Integrating Process and Organization Models of Collaborations through Object Petri Nets

Kamyar Sarshar[1], Thomas Theling[1], Peter Loos[1], Mirko Jerrentrup[2]

Institute for Information Systems at the German DFKI
Stuhlsatzenhausweg, Geb. 43.8
D-66123 Saarbruecken, Germany
{sarshar|theling|loos}@iwi.uni-sb.de

Interactive Software Solutions
Saarterrassen, Hochstr. 63
D-66115 Saarbruecken, Germany
jerrentrup@interactive-software.de

Abstract: The management of virtual enterprises needs extended and integrated approaches of business modeling. While most formal approaches to business process modeling consider only the control-flow perspective, it is essential in an inter-organizational context to link tasks with the enterprise responsible for their execution. This paper presents the concept of the transforming BPMN-conform XML representation of process models and a proprietary OMN-XML representation of organization models into a special type of object Petri nets called Reference net. The benefit of our approach is that by using Reference nets, the control-flow and the inter-organizational perspective of a business process can be integrated into a unique formalism ready to by analyzed and simulated by appropriate Petri net tools like RENEW. After introducing the transformation concept its application will be demonstrated by an example.

1 Introduction

To manage virtual enterprises and the collaboration of businesses, existing concepts for business process management need to be adapted and extended. A theoretical framework for collaborations is the theory of transaction costs based on the work of Coase [Co38; Wi95]. Institutions of the market like corporations, governmental organizations, or legal conditions are analyzed in this theory. Rights, goods, or outputs are transferred (delivered) between organizations by transactions, which are based either on contracts or hierarchies. Therefore the modeling notation used in our approach must be able to depict these contracts and deliveries resp. hierarchies in an adequate way.

Within the project ArKoS[1], a core team of eight universities and companies have determined requirements for the design, implementation and evaluation of an architecture appropriate for collaborative businesses. An essential component of the established architecture [Th05a; Th05b] is a distributed repository, which provides collaboration-wide process and organizational models. A challenge is the interoperability of different modeling notations within the architecture: On the one hand, collaborating corporations may use different modeling tools applying different modeling languages. On the other hand, collaboration-wide and enterprise-intern models are notated in different modeling languages. For this reason, all collaboration-wide required models are transformed and stored in repository-wide unique data formats. By using converters, different modeling software and modeling notations can be integrated. The data format applied to the repository is a BPMN-conform [Wh04] XML format for business processes, while inter-organization models are stored in a proprietary XML format, called OMN-XML. Both formats have been defined within the project ArKoS and are described below.

Another task of the established architecture is to support management and controlling of collaborations. In early collaboration phases, organizational and process models can be used to simulate the collaboration's behavior a priori. A formal notation of organizational and process models is an Object Petri net [Va04]. Object Petri nets allow to formalize and to integrate process and organizational models within *one* notation which can be used for analysis as well as simulation by according Petri net tools like RENEW [Ku04]. The major advantage of our approach is that it has a broader view on business processes by formalizing not only their control-flow but also considering their organization perspective. This enables to consider aspects like capacity utilizations, to estimate lead-times by process simulation, or to deliver useful data for improving collaborations' processes and organizational structures.

This paper presents the concept on how to transform the repository-wide XML-formats for semi-formal process and organization models into two interrelated PNML files (Petri Net Markup Language) [Bi03] representing a Reference net [Ku02] which is a special type of object Petri nets [Va04].

The remainder of this paper is organized as follows: After this introduction, the next section will give an overview of related work and assorts used modeling languages of the established architecture. Chapter 3 introduces briefly the object Petri nets and Reference nets. Chapter 4 presents the developed transformation concept, while chapter 5 shows an example for the conversion by a converter implemented based on the introduced concept. Chapter 6 gives a summary and an overview of future work.

---

[1] http://www.arkos.info

2 Previous Research

The concepts of virtual enterprises and collaborative business [Wi95] are discussed on the basis of several economical theories like transaction costs [Wi95; Co38], the market-based view proposing strategic groups to evaluate cooperations' effects on the market [CP77], or the resource-based view emphasizing competencies of corporations [HP90].

In order to setup and maintain collaboration between enterprises to jointly produce goods and services, it is essential to represent workflow and business processes by appropriate notations [Aa02; EM00; RKC98]. For the purpose of business process modeling, several notations have been discussed within literature. Petri nets, originating from the early work of Carl Adam Petri [Pe62] have successfully been applied to process modeling, analysis and simulation by several authors [AHH94; Zi77]. A further approach to process modeling is BPMN (Business Process Modeling Notation) [Wh04], which is a graphical notation and defines activities as well as control flows to visualize business process operations.

However, the focus of business process modeling is the representation of the execution order of activities which is described through constructs for sequence, choice, parallelism or synchronization. When business processes are jointly performed in an inter-organizational context, it is also essential to link tasks with the enterprise responsible for its performance. Numerous authors have investigated such an integrated approach to process and organization models. Zur Muehlen [Mu04] proposes a workflow lifecycle considering an organizational (or resource) perspective in each stage. Van der Aalst describes an organizational model in UML and converts it into an XML DTD in order to integrate it with XML based representations of workflows [AKV03].

The idea of mapping system and object nets to the organization and process perspective of business processes has been sketched by a minor case study of the Dutch Justice department [Va98]. Van der Aalst generalizes the idea and introduces based on Reference nets a conceptual framework for inter-organizational workflow enactment by which different perspectives of workflows including control-flow, resource, data, task and operation can be represented by reference nets [Aa99].

An alternative approach is to use elementary Petri nets like Place/Transition nets for the purpose of representing the control-flow and the organizational perspective of a business process [AH02]. However, using elementary Petri nets for business process modeling purposes would lead to complex models with reduced readability. This has led to prefer high level petri nets for business process modeling purposes. And this is even more the case when in addition to control-flow the organizational perspective is considered. The advantage of the object Petri net approach is to have a formalism which integrates both perspectives while each perspective is represented by a distinct Petri net (system and object net). These are integrated by their dynamic behavior.

3 Object Petri Net Approach

Petri net is a generic term for a number of modeling techniques, graphical representations and notational conventions that are all based on the concept of net formalism introduced by Carl Adam Petri [Pe62]. Since their introduction Petri nets have been extensively investigated within the scientific community, whereby different extensions and applications were introduced [RR98]. A significant difference between the Petri net types is their token concept. Within elementary Petri nets, undistinguishable black tokens represent the availability of pre-condition of transitions, while at high-level Petri nets, tokens represent passive data structures which are transformed by transitions. The use of structured tokens permits the representation of more complex systems, while they are still passive and have no dynamic behavior. However, with the emergence of object-orientation some research has been conducted to combine Petri net models with the object-oriented paradigm [GV03]. The object Petri nets approach adds dynamic behavior to tokens by defining them as Petri nets again [Va04]. The approach has its origins in works describing the execution of task systems in systems of functional units [JV87]. It allows a multi-level modeling technique whereby one or more so called object nets move through a system net as ordinary tokens.

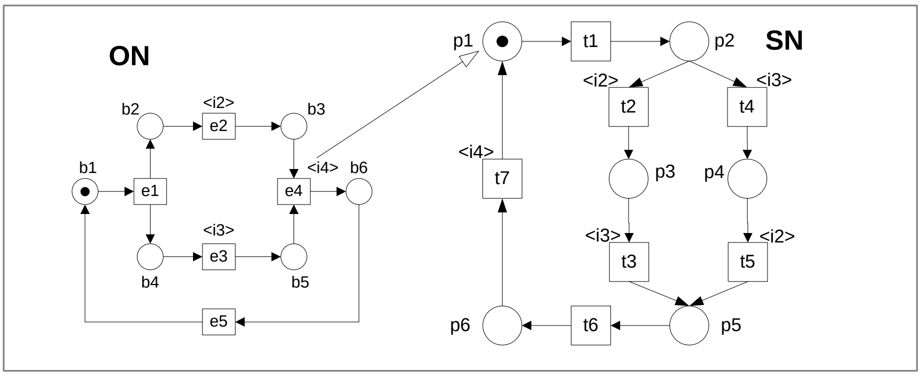


Figure 1: Object Petri Net Example [Va98]

Figure 1 gives an example of three possible interaction relations between the system and object nets of object Petri nets [Va98]. The object net (ON) illustrated on the left is located at the place p1 of the system net (SN). A label <i_n> synchronizes steps between the respective transitions of the object net and system net; a missing label indicates mutually autonomous steps. Since there is no such label at transition e1 and t1 an object autonomous step of the object net and a system autonomous step of the system net is possible. After theses steps, object and system net have reached a point where an interaction between the two levels at e2 and t2 as well as e3 and t4 are possible next steps.

A central characteristic of the approach is the distinction between reference and value semantics. The reference semantics restricts the system net to refer to identical copies of object nets. The dynamic behavior of the reference semantic is formalized by the bi-marking. On this basis, Reference nets have been developed and implemented by the

RENEW-Tool [Ku04]. The value semantics which allow instances of an object net to be independent copies use the p-marking in order to execute consistently. For more details on the object Petri net approach we refer to the referred literature.

To be able to simulate the object and system nets in a simulation engine, we have decided to apply the Reference nets and use the RENEW-Tool which has an import interface for PNML files. We interpret the elements of the Reference nets in the following way:

The object net represents a process. Transitions of the object net correspond to tasks and the states the conditions, which have to be fulfilled in order to execute a task. Token of the object net illustrate the case (process object) that is transformed by the execution of the process [Aa98]. The system net represents the organization structure. In this context, each place of the system net is interpreted as a single enterprise connected with other enterprises through contract and delivery relations (transitions of the system net). The object autonomous step stands for the execution of the process within an organization. A system autonomous step transports the process (object net) from one organization to another without changing the state of the object net. By an interaction, a task of the process is executed while it is transported to another organization. In other words, an interaction causes transitions of the object and system nets to fire synchronously. The synchronous execution of Reference nets is realized by labeling the according transitions with *downlinks* in case of the system net and corresponding *uplinks* at the transitions of the object net.

4 Transformation Concept

4.1 Conceptual Framework

Within the ArKoS-architecture, inter-organizational process and organization models are stored in a common repository. Figure 2 illustrates different notations and their application within the ArKoS-project. It distinguishes between different conceptual levels. At a visualization level the ARIS-Toolset [Da01] is used for modeling inter-organization structures of the dependencies of collaborating enterprises as well as the processes performed jointly by them. In order to prevent the repository from being dependent from the ARIS-Toolset and the Event-driven Process Chain (EPC), process models are exported into BPMN-XML, while organization models are exported into OMN-XML. These formats which are introduced in detail in section 4.2 are used to store the models in the common, collaboration-wide available repository. The transformation concept we introduce in this contribution uses these two XML files as input and produces two interrelated PNML files representing an Reference net, which will be introduced briefly within the next section.

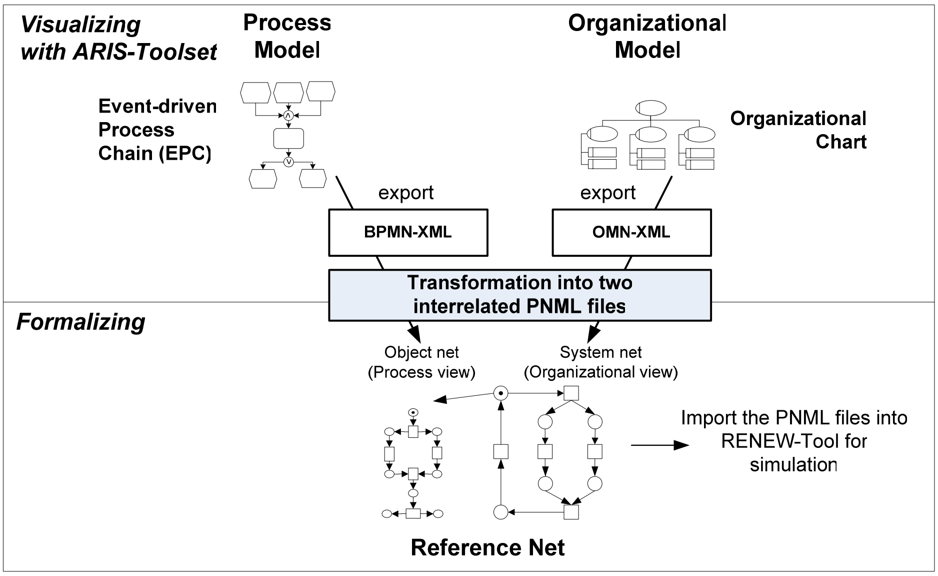
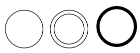


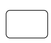
Figure 2: Selected Modeling Notations used in ArKoS

4.2 XML Representation of the Input Files

Table 1 illustrates the XML-representation of BPMN elements which is the format for process models. The included elements reflect basically the elements needed by the EPC since this was the notation used to model business processes with the ARIS-Toolset. Hence, from BPMN only events, activities, AND gateway and XOR gateway have an XML-representation. The only exception we made so far was to exclude the OR-connector because it has a non-local behavior [ADK02] and we have not developed a proper solution for its transformation yet.

BPMN	Graphical Representation	BPMN-XML-Representation

Element
EVENT
Subtypes
START,
INTERMEDIATE,
END
		`<ControlFlowObjectDefinition`
		`  controlflow-object-type="EVENT"`
		`  controlflow-object-definition-id="E1"`
		`  controlflow-object-subtype=`
		`    "start|intermediate|end"`
		`  artifact-idlist="..."`
		`  lane-idlist="..."`
		`  linked-process-id="...">`
		`</ControlFlowObjectDefinition>`

Element
ACTIVITY
		`<ControlFlowObjectDefinition`
		`  controlflow-object-type="ACTIVITY"`
		`  controlflow-object-definition-id="F2"`
		`  controlflow-object-subtype="task"`
		`  artifact-idlist="..."`
		`  lane-idlist="..."`

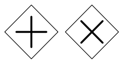
		linked-process-id="..."> </ControlFlowObjectDefinition>
Element **GATEWAY** Subtypes *AND, XOR,*		<ControlFlowObjectDefinition controlflow-object-type="GATEWAY" controlflow-object-definition-id="AND1" controlflow-object-subtype="AND|XOR" artifact-idlist="..." artifact-dirlist="..." lane-idlist="..." linked-process-id="..."> </ControlFlowObjectDefinition>

Table 1: XML-Representation of the Input BMPN Elements for the Process View

Equivalent to the processes which are modeled within ARIS-Toolset and exported into the BPMN-XML explained above, a proprietary OMN-XML is used to represent the organization model of ARIS-Toolset. The elements for representing the organization are the organization unit and three relations between them: the hierarchy, the delivery and the contract relation. The hierarchy relation would be sufficient as long as the organization units within one organization have to be represented. However, the relation between whole organizations can also be organized as a hierarchy or alternatively as a network. This was the idea behind adding contract and delivery relations to the OMN-XML in order to be able to represent network organization. With these two additional forms of relations we can indicate where the execution of a business process is passed from one organization to another.

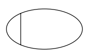
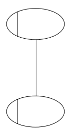
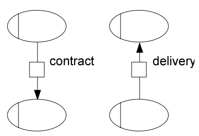
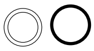
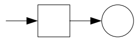
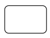
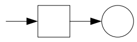
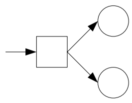
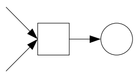
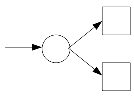
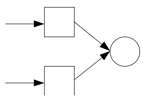
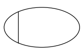
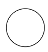
OMN	Graphical Representation	OMN-XML-Representation
Organization unit		<OMN-ObjectDefinition id="OE1" type="ORGUNIT" subtype="CONSORTIUM"> </OMN-ObjectDefinition>
Hierarchy between organizations		<OMN-EdgeDefinition id="E1" type="ORG" source-object-definition-id="OE1" target-object-definition-id="OE2"> </OMN-EdgeDefinition>
Contract resp. delivery relation between organizations		<OMN-EdgeDefinition id="E2" type="ORG" subtype="CONTRACT|DELIVERY" source-object-definition-id="OE1" target-object-definition-id="OE2"> </OMN-EdgeDefinition>

Table 2: XML-Representation of the Input OMN Elements for the Organization View

4.3 Transformation Rules

Table 3 and 4 intend to illustrate the transformation concept which is based on the semantics of the elements. Since a graphical representation of the transformation is easier to read then XML-code, we present the transformation rules in table 3 graphically. The XML-representation of the input elements can be derived easily by table 1 and 2. For output elements see table 4 where the PNML representation of each Petri net element is demonstrated. Table 4 also includes the characteristic of Reference nets of adding uplinks and downlinks to transitions in order to enable them to fire synchronously.

BPMN-XML		**PNML**	
Element **EVENT**			
Subtype *START*			
Subtypes *INTERMEDIATE, END*			
Element **ACTIVITY**			
Element **GATEWAY**			
Subtype *AND*		AND-Split	
		AND-Join	
Subtype *XOR*		XOR-Split	
		XOR-Join	
OMN-XML		**PNML**	
Organization			

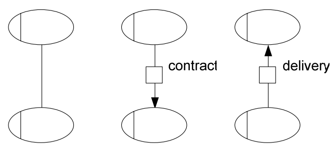
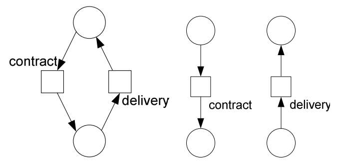
Relations between organizations		

Table 3: Converting Rules for the Process and Organization Perspective

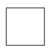
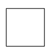
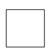
Petri Net Element	Graphical Representation	PNML-Representation
Unmarked place		<place id="..."> <text>...</text> </place>
Marked place		<place id="..."> <initialMarking> <text>...</text> </initialMarking> </place>
Transition		<transition id="..."> <name> <text>...</text> </name> </transition>
Arc	→	<arc id="E1_F1" source="E1" target="F1" />
Refence Net Chracteristic	**Graphical Representation**	**PNML-Representation**
Uplink	 *uplink-name*	<transition id="..."> <name> <text>...</text> </name> **<uplink>** **<text>...</text>** **</uplink>** </transition>
Downlink	 *downlink-name*	<transition id="..."> <name> <text>...</text> </name> **<downlink>** **<text>...</text>** **</downlink>** </transition>

Table 4: PNML-Representation of Petri Net Elements

5 Showcase

5.1 Visualizing Level

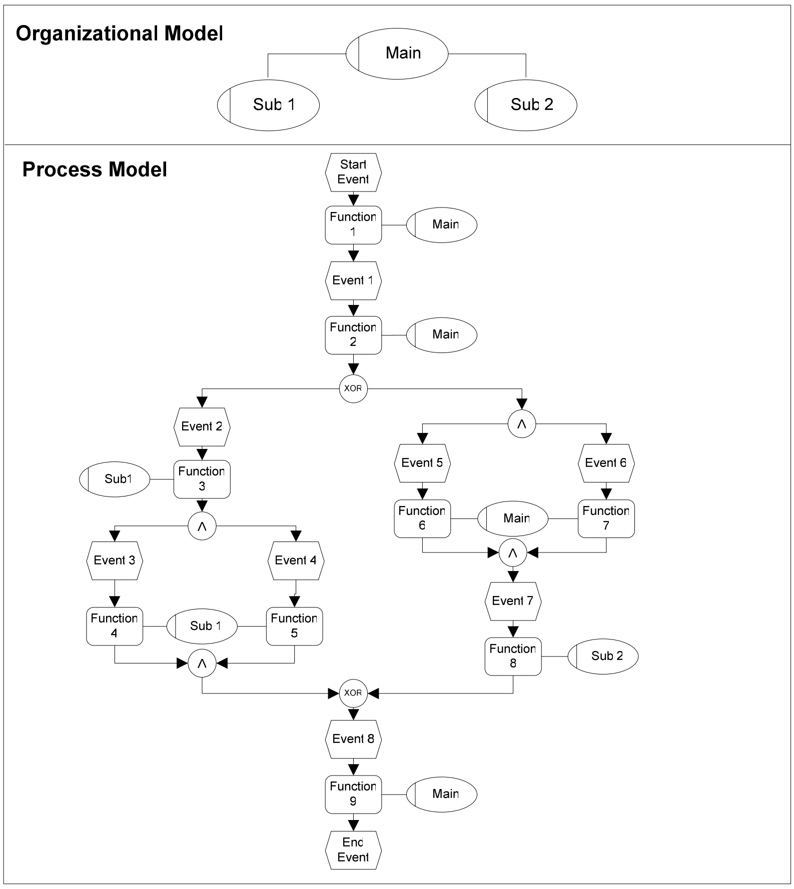


Figure 3: Example for Visualization of an Inter-Organizational Process and Organization Model

To demonstrate our concept we designed a business process concerning three companies, which are hierarchically organized. The organization model of the collaboration is depicted in the upper part of figure 3. The process is depicted as an EPC (cf. lower part of figure 3) and describes an abstract process. In this example a company *Main* coordinates the processes executed by *Sub 1* and *Sub 2*. After a Start Event, *Function 1* and *Function 2* are executed by *Main*. Afterwards, the process splits into two alternative parts: either *Function 3* is performed by *Sub1* or *Function 6* and *Function 7* are concurrently executed by *Main*. After *Function 3* in the left part, *Sub 1* performs also *Function 4* and *Function 5* concurrently. In the right path *Function 6* and *Function 7* are synchronized. Afterwards *Function 8* is performed by *Sub 2*. In any case, the process ends with *Function 9* performed by *Main*, and the *End Event*.

After visualizing, the process model as well as the organization model are exported into BPMN-XML and OMN-XML and stored in the repository. To implement the transformation rules introduced above, we have developed a JAVA-based converter which uses these files as input and creates two interrelated output files.

5.2 Interrelating the two PNML output files

The interrelation between the two PNML files is based on uplinks and downlinks indicating which transitions of the two models have to fire synchronously within object and system net. Hence, we first have to adopt a convention for naming the labels of such synchronous channels connecting the states of the system net and use them appropriately for the object net. This is illustrated by an example in figure 4. *Main*, *Sub1* and *Sub2* are connected by transition. Each transition of the system net has a downlink. The naming convention is that the label illustrates to which organization unit the transition transports the process when it is fired. For instance the label of the transition passing a process from *Main* to *Sub*" is *x:SUB_1()* (the syntax is based on the RENEW-Tool). In order to interlink the dynamic behavior of process and organization models we have to add these labels to appropriate transitions of the object net. To detect where an interlink is needed, every two successive functions of the process model represented by the object net have to be compared on their organization assignments. If they differ, an uplink has to be assigned to the event lying in between the two functions. This has lead to add the uplink *:SUB1()* to the *Event 2* which is located between *function 2* and *function 3* of the output PNML file representing the object net as shown in table 5.

Object Net (Process View)	System Net (Organizational View)

<pnml>
  <net type="RefNet" id="1">
    <name>
      <text>objectnet</text>
    </name>
    […]
    <transition id="Event 2">
      <name>
        <text>Event 2</text>
      </name>
      <uplink>
        <text>:SUB1()</text>
      </uplink>
    </transition>
    […]
  </net>
</pnml>
	<pnml>
	  <net type="RefNet" id="1">
	    <name>
	      <text>systemnet</text>
	    </name>
	    […]
	    <transition id="MAIN_SUB1">
	      <name>
	        <text>contract</text>
	      </name>
	      <downlink>
	        <text>x:SUB1()</text>
	      </downlink>
	    </transition>
	    […]
	  </net>
	</pnml>

Table 5: Interrelating the Output PNML Files

5.3 Graphical Representation of the output PNML files

Figure 4 depicts the graphical representation of both PNML files after importing and manually adjusting their representation in the RENEW-Tool.

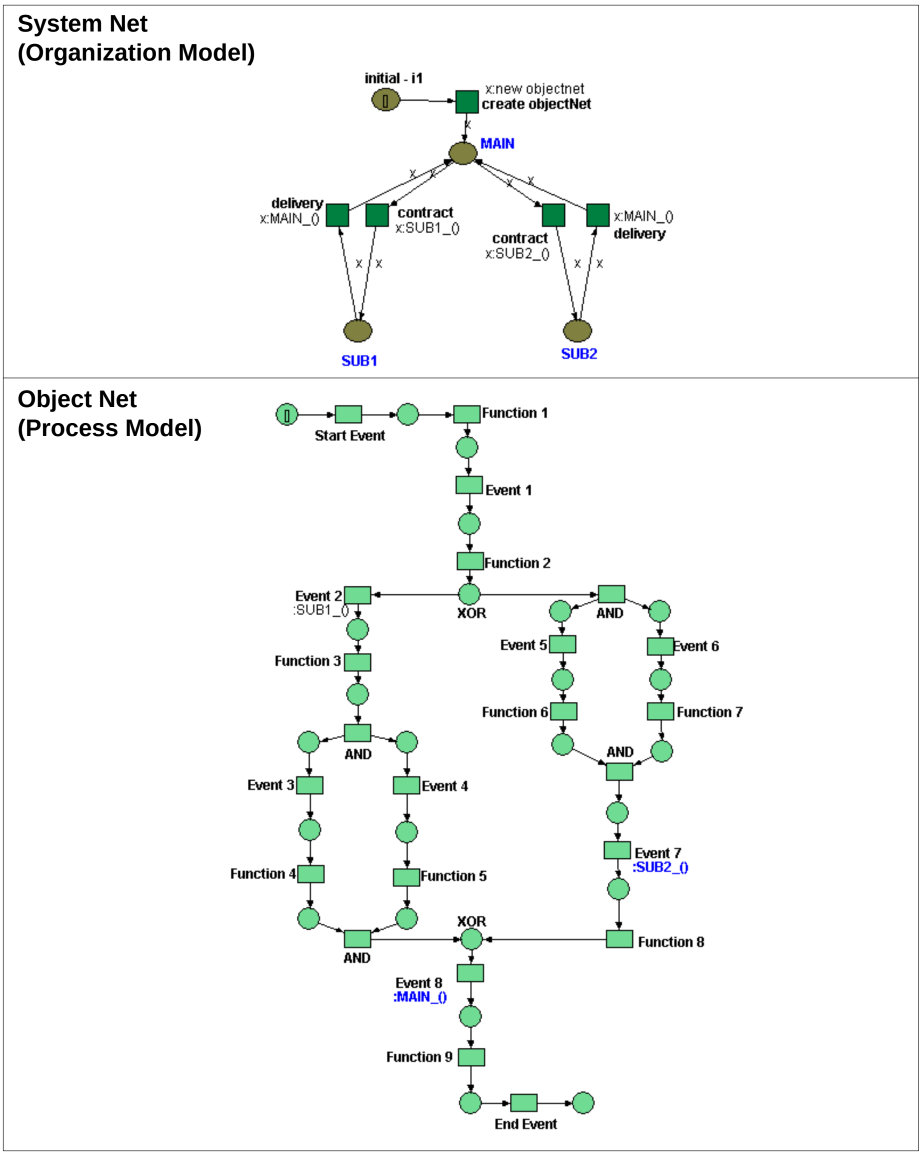


Figure 4: Transformation of the Models in Figure 3 into an Object Petri Net

The upper part represents the system net, which stands for the organization view. The lower part represents the process view. Relations between organization units are modeled by delivery and contract transitions. After instantiating the process model from the organization model with *x:new objectnet*, the model can be simulated. A validation has the potential to prove the resulting models concerning their correctness, e. g. interdependencies between organization units or established communications. The structure and behavior of the Object Petri net can be verified, so e. g. if the net is terminated correctly or deadlocks are reached. Different performance indicators such as process times, process costs, or the number of produced output can be ascertained. These can be used to detect bottle necks, deadlocks, or livelocks, so finally process and organizational models can be changed without having trouble in runtime processes.

The step-by-step simulation of the RENEW-Tool allows the tracking how in each organization the process is partly executed and then passed to the next organization. However, currently only one instance exists at a time and can be passed. Hence no process concurrencies involving multiple organizations can be realized with this design. The reason for this is that currently only one copy of the process instance is available. Multiple copies of the process instance would lead to the possibility to execute concurrent process tasks spitted by an AND-connector by different organizations. Additional limitation is based on the Reference net formalism deployed which is restricted to have one individual object net only in order to avoid inconsistent execution. To model the execution of different parts of the process within multiple organizations independently, we need to have multiple individual copies of the same process. This would cause to apply the value semantic of object Petri nets and the p-marking which is not possible with the RENEW-Tool.

6 Conclusion

This paper presented the concept of the transforming BPMN-conform XML for process models and a proprietary OMN-XML representing organization models into two interrelated PNML files representing a Reference net which is a special type of object Petri nets. The output PNML files can be imported into the RENEW-Tool and simulated considering control-flow as well as resource perspective of a business process.

Based on the introduced concept we have implemented a JAVA based converter using the BPMN-XML and OMN-XML and creating according PNML files. Currently the concept and our implementation are at a prototype stage. While it can demonstrate the general concept of interrelating process and organization models to a singe formal notation, there is still lot more to do. Firstly we need to broaden the concept of the system net to have multiple copies of one process instance. This would lead to model concurrent execution of tasks derived from an AND-split. Secondly, we need to add further BPMN elements including the OR-connector. Since Petri nets allow analyzing processes, it would be helpful to deliver processing time, resource availability, path probability and market demand information with the BPMN input processes. The converter could use this information to generate timed Petri nets which can be used for performance analysis and capacity planning or elementary Petri nets to apply common verification techniques. However, it has to be determined how these analysis techniques can be applied to the object Petri net approach.

Acknowledgement

The research project ArKoS – Architecture of Collaborative Scenarios – has been funded by the German Federal Ministry of Education and Research. The authors gratefully acknowledge the support of Ingo Taraske und Dariusz Kleczek with the implementation of the converter.

References

[Aa02] van der Aalst, W. M. P.: Making Work Flow: On the Application of Petri Nets to Business Process Management. In: J. Esparza; C. Lakos (Eds.): Applications and Theory of Petri Nets 2002: Proc. of 23rd International Conference (ICATPN). Bd. 2360, Adelaide, Australia 2002, pp. 1-22.

[Aa98] van der Aalst, W. M. P.: The Application of Petri Nets to Workflow Management. In: The Journal of Circuits, Systems and Computers 8 (1998) 1, pp. 21-66.

[Aa99] van der Aalst, W. M. P.; van der Moldt, D.; Valk, R.; Wienberg, F.: Enacting Interorganizational Workflow Using Nets in Nets. In: J. Becker; M. zur Muehlen; M. Rosemann (Eds.): Workflow Management '99. Münster, Germany 1999, pp. 117-136.

[ADK02] van der Aalst, W. M. P.; Desel, J.; Kindler, E.: On the semantics of EPCs: A vicious circle. In: F. J. Rump (Eds.): EPK 2002 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten. Trier, Germany 2002, pp. 71-79.

[AH02] van der Aalst, W. M. P.; van Hee, K. M.: Workflow Management : Models, Methods, and Systems. Cambridge 2002.

[AHH94] van der Aalst, W. M. P.; van Hee, K. M.; Houben, G. J.: Modelling and analysing workflow using a Petri-net based approach. In: G. D. Michelis; C. Ellis; G. Memmi (Eds.): 2nd Workshop on Computer-Supported Cooperative Work, Petri nets and related formalisms. Zaragoza, Spain 1994, pp. 31-50.

[AKV03] van der Aalst, W. M. P.; Kumar, A.; Verbeek, H. M. W.: Organizational Modeling in UML and XML in the context of Workflow Systems. In: J. Carroll (Eds.): Symposium on Applied Computing (SAC) 2003. Melbourne, Florida, USA 2003, pp. 603-608.

[Bi03] Billington, J.; Christensen, S.; van Hee, K. M.; Kindler, E.; Kummer, O.; Petrucci, L.; Post, R.; Stehno, C.; Weber, M.: The Petri Net Markup Language: Concepts, Technology, and Tools. In: E. Best (Eds.): Applications and Theory of Petri Nets 2003, 24th International Conference, ICATPN 2003. Bd. 2679, Eindhoven, The Netherlands 2003, pp. 483-505.

[Co38] Coase, R. H.: The Nature of Firm. In: Economica 4 (1938), pp. 386-405.

[CP77] Caves, R. E.; Porter, M. E.: From entry barriers to mobility barriers: conjectural decisions and contrived deterrence to new competition. In: Quarterly Journal of Economics 91 (1977) 2, pp. 241-261.

[Da01] Davis, R.: Business process modelling with ARIS : a practical guide. London et al. 2001.

[EM00] Evaristo, R.; Munkvold, B. E.: Collaborative Infrastructure Formation in Virtual Projects. Sixth Americas Conference on Information Systems (AMCIS). Long Beach, California, USA 2000, pp. 1705-1710.

[GV03] Girault, C.; Valk, R.: Petri Nets for System Engineering : A Guide to Modeling, Verification and Applications. Berlin et al. 2003.

[HP90] Hamel, G.; Prahalad, C. K.: The Core Competence of the Corporation. In: Harvard Business Review (1990) May 1991, pp. 79-91.

[JV87] Jessen, E.; Valk, R.: Rechensysteme: Grundlagen der Modellbildung. Berlin et al. 1987.

[Ku02] Kummer, O.: Referenznetze. Berlin 2002.

[Ku04] Kummer, O.; Wienberg, F.; Duvigneau, M.; Schumacher, J.; Köhler, M.; Moldt, D.; Rölke, H.; Valk, R.: An extensible editor and simulation engine for Petri nets: Renew. In: W. Reisig (Eds.): 5th International Conference on Application and Theory of Petri Nets (ICATPN 2004). Bd. 3099, Bologna, Italy 2004, pp. 484-493.

[Mu04] zur Muehlen, M.: Organizational Management in Workflow Applications - Issues and Directions. In: Information Technology and Management 5 (2004) 4.

[Pe62] Petri, C. A.: Kommunikation mit Automaten. Bonn 1962.

[RKC98] Rittenbruch, M.; Kahler, H.; Cremers, A. B.: Supporting Cooperation in a Virtual Organization. In: R. Hirschheim; M. Newman; J. I. DeGross (Eds.): Nineteenth International Conference on Information Systems (ICIS). Helsinki, Finland 1998, pp. 30-38.

[RR98] Reisig, W.; Rozenberg, G.: Lectures on Petri Nets I: Basic Models. Bd. 1491, Berlin et al 1998.

[Th05a] Theling, T.; Zwicker, J.; Loos, P.; Adam, O.; Hofer, A.: Enabling Dynamic Networks using an Architecture for Collaborative Scenarios. In: R. J. Scherer; P. Katranuschkov; S.-E. Schapke (Eds.): CIB W78 - 22nd Conference on Information Technology in Construction. Bd. 304, Dresden, Germany 2005, pp. 83-90.

[Th05b] Theling, T.; Zwicker, J.; Loos, P.; Vanderhaeghen, D.: An Architecture for Collaborative Scenarios applying a common BPMN-Repository. In: L. Kutvonen; N. Alonistioti (Eds.): Distributed Applications and Interoperable Systems: 5th IFIP WG 6.1 International Conference (DAIS). Bd. 3543, Athen, Greece 2005, pp. 169-180.

[Va04] Valk, R.: Object Petri Nets - Using the Nets-within-Nets Paradigm. In: G. Rozenberg (Eds.): Lectures on Concurrency and Petri Nets: Advances in Petri Nets. Bd. 3098, Berlin et al. 2004, pp. 819-848.

[Va98] Valk, R.: Petri Nets as Token Objects - An Introduction to Elementary Object Nets. In: J. Desel; M. Silva (Eds.): 19th Int. Conf. on Application and Theory of Petri Nets, ICATPN'98. Bd. 1420, Lisbon, Portugal 1998, pp. 1-25.

[Wh04] White, S. A.: Introduction to BPMN. http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf, Abruf 2005-11-17.

[Wi95] Williamson, O. E.: Organization Theory: From Chester Barnard to the Present and Beyond. Oxford 1995.

[Zi77] Zisman, M. D.: Representation, Specification and Automation of Office Procedures. Philadelphia, Pennsylvania 1977.