Business Model Interoperability using Enterprise Model Integration

Harald KÜHN, Marion MURZEK, Franz BAYER
BOC Information Systems GmbH, Rabensteig 2, 1010 Vienna, Austria
Tel: +43 1 513 27 36 10, Fax: + 43 1 513 27 36 28, Email: boc@boc-eu.com

Abstract: Managing complexity in developing enterprise-spanning applications is still a challenge. A vital field of delivering technical concepts and technologies to support interoperability in business systems is the area of Enterprise Application Integration (EAI). In modelling and developing large enterprise applications, it is necessary to integrate applications on the business and conceptual level as well. Nevertheless, standardisations on modelling languages and model exchange formats (incl. their industrial implementation) still can rarely be found. Because of today's diversity of models and heterogeneous modelling languages in the area of business modelling, we apply the Enterprise Model Integration (EMI) approach. This paper focuses on the model transformation part of EMI supporting business model interoperability. A technical overview of the BOC Model Transformer (BMT) is given and business benefits and their measurements are explained. A brief case study shows the application of the transformation approach.

1. Introduction

Managing complexity in developing enterprise-spanning applications is still a challenge. Amongst others, a vital field of delivering technical concepts and technologies for integrating heterogeneous applications and components to support interoperability in inter-organisational business systems is the area of Enterprise Application Integration (EAI). The main idea of EAI is to provide technical solutions to integrate workflows and heterogeneous parts of enterprise applications in a continuous business application [6].

In modelling and developing large enterprise applications, it is necessary to integrate applications on the business and conceptual level as well. Nevertheless, standardisations on modelling languages and model exchange formats (incl. their industrial implementation) still can rarely be found. The most prominent example is the UML and XMI for modelling object-oriented systems [11, 12]. Other examples from industrial research areas are BPML/BPMN [2, 3] and XPDL [15]. In addition, the OMG community started to establish a relatively new vision with the Meta Object Facility (MOF) and the Model Driven Architecture (MDA) to improve productivity in software development, applying object-orientation, meta level concepts and modelling [13, 14].

Because of today's diversity of models and heterogeneous modelling languages in the area of business modelling, we apply the Enterprise Model Integration (EMI) approach [7]. This approach consists of three major elements: (a) object-oriented metamodelling technology to describe context-specific, integrated modelling languages, (b) model transformation concepts for model exchange and model reuse, and (c) a process model describing the steps to use metamodelling technology and model transformations in the area of business systems development.

The remainder of the paper focuses on element (b) and is organised as follows: In chapter 2 levels in business modelling are explained and a classification schema for business modelling approaches are presented. Chapter 3 describes the BOC Model

Transformer (BMT) according to its architecture and its transformation rules. Chapter 4 provides an overview of business benefits using business model interoperability. In chapter 5 a brief case study applies the presented transformation concepts and shows the business benefits. Finally, chapter 6 gives an outlook to future work and developments.

2. Business Modelling Classification

In the following, a business model is defined as a model describing a certain part of a corporation from a business point of view. Typical business models are strategy models, goal models, product models, business process models, organizational models etc. As interoperability between business models we understand the consistent integration and/or transformation of business models.

To classify business modelling approaches we propose several classification criteria (see figure 1). In accordance to [14] we use following model categories: Computation Independent Model (CIM), Platform Independent Model (PIM) and Platform Specific Model (PSM). In each category we distinguish three abstraction levels: meta-metamodel, metamodel and model. Concerning the adaptability each modelling approach can be classified as open, extensible, adaptable or not adaptable. If a business modelling approach focuses on a certain branch, it is called branch-specific, otherwise branch-independent. To enable quick start modelling, some modelling approaches provide reference models.

Characteristic	Value			
Model category	CIM	PIM	PSM	
Abstraction Level	Meta-Metamodel	Metamodel	Model	
Adaptivity	open	extensible	adaptable	none
Branch	specific	independent		
Reference model	existing	not existing		

Figure 1: Classification schema for business modelling approaches

3. BMT: BOC Model Transformer Approach

The presented approach focuses on the level of model-to-model transformation. In particular it applies to business model transformations on the CIM level. In the face of modelling semantics models within the PIM level are often similar. All of them try to represent software on a "virtual" platform. Compared to models on the PIM level, business modelling approaches are often based on very different semantics. So the elementary problem at the CIM level is the transformation of the varying semantics in different business modelling languages [14]. This chapter will give an overview of existing approaches, the BMT approach and some concrete rules used for business model transformations.

3.1 – Related Work

Recently, a large number of model transformation approaches were proposed. Additionally, the OMG issued in 2002 a Request for Proposal (RFP) on Query/Views/Transformations to establish standardized model views, model transformations and model synchronizations. This RFP further pushed the number of model transformation approaches [5]. [4] provide an extensive overview of model transformation approaches and categorizes into model-to-code approaches and model-to-model approaches.

3.2 –BMT Approach

As mentioned above, the central problem of business model transformations is the semantic transformation. Due to this fact the BMT is based on an adaptable rule file which uses a (proprietary) transformation language. This language consists of semantic independent constructs to allow many different kinds of transformations. The only precondition is that all models must conform to (different) metamodels, which on their part must conform to the same meta-metamodel (meta^2 model). This fact is shown in figure 2 [9]. The transformation engine is the heart of the transformation process. It supports the transformation language, knows the "physical" structure of the model file and interprets the rule file. The history shown in figure 2 is needed for saving the relations between source elements and the related target elements. The contained information is used to preserve the original relations of the source models in the target models.

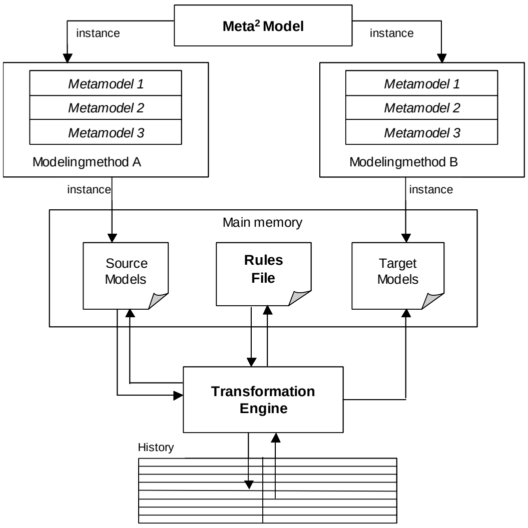

Figure 2: Architecture of the BMT

The transformation process as shown in figure 2 starts with the import of the source models and the rule file. Also an empty target model file is created. Based on the instructions in the rule file the transformation engine matches elements in the source models. Then the selected parts of the source will be transformed according to the rules. Within the scope of this step the transformation engine looks up for former transformations registered in the history, to create the proper target element in the target models. The next section shows rules for better understanding the process of the transformation.

3.3 – Definition of Transformation Rules

The rules for transformation have to be generic to allow any kind of semantic transformation. The rules used in the transformation language of the BMT can be categorized as follows:

- Rules for copying fragments like copy-model, copy-instance, copy-relation, copy-attribute etc.
- There are two kinds of rules for creating fragments. One type with reference to source elements and the other without (new elements). For example create-model, create-instance, create-relation, create-attribute etc.

Beside these rules additional constructs are needed to define transformations. These are navigation elements, to navigate in the source and target models. Conditions are used to constrain navigations and rules. Definitions hold values for recurrent use and functions provide the possibility to manipulate values. A further language element are post processing actions, these actions extend the rules and are processed at the end of the transformation. Typical postprocessing actions are sort actions and arrange actions to generate position information to arrange model content properly [8, 9].

4. Business Benefits of Business Model Interoperability

Sometimes, enterprise and business modelling approaches are criticised in means of:

- business modelling effort can be noticeable in comparison to gained results,
- modelling effort and corresponding benefits cannot be adequately measured, and
- results in business modelling cannot be easily reused in different application domains.

From our experiences in modelling large business systems, these objections are not valid if modelling goals are clearly defined, modelling efforts and associated investments are properly planned, modelling progress is continuously controlled, and model reuse potential is continually checked. Therefore, selected benefits and interrelated measurements in the context of business modelling are presented. The benefits are structured in two categories: (a) modelling benefits and (b) interoperability benefits.

Benefit	Description (D) and Measurement (M)
Modelling Benefits	
Complexity handling	D: Large business systems cannot be adequately handled by only using “pencil and paper”, word processors or spread sheets. A consistent, interrelated representation of all necessary business system elements is needed. M: Number of business processes, number of organizational resource units, number of connections and connection points between processes and units etc.
Maintenance support	D: Maintenance of large business systems often consumes enormous amount of resources because of hidden dependencies. Every improvement in maintenance tasks safe effort in time and money. Model representations help to make hidden dependencies visible. M: Number of system releases per year, ratio of specified system elements to unspecified business system elements etc.
Precise requirements definition	D: Business models represent (semi-formal) requirement definitions from a business point of view. Instead of textual descriptions, models help to formalize requirements and to enhance preciseness. M: Ratio of (semi-formal) modelled requirements to requirements textually specified, number of (new) requirements after deployment of business system etc.
Completeness of requirements definition	D: Modelling different perspectives of a business system such as goals, strategies, products, business processes, human and technical resources etc. raises noticeable the completeness and the quality of the system specification. M: Ratio of modelled perspectives to necessary perspectives, ratio of released specifications to number of total specifications etc.
Risk management	D: Based on the modelled perspectives a precise estimation of implementation effort (time, costs etc.) can be made. Therefore, business models play an active role in risk management in the domain of business systems development. M: Ratio of estimated costs to available project budget, ratio of real project cycle time to planned project cycle time etc.
Interoperability Benefits	
Enlarged usage area	D: Using model transformations, already existing business models can be reused for different application areas such as business process optimization, activity based costing, benchmarking, requirements definition, test specifications etc. M: Number of applications scenarios of model without model adaptation, total number of application scenarios of model incl. definition of transformation rules, number of saved person days avoiding re-work and re-modelling etc.
Viewpoint integration	D: Transforming model information and to ensure interoperability of different parts of a business model enables the integration of different interest groups into the process of business modelling. This raises the quality of business models and the acceptance of the modelling results in a corporation. M: Number of different viewpoints, number of necessary coordination meetings for final acceptance etc.
Integrated development process	D: This benefit is closely related with the enlargement of usage area. Using model transformations, results from different phases of a development project can be easily

	passed to and reused in other project phases. Therefore, the phases of the development process reach a better integration. M: Amount of person days saving re-work, cycle time from requirements definition to deployment of business system etc.
Adequate tool landscape	D: Normally, a set of different tools is used in developing business systems. Each tool is specialized for a certain task such as specification, coding, testing, documentation etc. Using model transformations a proper integration of the tool landscape can be reached by exchanging models with minimized loss of model information. M: Number of transformation interfaces, number of transformation directions, percentage of information loss between each transformation step etc.
Computable representation	D: Formal and semi-formal specifications of business models can be either directly processed by software or enriched to be processed by using transformation mechanisms. Herewith, the level of abstraction in executable specifications in the domain of business systems development can be raised. M: Ratio of computable business models to non-computable business models, number of levels in chosen business modelling approach (e.g. CIM, PIM, PSM and code in MDA-based approaches) etc.

Table 1: Business benefits in business model interoperability

5. Case Study

This chapter describes an application of BMT in the context of IT system consolidation.

5.1 – Case Description

A multi-national corporation integrated several companies via acquisitions resulting in a mix of three heterogeneous billing systems. To consolidate the amount of heterogeneous systems and to reach economies of scale, a pilot project to merge the billing systems into a single system is launched. The starting point for specifying the businesses supported by the heterogeneous billing systems build the already available business process models. Nevertheless, these business process descriptions are based on two different business process modelling languages. To enable the integrated usage of these models, a model transformation approach was chosen. The next sections describe the application of BMT in this context. As modelling environment BOC's meta business process management tool ADONIS is used.

5.2 – Delta Analysis of Business Modelling Languages

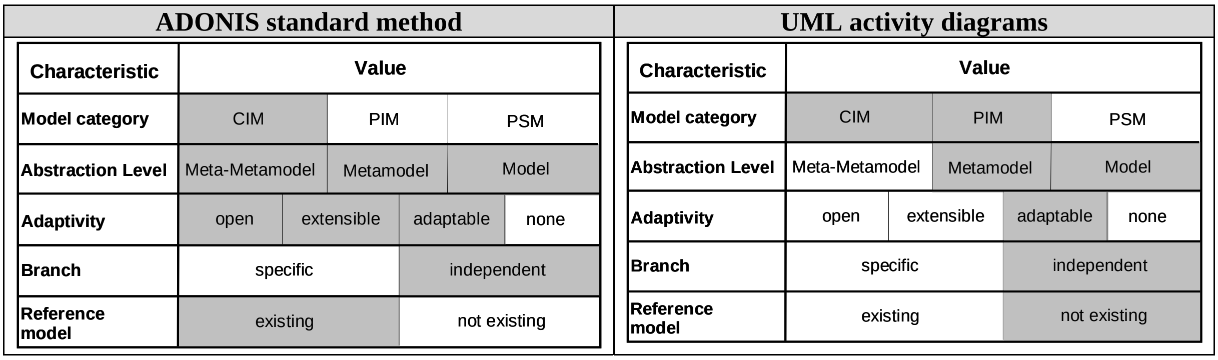

Figure 3: Classification of ADONIS standard method and UML activity diagrams

The first step is to analyse the differences between the two business process modelling languages. Therefore the underlying meta models of each language were compared. Table 2 shows some characteristics of the business modelling languages UML activity diagram [11] and ADONIS standard method [1].

To get the complexity of the whole transformation specification under control it is recommended to divide it into smaller parts. For example, differing on the level of detail, e.g. model, class, relation, and attribute. Additionally, the interdependencies between all elements must be considered.

The following table is a result of the analysis of the differences between the both business modelling languages.

ADONIS standard method	UML activity diagrams	Trivial mapping	Complex mapping
Process start	Start	x	
Activity	Activity	x	
-	Signals (send, receive)		x
Subsequent	Transition	x	
-	Object flow		x
Decision	Decision	x	
Transition condition	Guards	x	
Parallelity	Synchronization	x	
Merging	Join	x	
Resource	-		x
Variable	-		x
Swimlanes	Swimlanes	x	
Process end	Stop	x	

Table 2: Extract of result table of the comparison of ADONIS standard method vs. UML activity diagrams

5.3 – Business Process Model Transformation

This section shows an example of a transformation rule for the element "Signal" in the UML activity diagram to an activity with two extended attributes "send" and "receive" in ADONIS standard method.

UML to ADONIS

```
<CLASS name=”Send-Signal”>
  <RULE type="create-instance">
   <PARAM name="Instanceclass">Activity
   </PARAM>
  </RULE>
  <RULE type="create-attribute">
    <PARAM name="to">send</PARAM>
    <PARAM name="type">ENUMERATION</PARAM>
    <PARAM name="value">yes</PARAM>
  </RULE>
</CLASS>

<CLASS name=”Receive-Signal”>
  <RULE type="create-instance">
   <PARAM name="Instanceclass">Activity
   </PARAM>
  </RULE>
  <RULE type="create-attribute">
    <PARAM name="to">receive</PARAM>
    <PARAM name="type">ENUMERATION</PARAM>
    <PARAM name="value">yes</PARAM>
  </RULE>
</CLASS>

<CLASS name=” Activity”>
  <RULE type="copy-instance"/>
</CLASS>
```

ADONIS to UML

```
<CLASS name=”Activity”>
 <IF>
  <COMPARE>
   <LEFT-VALUE>
    <NAVIGATE type="child-node">
     <PARAM name="element-name">ATTRIBUTE</PARAM>
     <PARAM name="attribute-name">name</PARAM>
     <PARAM name="attribute-value">send</PARAM>
      <select-value/>
    </NAVIGATE>
   </LEFT-VALUE>
   <CONDITION>=</CONDITION>
   <RIGHT-VALUE>yes</RIGHT-VALUE>
  </COMPARE>
  <THEN>
   <RULE type="create-instance">
    <PARAM name="Instanceclass">Send-Signal</PARAM>
   </RULE>
  </THEN>
  <ELSE>
   <IF>
    <COMPARE>
     <LEFT-VALUE>
      <NAVIGATE type="child-node">
       <PARAM name="element-name">ATTRIBUTE</PARAM>
       <PARAM name="attribute-name">name</PARAM>
       <PARAM name="attribute-value">receive</PARAM>
        <select-value/>
      </NAVIGATE>
     </LEFT-VALUE>
     <CONDITION>=</CONDITION>
     <RIGHT-VALUE>yes</RIGHT-VALUE>
    </COMPARE>
    <THEN>
     <RULE type="create-instance">
      <PARAM name="Instanceclass">Receive-
Signal</PARAM>
     </RULE>
    </THEN>
    <ELSE>
     <RULE type="create-instance">
      <PARAM name="Instanceclass">Activity</PARAM>
     </RULE>
   </ELSE>
```

	</ELSE> </CLASS>

Table 3: Example of model transformation rule fragment

Transform to UML:

The attributes "send" and "receive" must be added to the class activity on the meta model level. These attributes should have the attribute type `ENUMERATION` with a value domain "yes" and "no". Otherwise it is not possible to differ between the UML classes "Signal" and "Activity". To transform these classes, the values of these two attributes are compared with the string "yes". If one of these attributes is set to "yes" the appropriated rule will be executed as shown in table 3.

Transform to ADONIS:

In this direction the transformation is a trivial one. Each UML class "Send-Signal", "Receive-Signal" and "Activity" will be matched in the source models and transformed to an ADONIS activity in the target models. Additionally either one of the attributes "send" and "receive" is set according to the specific UML class or non of the both attributes are set, if it is about the UML class "Activity".

Example of Measurement:

The transformation of these classes can be executed in both directions without any information loss. That means, that for this part of model transformation the measurement "information loss" for the benefit "adequate tool landscape" is 0%.

6. Outlook and Future Work

Model-based business engineering approaches are getting more and more state-of-the-art in the domain of business systems development. All large IT companies already have integrated or have announced model-based components in their business systems development environments, e.g. IBM's WebSphere, Microsoft's BizTalk Server or SAP's NetWeaver. Additionally, standardization bodies such as OMG are working on standards such as MOF and MDA.

We believe, that there won't exist a single business modelling standard in the future and many corporations will us a mix of different business modelling languages due to the heterogeneity of business system environments. Therefore, model transformation approaches will play a crucial success factor for business systems development and for model interoperability to lead to durable business benefits. Additionally, we are convinced that interoperability is one of the central research areas for the next years in the domain of business engineering [10].

Acknowledgement

This work is partially supported by the Commission of the European Communities under the sixth framework programme (INTEROP Network of Excellence, Contract N° 508011, <http://www.interop-noe.org>). The authors are indebted to WP9 and WP11.

References

[1] BOC: ADONIS 3.7 - User Manual III: ADONIS Standard Modelling Method. BOC GmbH, May 2003.

[2] BPMI.org: Business Process Modelling Language - Specification Version 1.0. http://www.bpmi.org/bpml-spec.esp, access 28 June 2004.

[3] BPMI.org: Business Process Modelling Notation - Specification Version 1.0. http://www.bpmi.org/bpmn-spec.esp, access 28 June 2004.

[4] Czarnecki, K.; Helsen, S.: Classification of Model Transformation Approaches. OOPSLA’03, Workshop on Generative Techniques in the Context of Model-Driven Architecture.

[5] Gardner, T.; Griffin, C.; Koehler, J.; Hauser, R.: A review of OMG MOF 2.0 Query / Views / Transformations Submissions and Recommendations Towards the Final Standard. OMG, IBM, July 2003.
[6] P. Johannesson; B. Wangler; P. Jayaweera: Application and Process Integration - Concepts, Issues, and Research Directions. In: Brinkkemper, S.; Lindencrona, E.; Solvberg, A. (Eds.): Information Systems Engineering Symposium CAiSE 2000, Springer Verlag, 2000.
[7] H. Kühn; F. Bayer; S. Junginger; D. Karagiannis: Enterprise Model Integration. In: Bauknecht, K.; Tjoa, A M.; Quirchmayr, G. (Hrsg.): Proceedings of the 4th International Conference EC-Web 2003 - Dexa 2003, Prague, Czech Republic, September 2003, LNCS 2738, Springer-Verlag, pp. 379-392.
[8] H. Kühn; M. Murzek; F. Bayer: Horizontal Business Process Model Interoperability using Model Transformation. In: INTEREST'2004 Workshop at ECOOP 2004, Oslo, Norway, June 2004.
[9] M. Murzek: Methodenübergreifende Modelltransformationen am Beispiel von ADONIS. Diploma Thesis, University of Vienna, April 2004 (in German).
[10] NoE InterOp, Network of Excellence Interoperability. EU project, http://www.interop-noe.org, access 28 June 2004.
[11] Object Management Group: OMG Unified Modeling Language Specification, Version 1.4, September 2001. http://www.omg.org/cgi-bin/doc?formal/01-09-67.pdf, access 28 June 2004.
[12] Object Management Group: OMG XML Metadata Interchange (XMI) Specification, Version 1.2, Januar 2002. http://www.omg.org/cgi-bin/doc?formal/02-01-01.pdf, access 28 June 2004.
[13] Object Management Group: Meta Object Facility (MOF) Specification, Version 1.4, April 2002. http://www.omg.org/cgi-bin/doc?formal/02-04-03.pdf, access 28 June 2004.
[14] Object Management Group: MDA Guide, Version 1.0.1, 12. Juni 2003. http://www.omg.org/cgi-bin/apps/doc?omg/03-06-01.pdf, access 28 June 2004.
[15] Workflow Management Coalition: Workflow Process Definition Interface - XML Process Definition Language. Document Number WFMC-TC-1025, Document Status-Version 1.0 Final Draft. October 2002. http://www.wfmc.org/standards/docs/TC-1025_10_xpdl_102502.pdf, access 28 June 2004.