Corinna Pütz, Elmar J. Sinz

Model-driven Derivation of BPMN Workflow Schemata from SOM Business Process Models[1]

The Business Process Model and Notation (BPMN) has emerged as one of the dominant graphical modelling languages for processes in recent years. Its usage is both for conceptual workflow modelling and for specification of executable workflow schemata. Nevertheless BPMN seems to be less suitable for modelling of business processes. Business process models describe production and delivery as well as control of goods and services according to the goals of an enterprise (task level). These characteristics are not covered explicitly by BPMN. In contrast, workflow schemata specify solution procedures for the execution of business tasks (actor level). In this paper a two-stage modelling approach is proposed in order to overcome the semantic gap between business process models and workflow schemata: In a first step a business process model according to the Semantic Object Model (SOM) is created and stepwise refined. In a second step a BPMN workflow schema is derived from an adequately refined business process model according to a metamodel-based schema transformation. The modelling approach is illustrated using the case study of an online auction house.

1 Introduction

In recent years the Business Process Model and Notation (BPMN) has emerged as one of the dominant graphical modelling languages for processes. BPMN has been standardised by the Object Management Group (OMG). Its current version is 1.2 (OMG 2009a). Version 2.0 is in preparation.

Different goals are pursued with the usage of BPMN (White and Miers 2008, p. 24; Allweyer 2008, pp. 9ff):

- *Conceptual modelling of workflows* aims for the documentation of workflows and for a communication basis for analysis and design of workflows in an enterprise. Execution of workflows is not focused.
- *Specification of executable workflow schemata* aims for a complete and detailed specification of workflows in order to transform them, for instance, to executable BPEL processes (WS-BPEL, Web-Services Business Process Execution Language (OASIS 2007)). For export of workflow schemata from a BPMN tool and import into a BPEL engine the representation language XML Process Definition Language (XPDL (WFMC 2008)) is used particularly.

Conceptual modelling of workflows can be used in order to expand the results subsequently towards executable workflow schemata (Silver 2009, pp. 9f).

In science and practice, an explicit distinction between business processes and workflows is not common (see, e.g., Huth and Wieland 2008; Allweyer 2008, p. 8). However, for the approach presented in this paper the differentiation between business process model and workflow schema is crucial:

- Targeted towards the overall predetermined goals of an enterprise, a business process

[1]Extended version of: Pütz, C., Sinz, E. J. (2010) Modellgetriebene Ableitung von BPMN-Workflowschemata aus SOM-Geschäftsprozessmodellen. In: Engels, G. et al. (Eds.) Modellierung 2010. March 24–26, 2010, Klagenfurt, Austria, Proceedings. Gesellschaft für Informatik, Bonn, pp. 253–268.

model specifies production and delivery as well as control of goods and services and references the used resources (Ferstl and Sinz 2008, pp. 193f). The description is carried out by means of business tasks and event relations between tasks. The notion of task is one of the fundamental concepts in business administration (Kosiol 1976) and describes a goal-oriented action which is performed on a task object. The goals of the tasks are derived from the goals of the enterprise.

- In contrast, a workflow schema describes a solution procedure which is carried out either by human or machine actors (persons or application systems) while executing one or several business tasks. Description is done by means of activities and relations between activities. An activity may be an elemental or non-elemental activity within the solution procedure.

The naming of BPMN suggests that the language aims at the modelling of business processes. However, following the previous differentiation, it becomes clear that BPMN is aimed primarily at the modelling of workflows. The main language elements are activities and their relations (message flow and sequence flow) (OMG 2009a). Though, for business process modelling as stated above, BPMN is less suitable. This is why the relationship between business tasks and the goals of the enterprise as well as the control of production and delivery of goods and services cannot be represented adequately.

The approach presented in this paper proposes to derive BPMN workflow schemata from business process models using a model-driven approach. Thereby, the Semantic Object Model (SOM, see Ferstl and Sinz 2006; Ferstl and Sinz 2008, pp. 192ff) is used for business process modelling.

Starting with initial flows of goods and services between the universe of the discourse (normally the enterprise) and its environment, a SOM business process model is stepwise refined. Thereby, control of the production of goods and services becomes visible in terms of coordinating relations between business objects and their tasks. After obtaining a sufficient level of detail, a BPMN workflow schema is derived from a business process model by model-driven derivation.

Compared to a direct specification of BPMN workflow schemata the approach as it is proposed here comprises a range of advantages: pools, choreography between pools, orchestration of activities within pools and other features of BPMN workflows are derived systematically from business process models. Semantic properties of business process models are taken over to enrich the workflow specification. The systematic derivation also aims at improving the model quality of the workflow schemata.

The paper is organised as follows: Section 2 outlines methodological foundations of the model-driven derivation of BPMN workflow schemata from SOM business process models. Section 3 introduces the case study, an online auction house which is described by means of a stepwise refined SOM business process model. The model-driven derivation of a BPMN workflow schema from a SOM business process model is presented in Sect. 4. Section 5 discusses the proposed approach. Finally, Sect. 6 provides an overview of related work and gives an outlook for further research.

2 Methodological foundations for the model-driven derivation of BPMN workflow schemata from SOM business process models

The Semantic Object Model (Ferstl and Sinz 2006; Ferstl and Sinz 2008, pp. 192ff) is an object- and business process-oriented methodology for comprehensive modelling of business systems. The SOM enterprise architecture (Fig. 1a) comprises three model layers: (1) The *enterprise plan* describes the global task of the business system from an outside perspective (outside of the business system) and specifies its goals to be pursued.

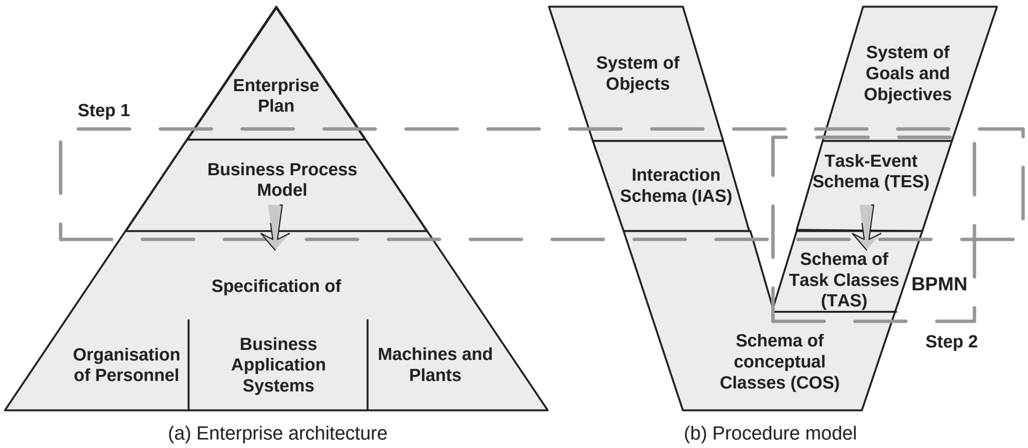

Figure 1: Enterprise architecture and procedure model of SOM methodology (based on Ferstl and Sinz 2006; Ferstl and Sinz 2008, pp. 193, 195)

(2) The *business process model* depicts the solution procedure for realizing the enterprise plan; it specifies the inside perspective on the tasks of the enterprise. (3) The *resource model* describes the actors for carrying out the business tasks of the business system from an inside perspective of the enterprise. Actors of the enterprise are either humans or machines.

According to the model layers of the SOM enterprise architecture the SOM procedure model (Fig. 1b) specifies structural and behavioural views for the representation of the three model layers. The main focus of this paper is on the interaction schema (IAS) and the task-event schema (TES) specifying respectively the structure and behaviour of a business process model, as well as on the schema of task classes (TAS) describing the behaviour of the resource model. In this article, the TAS is specified using a BPMN workflow schema.

The proposed methodology comprises two steps:

1. Creation and stepwise refinement of a SOM business process model (IAS and TES).
2. Model-driven derivation of a BPMN workflow schema from the most detailed level of a decomposed TES.

Step 1: The metamodel for SOM business process models is presented in Fig. 2. A *business object* comprises a set of *tasks* with associated goals, executed on one and the same task object. Each business object either belongs to the universe of the discourse (rectangle) or to the environment (oval) of a business system. The coordination of business objects is specified by business transactions. A *business transaction* always connects two tasks belonging to different business objects.

The SOM methodology uses two types of coordination principles (Ferstl and Sinz 2006; Ferstl and Sinz 2008, pp. 66ff). The first type is the *feedback control principle* according to which a business object is decomposed into two sub-objects, a management object and an operational object, as well as two transactions, a control transaction (r) and a feedback transaction (f). The second type, the *negotiation principle* is used to decompose a transaction into three successive transactions: an initiating transaction (i), a contracting transaction (c), and an enforcing transaction (e). In the initiating transaction, a server object and its client get to know each other. They exchange information on deliverable goods or services. In the contracting transaction both objects agree to a contract on the delivery of goods or services.

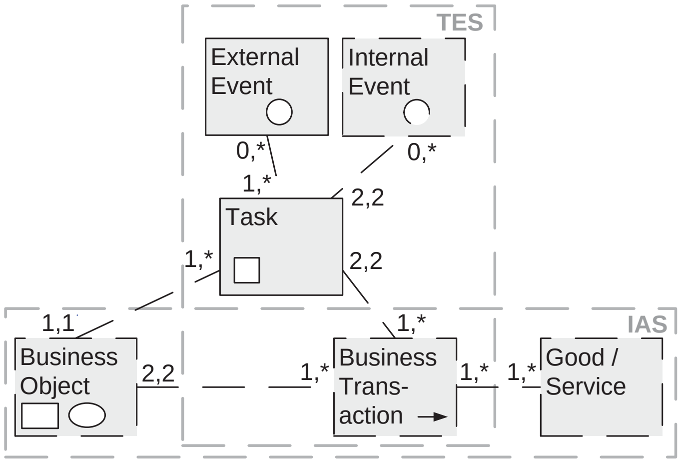

Figure 2: Metamodel for SOM business process models (Ferstl and Sinz 2008, p. 210)

The goods or services are transferred in the enforcing transaction.

The refinement of the coordination between business objects in a business process model is formally specified by decomposition rules:

a) Decomposition rule for business objects (feedback control principle):
O ::= {O',O",T_r(O',O"),[T_f(O",O')]}

b) Decomposition rule for business transactions (negotiation principle):
T(O,O') ::= [[T_i(O,O') seq] T_c(O',O) seq] T_e(O,O')

Rule (a) specifies the feedback control principle. A business object O is decomposed into subobjects O' and O", a controlling transaction T_r from O' to O" and an optional feedback transaction T_f from O" to O'. According to the negotiation principle, rule (b) specifies the decomposition of a transaction T from O to O' into three subsequent transactions: An initiating transaction T_i from O to O', a contracting transaction T_c from O' to O and an enforcing transaction T_e from O to O'. T_i or both T_i and T_c can be omitted, if server and client already know each other or if they rely on an existing contract.

The negotiation-principle is exemplarily shown in Fig. 3 by the interaction between buyer and online auction house from a structural view: The IAS consists of the business objects *buyer* (environmental object) and *online auction house* (object of the universe of the discourse). The buying of goods at an auction is initiated by sending *information* about available auctions to a (potential) buyer (initiation). The buyer places one or several binding *bids* (contracting) and receives, as the case may be, the *winning bid* (enforcing).

In Fig. 4 the corresponding behavioural view is shown by means of a TES. The TES specifies the event-controlled sequence of task executions (cf. Fig. 2). The foundation of TES is the Petri net paradigm (cf., e.g., Reisig 1986; Peterson 1977) (Fig. 4a). A Petri net is executed by the firing of feasible transitions. For instance, the transition *receive information* is feasible if the state *information transmission* is marked. After firing the transition, the state *info available* is marked and the state *information transmission* is unmarked.

A TES is perceived as an extended Petri net with the following characteristics: it is a coloured Petri net (cf., e.g., Jensen and Kristensen 2009, pp. 13ff) with distinguishable marks (distinction of buyers and bids). Transitions can be complemented by pre-conditions and post-conditions, specifying the execution in greater detail. The

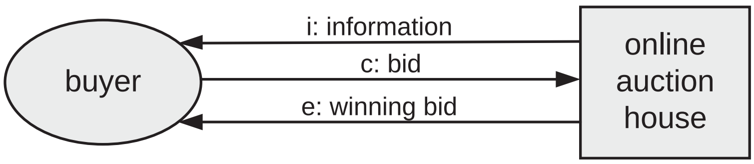

Figure 3: Negotiation principle between online auction house and buyer

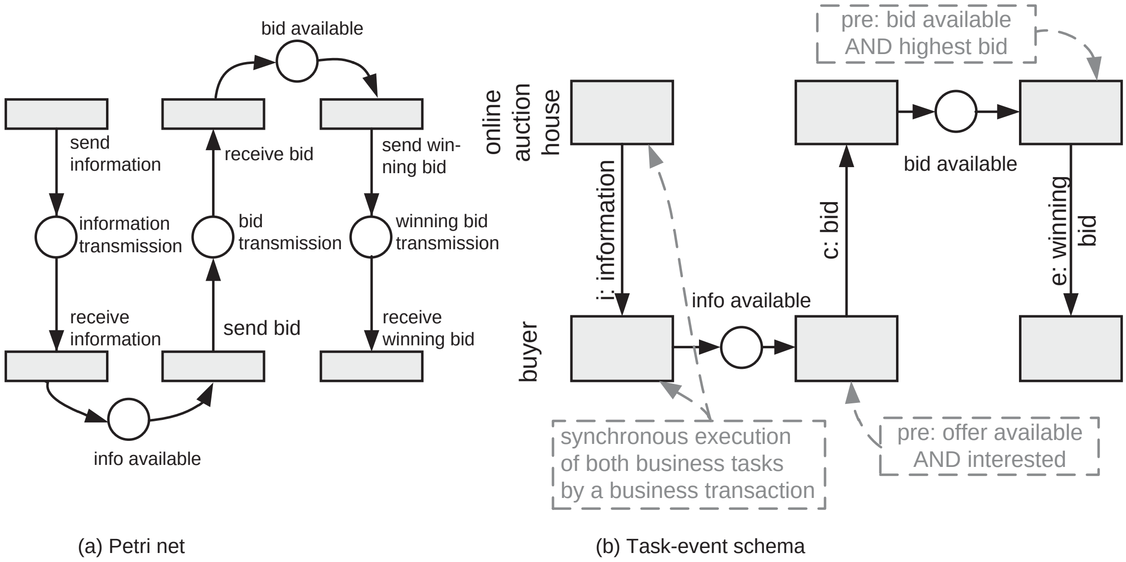

Figure 4: Task-event schema online auction house

two transitions, representing the business tasks associated with a business transaction, are supposed to fire synchronously. Therefore, the Petri net state combining the two transitions is not represented in TES. The TES corresponding to the Petri net is shown in Fig. 4b.

Step 2: The BPMN elements which are relevant for the derivation of the workflow schema are outlined in terms of a metamodel (Fig. 5). According to OMG, the elements are arranged into the main categories *swim lane, flow object* and *connecting object* (OMG 2009a). A *swim lane* specifies either a pool or a lane. A pool represents a participant and can be divided by lanes into roles. Available *flow objects* are (1) activities (atomic task or non-atomic sub process), (2) events (start event, intermediate event or end event) or (3) gateways (divergence and convergence of sequence flows). *Connecting objects* are either sequence flows which are used to specify the execution order of flow objects within a pool or message flows which specify the exchange of messages between two pools.

While a TES is based on the 'Petri net' paradigm, a BPMN schema conforms to the paradigm of 'algorithm'. The instantiation of a Petri net is specified by a set of tokens which are allocated to the various states. In contrast, a BPMN work-

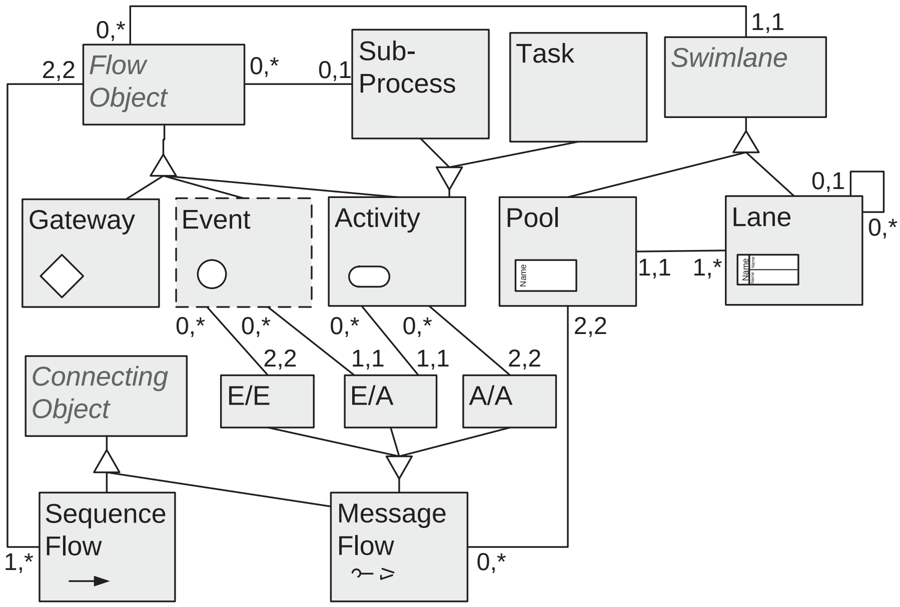

Figure 5: Metamodel for BPMN workflow schemata

flow schema can be instantiated several times. Each instance corresponds to a separate execution according to the schema. The current state of the execution is thereby marked by a single token.

The derivation of an initial BPMN workflow schema from a detailed TES is achieved by a metamodel-based schema transformation. This procedure is outlined in Sect. 4.

3 Case study of an online auction house

In the following, an online auction house as it is well known, e.g., in the form of eBay[2] is used as a case study. For this online auction house a SOM business process model has been developed. This is done by a multistage, stepwise decomposition of the IAS and the corresponding TES.

The initial IAS (Fig. 6a) shows the aggregated flows of services between the business objects from a structural view. The object *online auction house* (universe of the discourse) provides the object *buyer* (environment) with the service *sale* and the object *seller* (environment) with the service *mediation*. Both services are delivered in corresponding enforcing transactions. The delivery of the mediated good is carried out directly between *seller* and *buyer*. The corresponding TES is shown in Fig. 6b. The names of the TES tasks are derived from the names of the transactions. Thus, *mediation>* denotes the task 'production and delivery of mediation service' and *>mediation* the corresponding receiving task.

The coordination between *online auction house* and *seller* resp. *buyer* follows the negotiation principle. In a first decomposition the two enforcing transactions starting at the online auction house (Fig. 7) are refined following the (i, c, e) principle. After gaining access to the seller's account (initiating) the seller establishes an auction (contracting). After the auction has terminated, the seller is informed about its result and the accounting takes place (enforcing). Correspondingly the buyer receives information about existing auctions (initiating), he or she can place

[2] http://www.ebay.com/

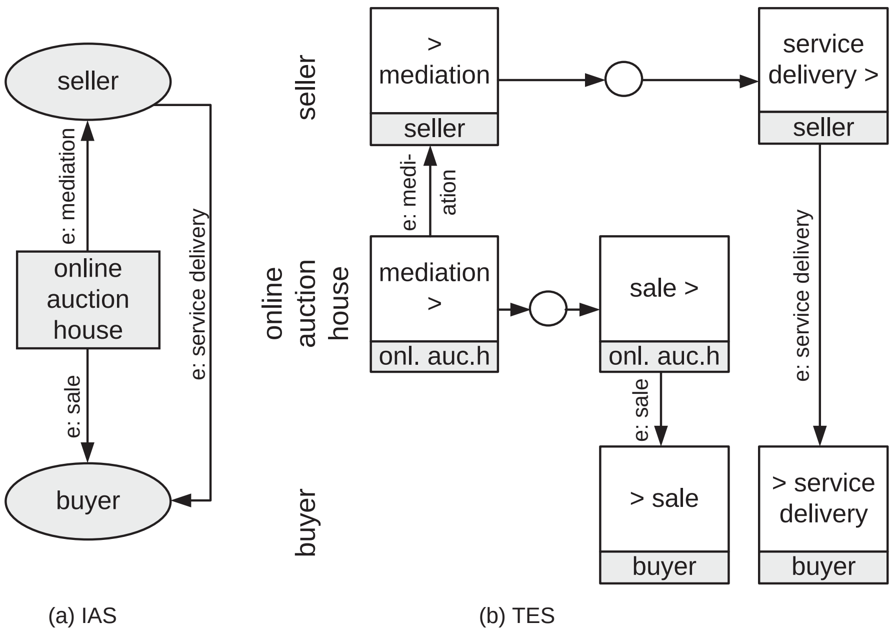

Figure 6: IAS and TES online auction house (initial model)

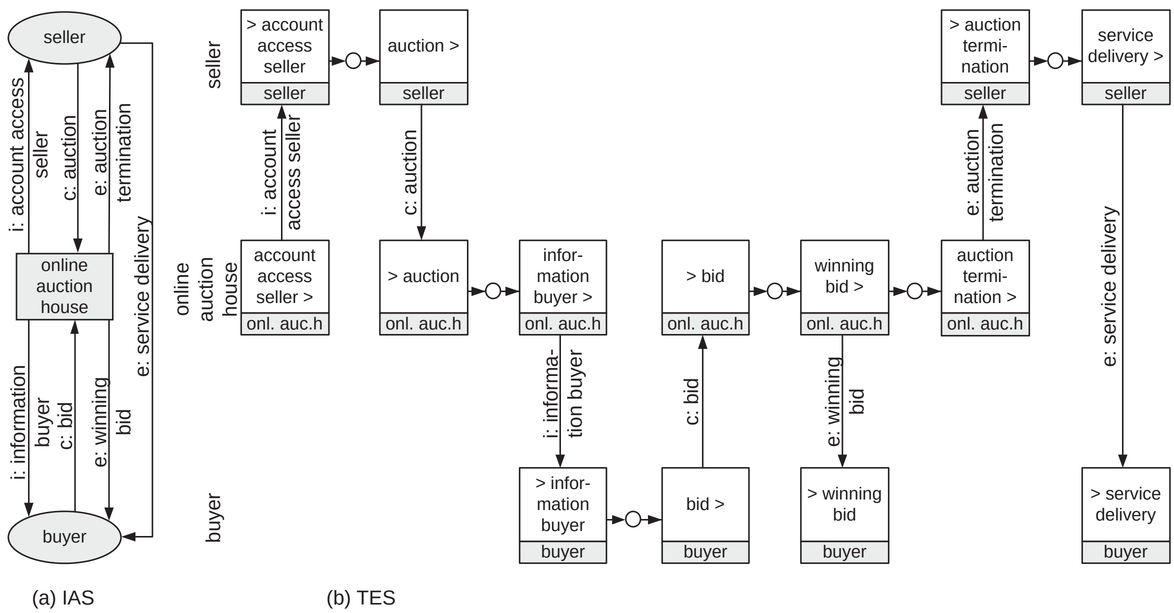

Figure 7: IAS and TES online auction house (first decomposition)

bids (contracting) and receives, as the case may be, the winning bid notification (enforcing).

In the second decomposition (Fig. 8 and Fig. 10), the coordination protocols between *online auction house* and *seller* resp. *buyer* are further refined. Moreover, the business object *online auction house* is decomposed. The latter decomposition leads to two sub-objects each coordinating the buyer's and the seller's side as well as to a third sub-object, managing the stock of bids and auctions. The object *bidding and auction management* receives orders from *bidding handling* and *auction handling* according to the negotiation principle. The whole decomposition protocol of objects and transactions is shown in Fig. 9.

Tasks of the second TES decomposition (Fig. 10) are complemented – if necessary – by pre-conditions and post-conditions. Those are entered over or under the respective task name and labelled with the key word PRE or POST. A buyer requests for example an auction overview (*request auction overview*>) only if he or she is signed on and interested in taking part at an auction.

In the following it is assumed that the second decomposition of TES is detailed enough in order to derive a BPMN workflow schema. This derivation is explained in the next chapter.

4 From a SOM business process model to a BPMN workflow schema

The derivation of a BPMN workflow schema from a SOM business process model methodically utilises a metamodel-based schema transformation according to the MDA-Pattern of the Model Driven Architecture (OMG 2003, p. 3-9; see also Frankel 2003; Gruhn et al. 2006) (Fig. 11, left). Derivation starts with the behavioural view, i.e., a TES of an adequately refined SOM business process model which is specified according to the related metamodel. The result of the derivation is an initial BPMN workflow schema according to the introduced BPMN metamodel. The derivation is specified by a mapping, relating model elements of the SOM metamodel to model elements of the BPMN metamodel (see dashed arrows in Fig. 11, right).

The most important relations of the mapping from a SOM business process model to a BPMN workflow schema are explained here briefly: Each business object is transformed into a pool. Business transactions between business objects lead to message flows between pools. Tasks are related to activities as well as to events if necessary. An object-internal event that combines two tasks inside a business object corresponds to a sequence flow within a pool. Gateways are derived from object-internal events combined with preconditions or post-conditions. Consecutive flow objects within a pool are connected additionally by sequence flows.

Following the derivation, it is achieved that sequences within a business object are transformed into sequences within a pool. Coordination between business objects leads to a choreography between pools that is specified by message flows (see OMG 2009a).

The BPMN workflow schema resulting from the derivation is shown in Fig. 12. The business objects *seller, auction handling, bidding handling* and *buyer* can be directly transformed into corresponding pools of the BPMN schema. In contrast, the derivation of the business object *bidding and auction management* needs special attention. This is due to the crossover of the TES' 'Petri net' paradigm to the BPMN schema's 'algorithm' paradigm. While the TES extract of the business object *bidding and auction management* shows three starting points in terms of the tasks >*auction notification incl. time-out,* >*request auction summary* and >*bidding notification*, a pool of an executable BPMN schema must have one unique starting event. Otherwise the corresponding processing is not algorithmically determined.

The three starting points of the business object *auction and bidding management* result from the

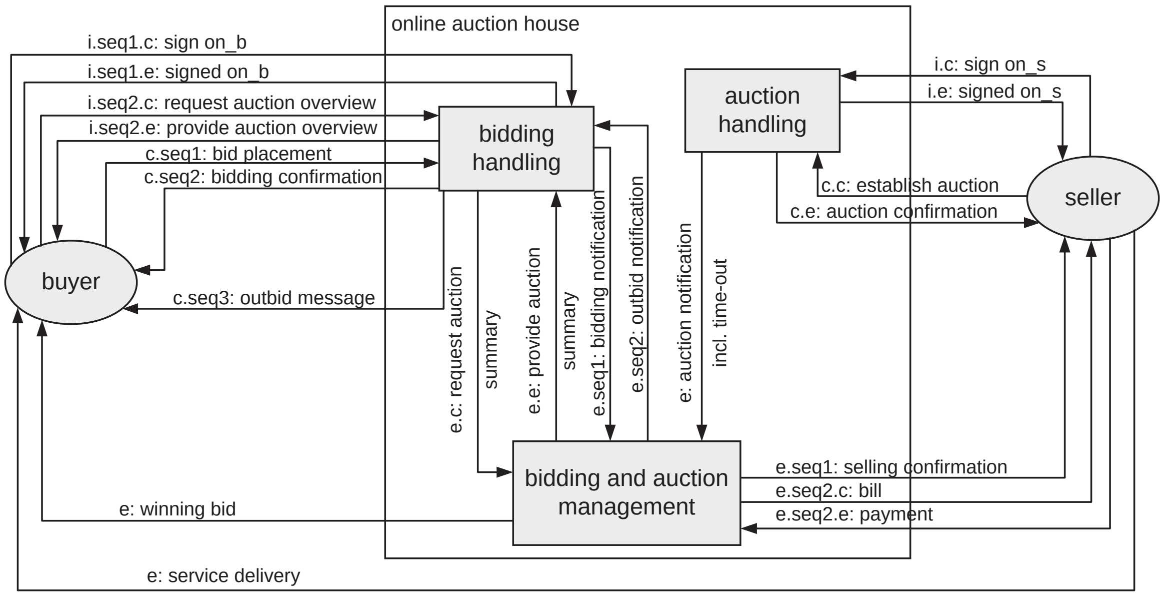

Figure 8: IAS online auction house (second decomposition)

Object Decomposition	Transaction Decomposition	
online auction house	e: sale	e: mediation
auction handling	i: information buyer	i: account access seller
e: auction notification incl. time-out	i.seq1: account access buyer	i.c: sign on_s
bidding and auction management	i.seq1.c: sign on_b	i.e: signed on_s
e: auction summary	i.seq1.e: signed on_b	c: auction
e.c: request auction summary	i.seq2: auction overview	c.c: establish auction
e.e: provide auction summary	i.seq2.c: request auction overview	c.e: auction confirmation
e: bidding message	i.seq2.e: provide auction overview	e: auction termination
e.seq1: bidding notification	c: bid	e.seq1: selling confirmation
e.seq2: outbid notification	c.seq1: bid placement	e.seq2: accounting
bidding handling	c.seq2: bidding confirmation	e.seq2.c: bill
buyer	c.seq3: outbid message	e.seq2.e: payment
seller	e: winning bid	e: service delivery

Figure 9: Object and transaction decomposition online auction house

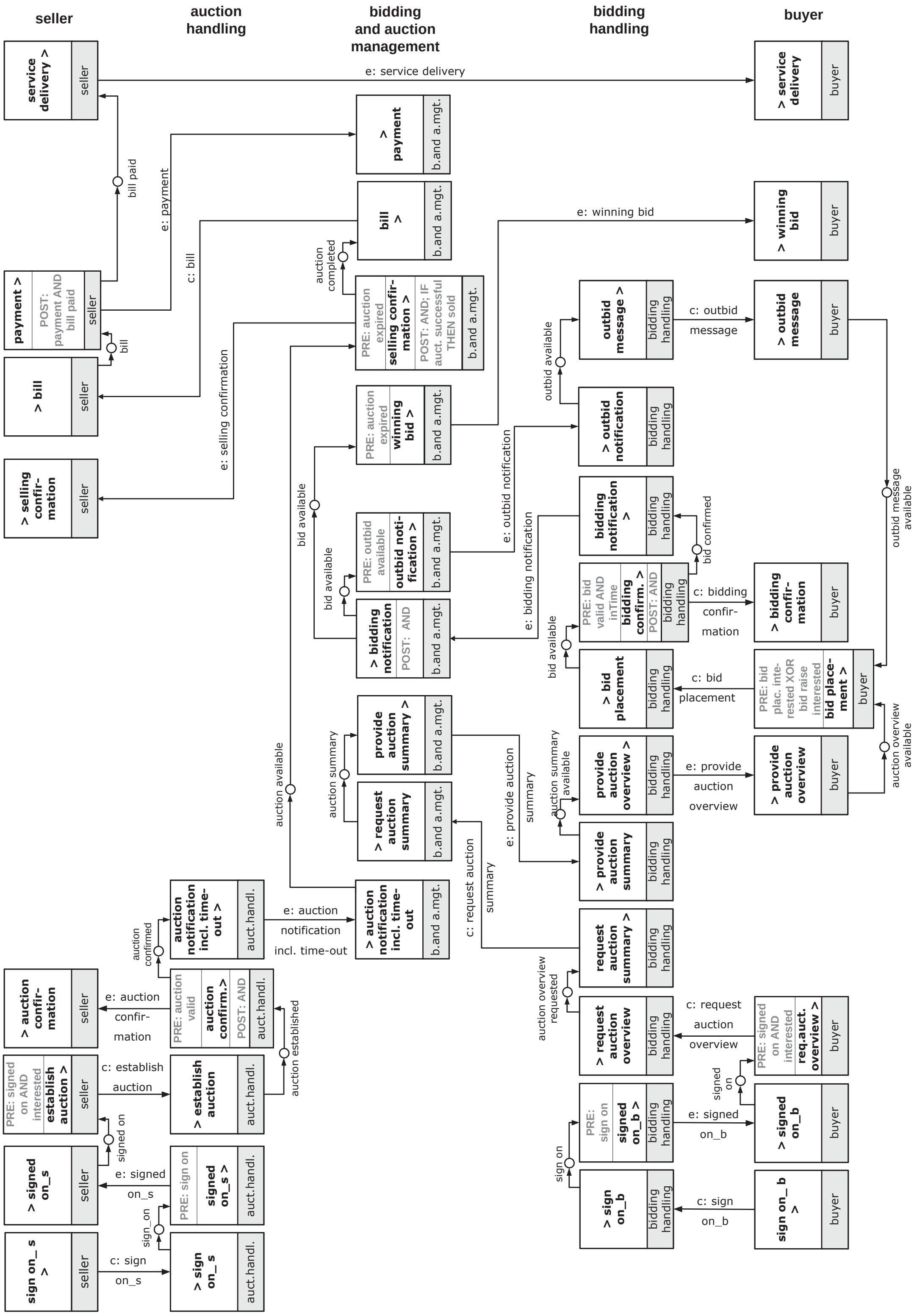

Figure 10: TES online auction house (second decomposition)

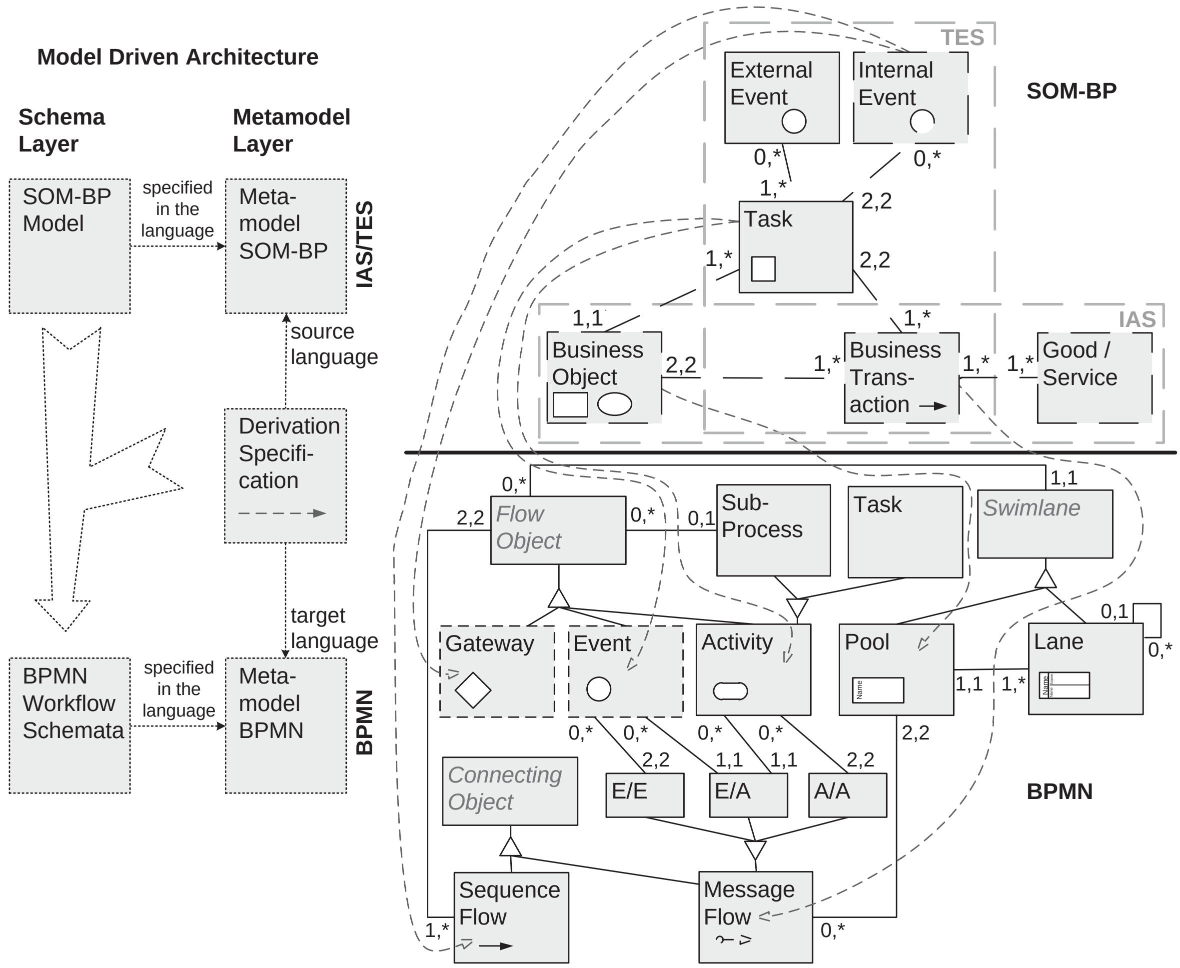

Figure 11: Mapping of metamodel SOM → BPMN (cf. OMG 2003, p. 3-9; Ferstl and Sinz 2008, p. 210)

fact that *auction notification incl. time-out, request auction summary* and *bidding notification* refer to a shared task object, i.e., bids and auctions. This is why the business object is not further decomposable from the viewpoint of object-oriented modelling. However, this adhesion of the task object is not relevant for a process-oriented BPMN schema. Therefore the business object *bidding and auction management* is transformed into several pools with one start event each (*auction info, bidding management, auction management*).

5 Discussion of the approach proposed

Starting point for the approach proposed in this article is an initial SOM business process model that incorporates the service relations of the business process and its environment according to an enterprise plan. The business process model is specified by a structural view (IAS) and a corresponding behavioural view (TES). It is stepwise refined by the decomposition of business transactions and business objects. Doing so, the transaction-based coordination of business objects is revealed step by step. Once an adequate

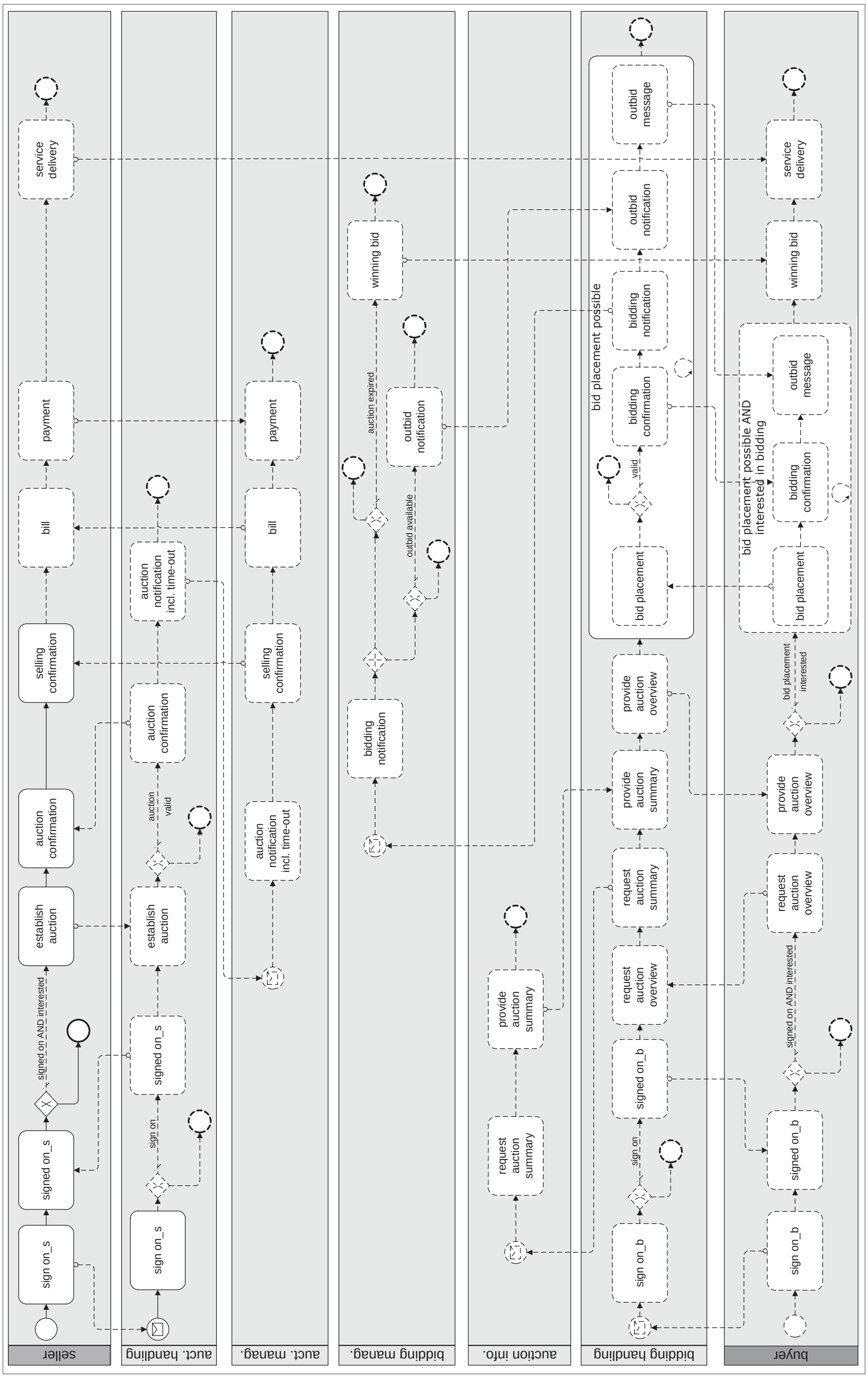

Figure 12: BPMN workflow schema online auction house

refinement of the business process model is reached, the model-driven derivation of an initial BPMN workflow schema from a TES at the most detailed level of decomposition takes place. An adequate refinement is achieved, when the solution procedures of all tasks of the business process model can be represented by just one corresponding activity each within the workflow schema.

Compared to a direct modelling of a BPMN workflow schema without a preceding SOM business process model, the model-driven derivation presented here yields to a number of advantages:

- Pools, the choreography between pools as well as the activities and relationships of the initial BPMN schema result directly from the derivation and do not have to be constructed or reconstructed (cf. also Allweyer 2008, pp. 53ff).
- The 'origin from the business process model' can be used for a semantic annotation of the workflow schema's artefacts. This enrichment increases the semantic expressiveness of a BPMN schema. A message flow for example can be supplemented with the information to which business transaction it conforms to in the business process model. The choreography between pools reflects the entire coordination protocol of the corresponding business objects.
- The semantic gap between an initial business process model, which is aligned to the enterprise plan and the workflow schema specifying the solution procedure for the execution of the detailed business process tasks, will be decreased to manageable and verifiable margins of complexity.
- The initial BPMN schema derived from model transformation provides the border within the BPMN workflow schema may be further processed. The workflow schema can be refined in order to specify, e.g., variants of tasks executions. However, the initial structure of the derived BPMN schema must not be altered as this would violate the alignment between the business process model and the workflow schema.

All in all, the approach presented here aims for improvement of model quality and semantical expressiveness of workflow schemata.

Prerequisite for the application of the approach is the conceptual differentiation of the notions of business process model and workflow schema (see Sect. 1). A business process model refers to the task level of a business system and describes the 'what' of a goal-oriented task fulfilment. On the other hand, a workflow schema refers to the actor level and describes the 'how' of the task execution. The approach proposed here leads to an alignment of both levels. At the same time it points out the degree of freedom of *how* to reach a given *what*. For instance, for a given business process there can be specified different variants of workflows by refining an initial workflow schema.

6 Related work and future research needs

Particularly the model transformation from and to BPMN has been previously covered in related work. Thereby, only a few approaches are using BPMN as a target language for the model transformation. Examples where event-driven process chains and UML activity diagrams are transformed into BPMN are given in Allweyer (2007) and Kalnins and Vitolins (2006).

Many approaches are using BPMN as a source language for the model transformation. For example Decker et al. (2008b) provide a transformation of BPMN into the workflow language YAWL, Dijkman et al. (2008) into Petri nets, and Cibrán (2009) into UML activity diagrams. A number of approaches cover the generation of BPEL specifications from BPMN (e.g., OMG 2009a; Ouyang et al. 2006a; Ouyang et al. 2006b). As there are efficient BPMN editors, transformers for different target languages and BPEL generators available, BPMN seems to be a good choice for the specification of workflow schemata.

Questions concerning the modelling of choreography in BPMN are among others discussed in Decker and Barros (2008). The new version 2.0 of BPMN will contain specific language elements for choreography modelling (OMG 2009b). Decker et al. (2008a) cover the transformation from BPMN to BPEL4Chor.

The 'auctioning example' is used in a number of papers to demonstrate a direct modelling of business processes resp. workflows. Examples that can be compared to the results achieved in this paper are Decker and Barros (2008), Decker and Puhlmann (2007) and Pascalau et al. (2009).

The suggested approach for a model-driven derivation of BPMN workflow schemata in its current form has still several limitations. These are in particular the consideration of data view of workflows (e.g., gateway conditions), exception handling of workflows (e.g., blocking of one workflow instance due to the behaviour of a corresponding workflow instance) as well as the explicit consideration of the degree of automation of business process tasks. Aspects of considering non-automated activities are also covered in the new version BPMN 2.0 and in BPEL4People (OASIS 2010).

Further need for research concerns the generalisation and formalisation of derivation rules from TES towards the BPMN workflow schema, the refinement of the activities and a suitable tool support for the transformation. The aim is to achieve an integrated and model-driven development of executable BPEL prototypes. Prerequisite for that is the consideration of task objects according to the SOM methodology (see Fig. 1b, schema of conceptual classes). These task objects have to be realised by services managing the persistent data which are underlying the activities of the workflow schema.

References

Allweyer T. (2007) Erzeugung detaillierter und ausführbarer Geschäftsprozessmodelle durch Modell-zu-Modell-Transformationen. In: Nüttgens M., Rump F. J., Gadatsch A. (eds.) EPK 2007 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten. 6th Workshop of Gesellschaft für Informatik e.V. (GI), November 2007, St. Augustin, pp. 23–38

Allweyer T. (2008) BPMN - Business Process Modeling Notation. Einführung in den Standard für die Geschäftsprozessmodellierung. Books on Demand, Norderstedt

Cibrán M. A. (2009) Translating BPMN Models into UML Activities. In: Ardagna D., Mecella M., Yang J. (eds.) Business Process Management Workshops. BPM 2008 International Workshops, Milan, Italy, September 1-4, 2008, Revised Papers. Lecture Notes in Business Information Processing Vol. 17. Springer, Berlin, pp. 236–247

Decker G., Barros A. (2008) Interaction Modeling Using BPMN. In: ter Hofstede A. H. M., Benatallah B., Paik H.-Y. (eds.) Business Process Management Workshops. BPM 2007 International Workshops, BPD, BPI, CBP, ProHealth, RefMod, semantics4ws, Brisbane, Australia, September 24, 2007, Revised Selected Papers. Lecture Notes in Computer Science Vol. 4928. Springer, Berlin, pp. 208–219

Decker G., Puhlmann F. (2007) Extending BPMN for Modeling Complex Choreographies. In: Meersman R., Tari Z. (eds.) On the Move to Meaningful Internet Systems 2007. OTM Confederated International Conferences CoopIS, DOA, ODBASE, GADA, and IS 2007, Vilamoura, Portugal, November 25-30, 2007, Proceedings. Lecture Notes in Computer Science Vol. 4803. Springer, Berlin, pp. 24–40

Decker G., Kopp O., Leymann F., Pfitzner K., Weske M. (2008a) Modeling Service Choreographies Using BPMN and BPEL4Chor. In: Bellahsène Z., Léonard M. (eds.) Advanced Information Systems Engineering. 20th International Conference, CAiSE 2008, Montpellier, France, June 16-20, 2008, Proceedings.

Lecture Notes in Computer Science Vol. 5074. Springer, Berlin, pp. 79–93

Decker G., Dijkman R. M., Dumas M., Garcia-Banuelos L. (2008b) Transforming BPMN Diagrams into YAWL Nets. In: Dumas M., Reichert M. (eds.) Business Process Management. 6th International Conference, BPM 2008, Milan, Italy, September 2 - 4, 2008, Proceedings. Lecture Notes in Computer Science Vol. 5240. Springer, Berlin, pp. 386–389

Dijkman R. M., Dumas M., Ouyang C. (2008) Semantics and analysis of business process models in BPMN. In: Information and Software Technology 50(12), pp. 1281–1294

Ferstl O. K., Sinz E. J. (2006) Modeling of Business Systems Using SOM. In: Bernus P., Mertins K., Schmidt G. (eds.) Handbook on Architectures of Information Systems. International Handbooks on Information Systems. Springer, Berlin, pp. 347–367

Ferstl O. K., Sinz E. J. (2008) Grundlagen der Wirtschaftsinformatik, 6th ed. Oldenbourg, München

Frankel D. S. (2003) Model Driven Architecture. Applying MDA to Enterprise Computing. OMG Press. Wiley, Indianapolis, Ind.

Gruhn V., Pieper D., Röttgers C. (2006) MDA. Effektives Software-Engineering mit UML 2 und Eclipse. Springer, Berlin

Huth S., Wieland T. (2008) Geschäftsprozessmodellierung mittels Software-Services auf Basis der EPK. In: Nissen V., Petsch M., Schorcht H. (eds.) Service-orientierte Architekturen. Chancen und Herausforderungen bei der Flexibilisierung und Integration von Unternehmensprozessen. Springer-11775 Vol. Part 2 - Modellierung und SOA-Entwurf. Deutscher Universitäts-Verlag, Wiesbaden, pp. 61–76

Jensen K., Kristensen L. M. (2009) Coloured Petri Nets. Modelling and Validation of Concurrent Systems. Springer, Berlin

Kalnins A., Vitolins V. (2006) Use of UML and Model Transformations for Workflow Process Definitions. In: Communications of the 7th International Baltic Conference on Databases and Information Systems (Baltic DB&IS'2006). Vilnius, Lithuania, July 3-6, 2006, pp. 3–14

Kosiol E. (1976) Organisation der Unternehmung, 2nd ed. Gabler, Wiesbaden

OASIS (2007) Web Services Business Process Execution Language Version 2.0. OASIS Standard. 2007-04-11. http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html. Last Access: 2010-05-21

OASIS (2010) WS-BPEL Extension for People (BPEL4People) Technical Committee. http://www.oasis-open.org/committees/bpel4people/charter.php. Last Access: 2010-05-14

OMG (2003) MDA Guide Version 1.0.1. Document Number: omg/2003-06-01. http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf. Last Access: 2010-05-14

OMG (2009a) Business Process Model and Notation (BPMN) - Version 1.2. OMG Document Number: formal/2009-01-03. http://www.omg.org/spec/BPMN/1.2/PDF. Last Access: 2010-05-10

OMG (2009b) Business Process Model and Notation (BPMN) - FTF Beta 1 for Version 2.0 OMG. OMG Document Number: dtc/2009-08-14. http://www.omg.org/cgi-bin/doc?dtc/09-08-14.pdf. Last Access: 2010-05-14

Ouyang C., van der Aalst W. M. P., Dumas M., ter Hofstede A. H. M. (2006a) Translating BPMN to BPEL. In: BPM Center Report BPM-06-02, BPMcenter.org

Ouyang C., Dumas M., ter Hofstede A. H. M., van der Aalst W. M. P. (2006b) From BPMN Process Models to BPEL Web Services. In: International Conference on Web Services (ICWS 2006), September 18-22, 2006, Chicago, Illinois, USA, Proceedings. IEEE Computer Society, Los Alamitos, pp. 285–292

Pascalau E., Giurca A., Wagner G. (2009) Validating Auction Business Processes using Agent-based Simulations. In: 2nd International Conference on Business Process and Services Computing. BPSC2009. Leipzig, 2009, Proceedings, pp. 95–109

Peterson J. L. (1977) Petri Nets. In: ACM Computing Surveys 9(3), pp. 223–252

Reisig W. (1986) Petrinetze. Eine Einführung, 2nd ed. Springer, Berlin

Silver B. (2009) BPMN method and style. Cody-Cassidy, Aptos

WFMC (2008) Process Definition Interface - XML Process Definition Language. Version 2.1a. Document Number WFMC-TC-1025. http://www.wfmc.org/index.php?option=com_docman&task=doc_download&Itemid=72&gid=132. Last Access: 2010-05-13

White S. A., Miers D. (2008) BPMN. Modeling and Reference Guide. Understanding and using BPMN. Develop rigorous yet understandable graphical representations of business processes. Future Strategies Inc., Lighthouse Point

Corinna Pütz, Elmar J. Sinz

Department of Information Systems –
Systems Engineering
University of Bamberg
Feldkirchenstr. 21
96045 Bamberg
Germany
{corinna.puetz | elmar.sinz}@uni-bamberg.de