’s economy, the “material” that most organizations work on is information, and for the time being we will also refer to this as material, even though it is not. Any of the things that organizations do to accomplish their objectives involve both the processing of their materials and the processing of information about what to do with them. To begin, at the beginning, then, we need to have some basic terms and concepts to describe what these things are, and what is going on in the workflow we want to map.Workflow Mapping and Analysis (WFMA) is a graphic method of completely describing the materials and information flows necessary to accomplish one or more specific objectives of work, in their correct sequence, in a single job, a process, an organizational unit, or an entire organization [1].

In the process dimension, it is specified which tasks need to be executed and in what order. Modeling a workflow process definition in terms of a Petri net is rather straightforward: tasks are modeled by transitions, conditions are modeled by places, and cases are modeled by tokens. To illustrate the mapping of workflow management concepts onto Petri nets we consider the processing of complaints. First the complaint is registered (task register), then in parallel a questionnaire is sent to the complainant (task send questionaire) and the complaint is evaluated (task evaluate). If the complainant returns the questionnaire within two weeks, the task process questionnaire is executed. If the questionnaire is not Returned within two weeks, the result of the questionnaire is discarded (task time out). 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 questionnaire is processed or a time-out has occurred. The processing of the complaint is checked via task check processing. Finally, task archive is executed

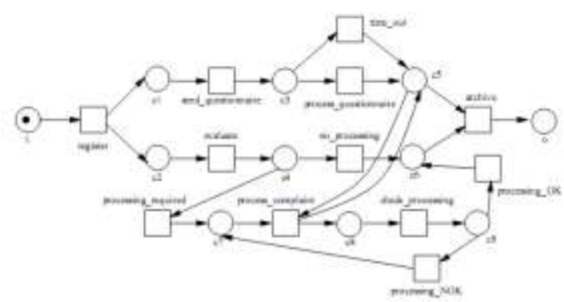

Fig 1: Tokens have a case identifier which allows for the separation of cases.

Figure 1 shows a work flow process definition for the processing of complaints specified in terms of a Petri net. The tasks register, send questionnaire, evaluate, process questionnaire, time out, process complaint, check processing. The transitions no processing and processing required have been added for similar reasons. To model the states between tasks, conditions have been added. Each condition is modeled by a place. For example, place c2 corresponds to the condition ‘ready to evaluate complaint’. Condition c5 is true (i.e. place c5 contains a token) if the questionnaire has been processed or a time-out has occurred.

Note that c5 is a prerequisite for the task archive and the task process complaint. Condition i is the start condition and condition o is the end condition. The workflow process definition showed in Figure 5 models the life-cycle of a single case. In general, there are many cases which are handled according to the same workflow process definition. Each of these cases corresponds to one or more tokens. If tokens of multiple cases reside in the same Petri net, then these tokens may get mixed. For example, transition archive may consume two tokens which correspond to different cases. Clearly, this is undesirable. There are two ways to solve this problem. First of all, it is possible to use a high-level Petri net where each token has a value (color) which contains information about the identity of the corresponding case (case identifier). Transitions are not allowed to fire if the case identifiers of the tokens to be consumed do not match, i.e., a precondition is used which inspects and compares token values. Another way to solve this problem is the following. Each case corresponds to a unique instance of the Petri net. If there are n cases, then there are **n** instances of the Petri net. One can think of such an instance as a layer. If these layers are put on top of each other, it is possible to see the cases in the same diagram. The latter is interesting from a management point of view, because one gets an overview of the state of the workflow. For example, if a place contains a lot of tokens, this might indicate a bottleneck. In the remainder of this paper, we consider Petri nets which describe the life-cycle of one case in isolation.

II. ROUTING CONSTRUCTS

In the process dimension, building blocks such as the AND-split, AND-join, OR-split and OR-join are used to model sequential, conditional, parallel, and iterative routing (WfMC [WFM96]). Clearly, a WF-net can be used to specify the routing of cases. In this Section, four types of routing have been identified: sequential, parallel, conditional, and iteration.

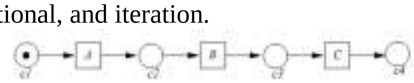

Figure 2: Sequential Routing

Sequential routing is used to deal with causal relationships between tasks. Consider two tasks A and B. If task B is executed after the completion of task A, then A and B are executed sequentially. Figure 2 shows that sequential routing can be modeled by adding places. Place c2 models the causal relationship between task A and task B, i.e., place c2 represents a postcondition for task A and a precondition for task B. Place c3 models the causal relationship between task B and task C.

Parallel routing is used in situations where the order of execution is less strict. For example, two tasks B and C need to be executed but the order of execution is arbitrary. To model such a parallel routing, two building blocks are used: (1) the AND-split and (2) the AND-join. Figure 4 shows that both building blocks can be modeled by ordinary transitions. One can think of these transitions as control tasks that have been added for routing purposes. However, a normal task can also act as an AND-split and/or an AND-join. The execution of AND-split A enables both task B and task C. AND-join D is enabled after execution both B and C, i.e., D is used to synchronize two subflows. As a result, task B and task C are executed in parallel. This means that B and C are executed in arbitrary order. It is even possible that the execution of B and C coincides, i.e., for one case, tasks overlap in time. In several workflow management systems, it is not possible to execute two tasks for the same case at the same time. In this case, we use the term semi parallel. If semi-parallel routing is used in Figure 4, then B is executed before C or vice versa, i.e., interleaving semantics are used to interpret the workflow process definition. Sometimes, semi-parallel routing is used to avoid serious technical problems. For example, for each case, applications and/or tasks in the workflow management system share data. In this case, the locking mechanism in the database management system may prevent tasks to be executed concurrently for the same case.

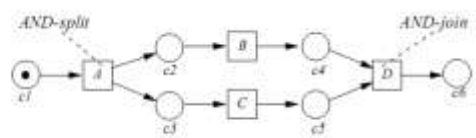

Figure 3. Parallel routing

AND-split and (2) the AND-join. Figure 4 shows that both building blocks can be modeled by ordinary transitions. One can think of these transitions as control tasks that have been added for routing purposes.However, a normal task can also act as an AND-split and/or an AND-join. The execution of AND-split A enables both task B and task C. AND-join D is enabled after execution both B and C, i.e., D is used to synchronize two subflows. As a result, task B and task C are executed in parallel. This means that B and C are executed in arbitrary order. It is even possible that the execution of B and C coincides, i.e., for one case, tasks overlap in time. In several workflow management systems, it is not possible to execute two tasks for the same case at the same time. In this case, we use
the term semi-parallel. If semi-parallel routing is used in Figure 4, then B is executed before C or vice versa, i.e., interleaving semantics are used to interpret the workflow process definition. Sometimes, semi-parallel routing is used to avoid serious technical problems. For example, for each case, applications and/or tasks in the workflow management system share data. In this case, the locking mechanism in the database management system may prevent tasks to be executed concurrently for the same case[2].

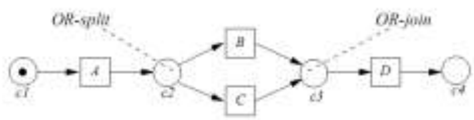

Figure 4. Conditional routing

Conditional routing is used to allow for a routing which may vary between cases. In this way, the routing of a case may depend on the workflow attributes of a case, the behavior of the environment, or the workload of the organization. To model a choice between two of more alternatives, two building blocks are used: (1) the OR-split and (2) the OR-join (in both cases an exclusive OR). An OR-split can be modelled Conditional routing is used to allow for a routing which may vary between cases. In this way, the routing of a case may depend on the workflow attributes of a case, the behavior of the environment, or the workload of the organization. To model a choice between two of more alternatives, two building blocks are used: (1) the OR-split and (2) the OR-join (in both cases an exclusive OR). An OR-split can be modeled by a place with multiple outgoing arcs; an OR-join is modeled by a place with multiple ingoing arcs. Figure 4 shows the situation where task A is followed by either task B or task C, i.e., a choice is made between B and C. The execution of one of these two tasks is followed by the execution of task D. Place c2 is a precondition for both B and C However, just one of these two tasks will be executed for the case in place c1.If c2 contains a token; a non-deterministic choice is made between B and C. However, the choice between alternatives often depends on workflow attributes. If the choice is based on workflow attributes, it is a deterministic choice. For example, the routing of an insurance claim may depend on the compensation costs. Therefore, a workflow attribute is used to take into account the compensation costs. If these costs exceed a certain amount, additional checks are necessary. The routing of a traffic violation may depend on the type of the traffic violation represented by the corresponding workflow attribute. There are two ways to model a choice based on workflow attributes. (Recall that the extension with color is used to model workflow attributes.) We can use the construct shown in Figure 4 and add a precondition (i.e. an additional enabling requirement based on one or more work-flow attributes) to each of the tasks such that either B or C is enabled if c2 contains a token. Another way to model a deterministic choice between B and C is shown in Figure 5. Transition A has two output places c2 and c3. Transition A produces either a token in c2 or c3. The choice between c2 and c3 is based on workflow attribute **x**. If **x** is positive, task B will be executed, otherwise task C. A special symbol is used to denote the fact that task A is an OR-split (exclusive OR). Note that the nondeterministic choice in Figure 4 differs from the choice in Figure 5 with respect to the moment of choice. In Figure 5 the choice is made the moment task A is completed, in Figure 4 the choice is made the moment B or C is executed. We will use the term explicit OR-split for a choice based on workflow attributes. The term implicit OR-split is reserved for the situation where the moment of choice is as late as possible. For workflow modeling it is of the utmost importance to distinguish between implicit and explicit OR-splits denote the fact that a task is an AND-split, an AND-join, and/or an OR-join. The AND-split and the AND-join correspond to the normal behavior of a transition in a classical Petri net. The implicit OR-split and OR-join are modeled by places. The explicit OR-split is modeled by a transition which produces one token in one of its output places, i.e., it is an exclusive OR (XOR). The place is selected on the basis of the workflow attributes. The explicit OR-join is modeled by a transition which is enabled if one of the input places contains a token. In general, the explicit OR-join can be modeled by an implicit OR-join (i.e. a place). There is no compelling need to distinguish between implicit and explicit OR-joins. Therefore, we will avoid the use of explicit OR-joins in this paper. However, the distinction between the implicit OR-split and the explicit OR-split is of practical relevance and will be discussed in more detail when the concept of triggering is introduced

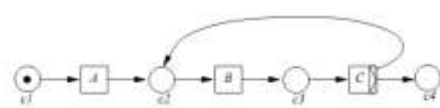

Figure 5: Iteration routing

III. Conclusion

In this paper we introduced a number of workflow concepts and showed how these concepts can be mapped onto Petri nets. The four routing constructs identified by the WfMC have been mapped onto WF-nets.

References

[1] John L. Kmetz "A Brief Introduction to Workflow Mapping and Analysis", 2010

[2] W.M.P. van der Aalst "The Application of Petri Nets to Workflow Management". The Journal of Circuits, Systems and Computers, 8(1):21–66, 1998.

[3] "Research on Workflow Patterns based on Petri nets "Liang Zhang IEEE Conference on Robotics, Automation and Mechatronics, June 2006

[4] P. Chrzastowski-Wachtel." Top-down Petri Net Based Approach to Dynamic Workflow Modeling University of New South Wales, Sydney, 2002

[5] J. Dehnert and P. Rittgen. "Relaxed Soundness of Business Processes." In K.R. Dittrich,A. Geppert, and M.C. Norrie, editors, Proceedings of the 13th International Conference on Advanced Information Systems Engineering (CAiSE'01), volume 2068 of Lecture Notes in computer Science, pages 157–170. Springer-Verlag, Berlin, 2001

[6] W.M.P. van der Aalst and K.M. van Hee. "Workflow Management: Models, Methods, and Systems". MIT press, Cambridge, MA, 2002.