Electronic Communications of the EASST (2010)

# Pre-Proceedings of the Fourth International Workshop on Graph-Based Tools (GraBaTs 2010)

## Design of a SOM Business Process Modelling Tool based on the ADOxx meta-modelling Platform

Domenik Bork and Elmar J. Sinz

13 Pages

Guest Editors: Juan de Lara, Daniel Varro
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

# Design of a SOM Business Process Modelling Tool based on the ADOxx Meta-modelling Platform

**Domenik Bork and Elmar J. Sinz**

Department of Information Systems – Systems Engineering
University of Bamberg
Feldkirchenstraße 21
D-96045 Bamberg
{domenik.bork | elmar.sinz}@uni-bamberg.de

**Abstract:** The Semantic Object Model (SOM) is a comprehensive methodology for business systems modelling. An emphasis of SOM is on modelling of business processes. According to SOM, a business process model specifies the task layer of a business system from an inside perspective. SOM business process modelling is grounded in systems theory and organisational theory. A SOM business process model is perceived as a distributed system, consisting of business objects which are coordinated in business transactions. Both business objects and business transactions can be refined recursively. SOM business process models are specified using a graph-based multi-view approach which comprises a structural view, a behavioural view as well as views on the decomposition of business transactions and business objects.
This article reports on the design of a tool which facilitates multi-view modelling of SOM business processes. The tool is based on the ADOxx[1] meta-modelling platform. The focus of the article is on the design of the multi-view approach and corresponding tool functions on the basis of the ADOxx platform.

**Keywords:** Semantic Object Model (SOM), ADOxx, graph-based multi-view modelling, tool design

## 1 Introduction

The Semantic Object Model (SOM) is a comprehensive methodology for business systems modelling (Ferstl and Sinz 1995, Ferstl and Sinz 1996, Ferstl and Sinz 2008). The conceptual framework of the SOM methodology is an enterprise architecture which comprises the layers of *enterprise plan*, *business process model* and *specification of resources* (figure 1). The enterprise plan constitutes an outside perspective on an enterprise. It focuses on the global enterprise task and the resources to fulfil this task. The business process model constitutes an inside perspective on an enterprise, specifying the tasks and task relations collectively carrying out the global enterprise task. Thus, a business process model can be considered as a procedure for executing the enterprise task. Finally, the third layer specifies the resources needed to fulfil the business processes, particularly personnel for the execution of non-automated tasks and business application systems for the execution of automated tasks.

[1] ADOxx is a registered trademark of BOC AG

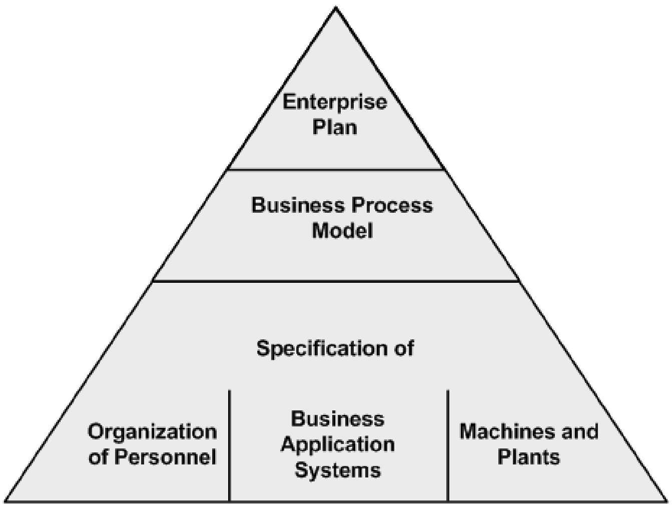



**Figure 1: Enterprise Architecture of the SOM methodology (Ferstl and Sinz 2006)**

This paper concentrates on the middle layer of the SOM enterprise architecture, the business process model. A SOM business process model is specified according to a graph-based multi-view approach. Based on an integrated internal representation, two different diagrams, an *interaction schema (IAS)* and a *task-event schema (TES)*, representing a structural view and a behavioural view on the business process model, are defined.

In contrast to other modelling approaches which focus on the drawing of a specific diagram at a particular time, the SOM methodology utilises an integrated kind of modelling. At any time, the different views are derived from an integrated model. Moreover, the decomposition of the different artefacts of a business process model is an integral part of the model, too. This characteristic allows navigation through the model by zooming in and zooming out the areas considered at a particular time.

Over the last two decades several software tools supporting the SOM methodology have been developed (e.g. Ferstl et al. 1994). These tools helped to utilise the SOM methodology and enabled its application even to industry-sized projects. However, based on native software platforms like C++, these tools were not suitable to provide enduring availability. To overcome this shortness, it was decided to use a specialised software platform for tool development, allowing high software productivity, easy adaption and extension of the methodology, and integration into a tool family sharing the same platform as well as bridging different methodologies.

This article reports on the design of a new SOM tool which is aimed to meet the requirements mentioned above. The tool is based on the ADOxx[2] meta-modelling platform. After a short introduction to SOM business process modelling and the ADOxx platform, the design of the software tool is outlined by means of mapping the SOM meta-model to the ADOxx meta-meta-model, graph-based visualisation of the multiple views on a business process model as well as tool functions and modelling transactions.

[2] ADOxx is a registered trademark of BOC AG

The paper is organised as follows: Chapter 2 gives a brief introduction to basic concepts of the SOM methodology, especially to SOM business process modelling. Chapter 3 shortly introduces the meta-modelling platform ADOxx. The core of the paper is chapter 4, outlining the design of the SOM business process modelling tool on the ADOxx platform. The paper ends with some conclusions and an outlook to future work.

## 2 SOM Business Process Modelling

The meta-model for SOM business process modelling is shown in figure 2. According to this meta-model, a SOM business process model consists of a set of business objects, each belonging either to the considered business system (symbol: rectangle) or to its environment (symbol: oval). A business object encapsulates one to many tasks sharing common states and pursuing joint goals. A task drives one to many transactions (symbol: arrow), each of them driven by exactly two tasks belonging to different business objects. Each transaction either transmits goods or services from one business object to another or it participates in coordinating business objects or other transactions. Related tasks within a business object are coupled by internal events (symbol: circle). External events model occurrences in the environment of a business system (e.g. “the first day of a month”).

The SOM methodology for business process modelling utilises two different coordination principles (Ferstl and Sinz 2006).

- According to the *negotiation principle* a transaction is decomposed into a sequence of three transactions: (1) An initiating transaction $T_i$, where the objects learn to know each other and exchange information on deliverable goods or services, (2) a contracting transaction $T_c$, where both objects agree to a contract on the delivery of goods or services, and (3) an enforcing transaction $T_e$ where the objects transfer the goods or services. The negotiation principle can be formally specified as

  $T(O,O') ::= [\ [\ T_i(O,O')\ \text{seq}\ ]\ T_c(O',O)\ \text{seq}\ ]\ T_e(O,O').$

- According to the *feedback control principle* a business object O is decomposed into two sub-objects and two transactions: A management object O’, an operational object O’’ as well as a control transaction $T_r$ from O’ to O’’ and a feedback transaction $T_f$ in the opposite direction. These components establish a feedback control loop, specified by a set of components:

  $O ::= \{\ O', O'', T_r(O',O'')\ [\ , T_f(O'',O')\ ]\ \}.$

The structural view (IAS) and the behavioural view (TES) are defined by projections onto the meta-model (figure 2). Except for the notion of business transaction the views are disjoint. Business transactions appear in both views representing the structural aspect (communication channel) and the behavioural aspect (event) respectively. Beyond that, decomposition of business objects and transactions establish two further views on a business process model.

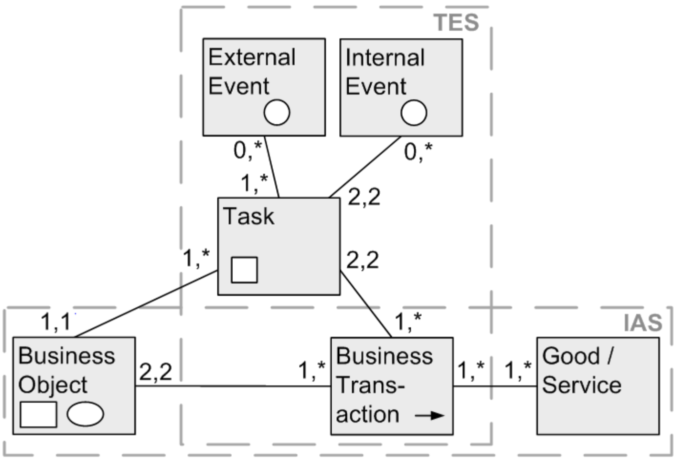
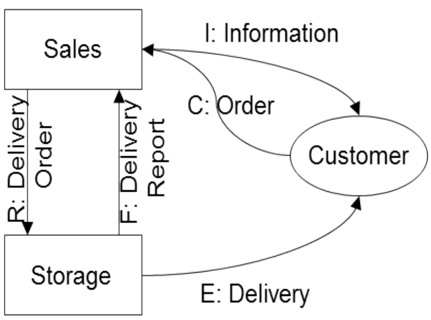



**Figure 2: Business Process Meta-model of the SOM methodology (Ferstl and Sinz 2008)**

To give an example, figures 3 and 4 demonstrate the IAS and TES of the simple business process model of a trading company. An initial transaction *distribution of goods* from *trading company* to *customer* has been decomposed according to the negotiation principle into an initiating transaction *information*, a contracting transaction *order* and an enforcing transaction *delivery*. The trading company itself has been decomposed applying the feedback control principle into a management object *sales*, an operational object *storage*, a control transaction *delivery order* and a feedback transaction *delivery report*. These two decompositions result in the IAS depicted in figure 3. Figure 4 shows the corresponding process flow by means of a TES. The TES specifies the sequence of the tasks obtained from the decompositions named above. The *information* transaction is performed by the tasks *I>* ("send information") of the business object *sales* and *>I* ("receive information") of the business object *customer*. After customer has received necessary information, he/she is able to send an *order* (task *C>*) to *sales* and so on.



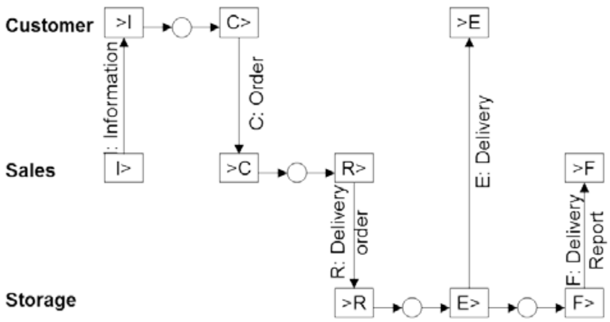




**Figure 3: SOM Interaction Schema (IAS)** **Figure 4: SOM Task-Event Schema (TES)**

There are several well-established languages which are widely used for business process modelling. To clarify the specific characteristics of SOM, some key properties of SOM business process modelling are contrasted to Event-driven Process Chains (EPC) (Scheer 2000) and Business Process Model and Notation (BPMN) (Allweyer 2008, White and Miers 2008).

While EPC focus on the behaviour of a business process, SOM focuses both on structure (IAS) and behaviour (TES). The comprehension of a structural view allows modelling of the transaction-based coordination of business objects. Furthermore, SOM allows hierarchical decomposition of business process models by refining business objects and business transactions.

As pointed out in Pütz and Sinz (2010), BPMN is rather a language for workflow modelling than a language for business process modelling. While a business process model refers to related business tasks producing some goods or services, a workflow model specifies a procedure for the execution of one or more business tasks, carried out by human or machine actors (persons or application systems). Nevertheless, the SOM methodology allows model-driven derivation of an initial BPMN workflow schema from a decomposed SOM business process model.

## 3 The ADOxx meta-modelling Platform

ADOxx is a meta-modelling platform which is aimed to facilitate design and implementation of modelling tools for domain specific languages (DSL). The platform has been developed by the BOC-Group[3], a spin-off of the University of Vienna. Over the last decades, ADOxx has been used to implement modelling tools for a wide area of domains like e-learning, knowledge management, strategic management and many others more (Schwab et al. 2010, Fill 2005, Lichka et al. 2002, Karagiannis and Bajnai, Karagiannis and Telesko 2000, Junginger et al. 2000).

Using a meta-modelling platform for the new SOM tool leads to less effort for implementation and maintenance and thus helps to provide enduring availability. Currently, there exist several meta-modelling platforms which can be considered as candidates for the SOM tool under design (e.g. Graphical Modeling Framework (GMF)[4] which is based on the Eclipse Modeling Framework (EMF) and the Graphical Editing Framework (GEF)). An in-depth comparison of the features of these different meta-modelling platforms would need a paper of its own. Besides other criteria, the choice of ADOxx as a meta-modelling platform is motivated by the fact that ADOxx is the heart of the Open Model Initiative[5] (OMI), which is aimed at bringing together different modelling tools and establishing a community of modelling experts and model users.

From a technical perspective, ADOxx provides the designer of a modelling tool with basic functions for representation and editing of diagrams, persistent storage of models, simulation and evaluation of models as well as import/export of models. To utilise these functions, a tool designer has to map the meta-model of the DSL onto the meta-meta-model of ADOxx. In other words, the meta-model of the DSL has to be specified using the concepts provided by the ADOxx meta-meta-model. In this way the ADOxx platform allows efficient design and implementation of both powerful and flexible DSL modelling tools.

---

[3] http://www.boc-group.com/de/, last visit: 16.11.2010
[4] http://www.eclipse.org/gmf/, last visit: 16.11.2010
[5] http://www.openmodels.at/web/omi, last visit: 16.11.2010

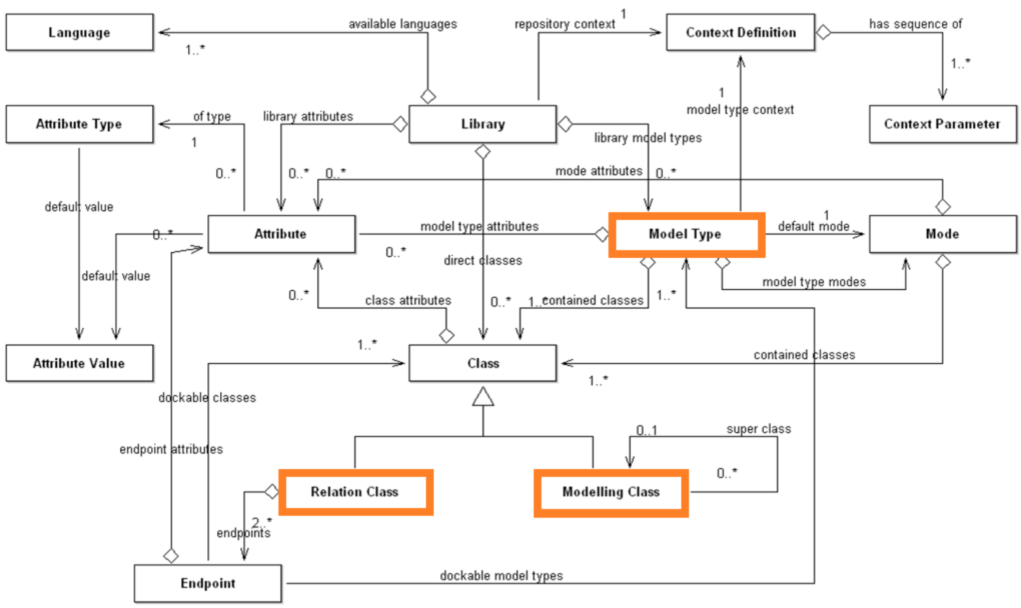



**Figure 5: Extract of the ADOxx meta-meta-model (Junginger et al. 2000)**

The ADOxx platform fosters the concept of meta-modelling which is widely used in the field of modelling (see e.g. OMG Meta Object Facility Object Management Group 2006). Within a hierarchy of meta levels a model (schema) at level 1 represents an instance of a corresponding meta-model at level 2 (Ferstl and Sinz 2008). The meta-model in turn is an instance of a meta-meta-model at level 3. Conversely, several meta-models which comply with a given meta-meta-model usually can be specified and many schemata may meet a given meta-model.

On level 3 the ADOxx platform provides a meta-meta-model (often also called meta²model) defining some generic modelling classes and relations as well as corresponding attributes and constraints. The meta-meta-model is implemented in C++ and cannot be modified by a tool developer. The core of the ADOxx meta-meta-model is shown in figure 5. The most important concepts, *Modelling Class*, *Relation Class and Model Type* are highlighted.

To implement a specific DSL on the ADOxx platform, the meta-model of the DSL has to be specified as an instance of the meta-meta-model. Thereby the concepts of the meta-model have to be mapped to those of the meta-meta-model (e.g. an activity of the BPMN language is mapped onto the concept modelling class). After this step has been finished, the meta-model of a methodology is defined on ADOxx. The tool is now ready to create simple diagrams, meaning that objects can be placed and connected by arcs. For a methodology like SOM, the more important and challenging step is the implementation of the tool functions (e.g. refinement of artefacts, model visualisation, zooming) and modelling transactions (e.g. compliance with decomposition rules, model consistency). For this purpose the scripting language AdoScript is provided by the platform.

# 4 Design of a SOM Business Process Tool on ADOxx

The design of a modelling tool for SOM business process models within the ADOxx meta-modelling platform consists of three major steps. First, the meta-model for SOM business process models is mapped to concepts provided by the meta-meta-model of ADOxx (section 4.1). The result of this step is a representation of the SOM meta-model as an instance of the ADOxx meta-meta-model. In the second step the visualisation of the models is conceived using a graph-based multi-view approach (section 4.2). The third step comprises the design of the tool functionality including the concept of modelling transactions (section 4.3). A modelling transaction is perceived as a sequence of editing operations which transform a consistent state of the model into a new state, which again is consistent according to syntax and semantics.

## 4.1. Meta-model Instantiation

As already mentioned, the first step in creating a modelling tool on ADOxx is to specify the domain-specific meta-model as an instance of the meta-meta-model provided by the platform (figure 5). In case of the SOM business process meta-model (figure 2) the instantiation is basically defined as follows:

<table>
<tr><th>ADOxx Meta-meta-model</th><th>SOM Meta-model</th></tr>
<tr><td rowspan="2">Modelling Class</td><td>Business Object</td></tr>
<tr><td>Task</td></tr>
<tr><td rowspan="2">Relation Class</td><td>Business Transaction</td></tr>
<tr><td>External/Internal Event</td></tr>
<tr><td rowspan="4">Model Type</td><td>Interaction Schema</td></tr>
<tr><td>Task-Event Schema</td></tr>
<tr><td>Object Decomposition</td></tr>
<tr><td>Transaction Decomposition</td></tr>
</table>

**Table 1: Meta-model instantiation**

Furthermore, the first step requires a specification of the syntax of the modelling language, particularly the feasible connections between classes. Finally the attributes (*AttrRep*) and a graphical representation (*GraphRep*) of the classes have to be defined.

As mentioned before, a SOM business process model consists of an internal representation and several corresponding views. The views are diagrams, representing an interaction schema, a task-event schema as well as the decomposition of business objects and business transactions. Both the internal representation of a model and the views require the definition of each a model type in ADOxx. A model type specifies a set of classes and relations between classes.

Having finalised these specifications, a modeller is able to create diagrams by dropping model elements on the drawing board and linking them by arcs. It is worth mentioning, that at this point the implementation of a tool for a common modelling language on ADOxx is done. However, SOM business process modelling is a sophisticated methodology which requires some more effort for tool implementation. The following sections give an overview of selected tool features which establish the difference between the way of SOM modelling and the widespread *one diagram at a time* and *drag and drop modelling*. These tool features are implemented using the scripting language AdoScript provided by the ADOxx platform.

## 4.2. Graph-based multi-view visualisation

A SOM business process model is represented by several complementary views derived from one integrated internal model. The model-view-controller (MVC) paradigm (Reenskaug 1979) has been approved to be a suitable design pattern for such a requirement. For the design of the SOM modelling tool the MVC paradigm is utilised to propagate any action the modeller performs on a particular view to the internal model representation and from there to all other views as far as these views are concerned by these actions. For example, changing the name of a business object in one view causes changing the name in all other views showing the same business object. Consistency between objects and views has to be specified in AdoScript.

For the visualisation of SOM business process models three different types of diagrams, each corresponding to an ADOxx model type, have been designed and implemented on the platform:

1. *Decomposition diagram*
   Decomposition of both business objects and business transactions is represented using a tree-based graph. This style of visualisation helps the modeller to quickly recognise the recursive decomposition of objects and transactions. The decomposition of objects is displayed on the upper right side, the decomposition of transactions is shown on the upper left side of the tool window. Within a decomposition diagram, no information about the connections of business objects and business transactions is displayed. This information is subject of the following two diagrams.

2. *Structure diagram*
   This type of diagram is used to represent the structural view on a SOM business process model, the interaction schema (IAS). An IAS consists of business objects which are connected by business transactions. The IAS is displayed on the lower left side of the tool window.

3. *Behaviour diagram*
   Complementary to the structure diagram, this type of diagram shows the behavioural view on a SOM business process model by a task-event schema (TES). A TES consists of tasks, each belonging to a business object. Tasks are connected by internal events (if the tasks belong to the same business object) or business transactions (if the tasks belong to different business objects). The TES is displayed on the lower right side of the tool window.

Figure 6 illustrates the graph-based multi-view visualisation of SOM business process models on the ADOxx platform.

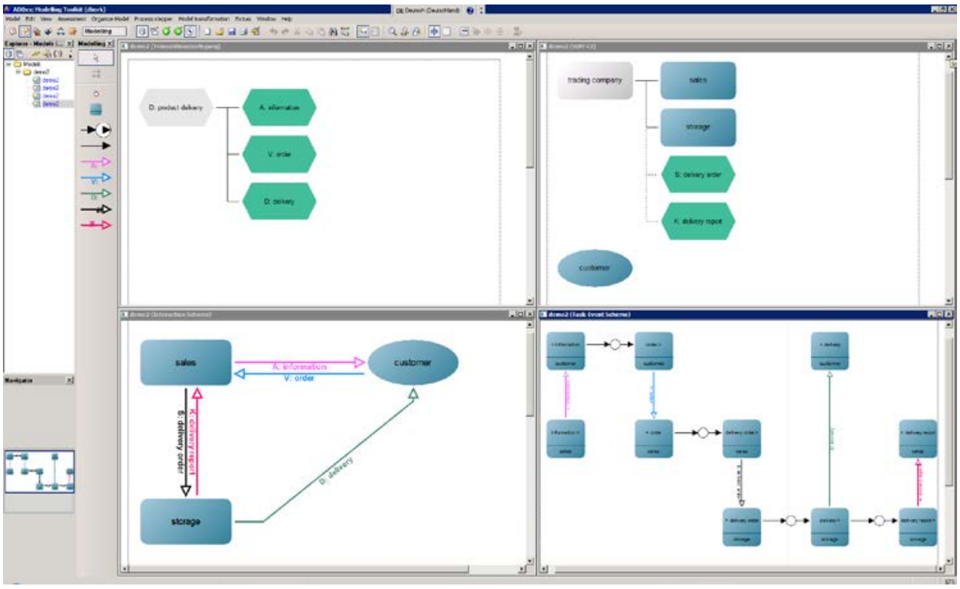


**Figure 6: Graph-based multi-view visualisation in ADOxx**

With increasing model size, comprehension of a SOM business process model, presented by four different views, can easily overstrain the modeller. Therefore, the visualisation of the model has been improved effectively using different techniques.

First, any object or transaction that is currently not visible in the IAS and TES will be grey-shaded in the object and transaction decomposition window. This retains the modeller from getting lost when handling large models. Furthermore, a red border is drawn around any object that is selectable in the process of reconfiguring transactions to new objects which are resulting from the decomposition of a given object (section 4.3 for a detailed description of the tools' functions).

Additionally, two layout algorithms have been implemented in order to optimize the positioning of business objects and the routing of the business transactions in the interaction schema:

- *Auto-Layout*
  The auto-layout algorithm draws every business transaction either directly, if the connected business objects are placed on the same vertical or horizontal position or with a right angle otherwise. Any additional ingoing or outgoing transaction is shifted a bit to the left or right alternatingly in order to prevent overlapping transactions.

- *Smooth Edges*
  Calling this algorithm ensures, that any business transaction is drawn on the most direct way between the connected business objects. Again, multiple ingoing or outgoing transactions are shifted a bit to the left or right.

Besides the discussed algorithms, the modeller is able to optimize the positioning of modelling elements and relations using the functionality provided by the platform (e.g. adding an edge to a relation or moving of elements with the associated relations on the modelling area using the mouse or keyboard).

### 4.3. Tool Functions and Modelling Transactions

As any modelling tool, the SOM tool provides a set of functions via its user interface. In many cases, a modelling step requires the execution of a sequence of functions. Such a sequence, transforming a consistent state of the model into a new state, which again is consistent with respect to syntax and semantics, is perceived as a modelling transaction. In the following, tool functions as well as modelling transactions are illustrated by means of use cases. Each use case refers to a typical scenario a modeller carries out when using the tool.

- *Decomposition of business objects and business transactions*
  The SOM methodology comprises a set of rules for the decomposition of business objects and business transactions (chapter 2 for selected rules). The rules can be applied recursively in order to refine a business process model thereby revealing its coordination (e.g. the decomposition of the initial enforcing transaction *delivery of goods* shown on the upper left side of figure 6). Depending on the currently selected business object or business transaction, the tool displays the decomposition options available to the modeller. Decomposition of objects and transactions can be triggered from any diagram.

- *Reconfiguration of relationships within a business process model*
  After decomposing a business object or a business transaction, the modeller must reconfigure the new objects or transactions in order to adjust the IAS or TES to the more detailed level of the business process model. The tool provides some guidance for the reconfiguration process by means of highlighting the considered business objects and business transactions. Furthermore, the tool ensures a consistent model state after reconfiguration has been completed. The reconfiguration of relationships has to be performed within the IAS or TES schemata.

- *Navigation within a business process model*
  Within the hierarchy of business objects and business transactions, the modeller is able to navigate. The navigation is supported by a zoom operator, meaning to perform no changes on the integral model. Navigation enables browsing through a large business process model in order to retain overview. Navigation has to be performed using the context menu of business objects and business transactions within the decomposition diagrams.

A typical modelling transaction consists of the decomposition of some business object or business transaction followed by a reconfiguration step to adjust the IAS and TES to the new refinement level. The SOM tool supports modelling transactions by providing guidance for the modeller through the steps necessary to transform a consistent state of the model to a new one.

Generally, usage of the tool is dialog-driven and guided by the context menu of an object or a transaction. In contrast to other modelling tools, drag & drop technique is utilised scarcely. The only case in which the modeller draws a relation between two modelling elements is when adding internal events to a task-event schema in order to determine the behaviour of the business process. Figure 7 illustrates the dialog-driven procedure of reconfiguring the relations on a more detailed level of the business process model. As reconfiguration has no effect on the object decomposition view as well as the transaction decomposition view, only IAS and TES are displayed.

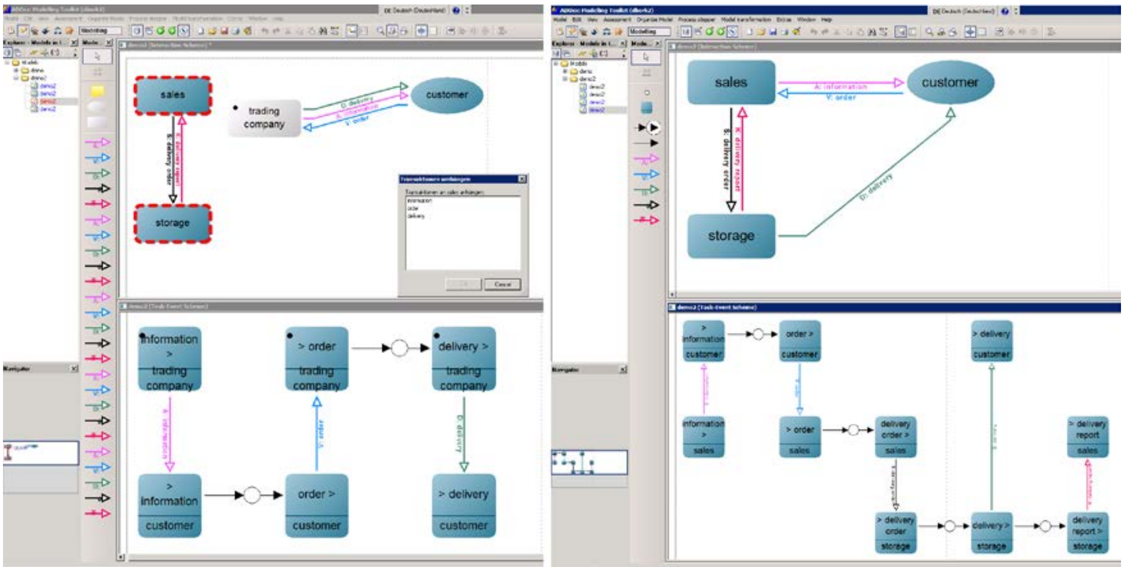


**Figure 7: Reconfiguring the relations within a business process model**

On the left side the modeller has decomposed the business object *trading company* into the objects *sales* and *storage* according to the feedback control principle (chapter 2). The decomposed trading company object is still displayed on the left side, but grey-shaded to give the modeller a hint about which objects are going to be removed and which transactions he has to reconfigure using the new objects. The dialog box in the middle of the left window displays the transactions currently not considered and asks the modeller alternatingly which transactions he wants to reconfigure to the currently regarded object. This process is done until all transactions are related to either *sales* or *storage*. The objects selectable while connecting the transactions are highlighted using a dashed edge around them.
On the right side then, all transactions are reconfigured and the procedure ends with an update of the IAS and TES diagrams using the implemented graph algorithms. Finally, the dashed edges around the objects sales and storage are removed.

## 5 Conclusion and future work

This article outlines the design of a new software tool for SOM business process modelling based on the meta-modelling platform ADOxx. The focus is on how the ADOxx platform can be used to establish a graph-based multi-view visualisation of comprehensive business process models based on an integrated meta-model. The paper is research-in-progress. A first tool prototype is available within the Open Model Initiative. After refinement and user feedback have led to a mature version, the tool will be primarily used in higher education and research.

Next steps will concentrate on the third layer of the SOM enterprise architecture, namely the functional specification of business application systems. In the SOM methodology, application systems are specified from a functional viewpoint by two complementary schemata: a s*chema of task classes (TAS)* which refers to the workflow of an application system and a s*chema of conceptual classes (COS)* which provides corresponding business functions. Both TAS and COS can be derived initially from a SOM business process model via a model-driven approach. Because of its wide acceptance, BPMN is a candidate language for workflow schemata. Pütz and Sinz report on model-driven derivation of BPMN workflow schemata from SOM business process models (Pütz and Sinz 2010).

## References


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

Ferstl OK, Sinz EJ (2006) Modelling of business systems using SOM. In: Bernus Peter, Mertins Kai, Schmidt Günter (Hrsg) Handbook on Architectures of Information Systems, second edition. Springer, Berlin, S. 347–367

Ferstl OK, Sinz EJ (2008) Grundlagen der Wirtschaftsinformatik, 6th, revised and extended edition. Oldenbourg, München

Ferstl OK, Sinz EJ (1995) Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen. Wirtschaftsinformatik 37(3):209-220

Ferstl OK, Sinz EJ (1996) Multi-Layered Development of Business Process Models and Distributed Business Application Systems. An Object-Oriented Approach. Distributed Information Systems in Business. Springer, Berlin:159-179

Ferstl OK, Sinz EJ, Amberg M, Hagemann U, Malischewski C (1994) Tool-Based Business Process Modeling Using the SOM Approach. In: Wolfinger B. (Hrsg) 24. GI Jahrestagung 1994. Innovationen bei Rechen- und Kommunikationssystemen, S. 430–436

Fill H (2005) UML Statechart Diagrams on the ADONIS Metamodeling Platform. Electronic Notes in theoretical Computer Science (ENTCS)(127):27–36

Junginger S, Kühn H, Strobl R, Karagiannis Dimitris (2000) Ein Geschäftsprozessmanagement-Werkzeug der nächsten Generation - ADONIS: Konzeption und Anwendungen. Wirtschaftsinformatik 42(5):392–401

Karagiannis D, Bajnai J ADVISOR® - An Educational Management Tool. submitted to the Symposium Towards the New Education Society, Zvolen, Slovakia

Karagiannis D, Telesko R (2000) The EU-Project PROMOTE: a process-oriented approach for knowledge management. In: Proc. of the Third Int. Conf. of Practical Aspects of Knowledge Management.

Lichka C, Kühn H, Karagiannis D (2002) ADOscore® - IT gestützte Balanced Scorecard. wisu - das wirtschaftsstudium(7):915–918

*Object Management Group* (2006) OMG Meta Object Facility (MOF) Version 2.0. http://www.omg.org/spec/MOF/2.0/, last visit: 2010-05-18

Pütz C, Sinz EJ (2010) Model-driven Derivation of BPMN Workflow Schemata from SOM Business Process Models. In: Enterprise Modelling and Information Systems Architectures, Vol. 5, No. 2, October 2010: 57-72

Reenskaug T (1979) Thing-Model-View-Editor, an example from a planningsystem. In: technical note, Xerox Parc

Scheer A (2000) ARIS - business process modeling, 3. ed. Springer, Berlin

Schwab M, Karagiannis D, Bergmayr A (2010) i* on ADOxx®: A Case Study. In: Proceedings of the 4th International i* Workshop

White SA, 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, Fla.