Modelling Interoperability: The Modelling Framework of BREIN

Robert Woitsch[1], Hannes Eichner[1], Vedran Hrgovcic[1]

[1] BOC Asset Management GmbH, Bäckerstraße 5,
1010 Vienna, Austria
{Robert.Woitsch, Hannes.Eichner, Vedran.Hrgovcic}@boc-eu.com

Abstract. In BREIN a modelling framework to integrate all project related modelling efforts has been defined. Based on the project requirements and a state of the art analysis the BREIN related modelling challenges were identified on three levels: Syntactical, semantical and contextual. This paper describes the concept of the BREIN Modelling Framework and how the identified modelling challenges are addressed. The contextual challenge is discussed in more detail, introducing how models may be used for the generation of ontologies and how modelling may profit from the use of ontologies.

Keywords: Business process modeling, Interoperability, Modelling framework

1 Introduction

BREIN is an IST project funded by the EU as Integrated Project in the 6th Framework Programme being part of the Advanced Grid Technologies, Systems and Services action line.

One of the goals of the BREIN project (IST-FP6-034556) [1] is to realise flexible, intelligent Virtual Organisations support, to significantly reduce the complexity of modern day business-to-business collaborations. Companies and enterprises of any size will be able to compete equally in a complex and demanding market by increasing the business capabilities.

In BREIN this will be demonstrated with two scenarios, namely the Virtual Engineering scenario and the Airport scenario. Both scenarios involve complex relations between different service-providers that interact with each other in order to deliver a service to the customer. The aim of BREIN is transferring these business collaborations onto the Grid. This means taking the existing business and its services, but changing the underlying technology by transferring it to the Grid. In order to succeed in doing this, it’s necessary to analyse the concrete business processes of the scenarios and to analyse the added value of the Grid. In a further step these business processes have to be mapped onto the Grid.

We also observe that business process model is commodity today. Business processes exist in every business field and they need consideration in the related IT-systems. Another observation is that creating software is more and more related to the configuration of systems and services than actually implementing them. The configuration is usually done with the help of models. We conclude that models are gaining importance in software engineering as well in the use of software systems.

A Modelling Framework capable of performing above mentioned tasks has been defined within the BREIN project.

This paper presents the approach taken to realize the BREIN Modelling Framework and to address the according modelling challenges.

The remainder of the paper is structured as follows: Chapter 2 discusses the aspects of the BREIN requirements concerning modelling and the resulting BREIN Modelling Challenge. Chapter 3 deals with the mayor technologies involved and the concept of the BREIN Modelling Framework. Finally chapter 4 explains how the challenges are addressed by the modelling framework and describes two components of the framework in more detail.

2 The Modelling Challenge in BREIN

As already mentioned modelling continuously gains importance for the development and configuration of software systems. This also counts for BREIN where models focusing on business aspects as well as various technical models like UML diagrams and ontologies have been created.

The aim is to integrate all these models into a single framework - the BREIN Modelling Framework [2]. This framework should support all modelling related tasks and integrate all modelling efforts of the project.

The BREIN Modelling Framework has been defined to enable the process-oriented requirements analysis of applications, the specification of the software architecture and the externalisation of expert know-how.

The Modelling Framework distinguishes between two levels of abstraction:

- Business Modelling Level
- IT Modelling Level

In addition to these two levels, a "Best Practise Roadmap" deals with knowledge management [3]. It should allow the knowledge transfer to new or external developers that do not belong to the project. This expert know-how will be made explicit and easily accessible through the roadmap which will be integrated with the rest of the modelling framework, as a lot of knowledge is available in form of models (e. g. UML diagrams of the architecture, interface descriptions, etc.).

Considering the BREIN scenarios that are characterized through complex supply chains and inter-organisational collaboration that should be automated, the requirements for the modelling methods used in the business modelling layer and the IT modelling layer, have been derived.

The BREIN Modelling Framework (1) has to deal with several modelling languages that have different focus (e.g. business or technical level) and (2) has to consider several languages within one modelling domain.

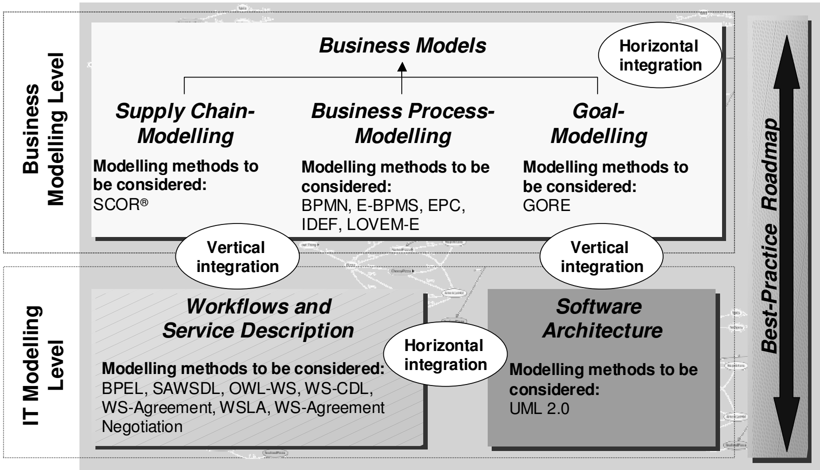

Fig. 1. The BREIN Modelling Framework [2]

The Business Modelling Level of the BREIN Modelling Framework has been instantiated with:

- Supply chains to represent the inter-organisational view,
- Business processes to model the intra-organisational perspective and
- Goal models for steering the business collaborations.

For the modelling of supply chains the SCOR method [4] is considered as it is one of the very few dedicated methods and has also a high relevance for many industries. Regarding business process modelling there are various modelling languages available such as BPMN [5], E-BPMS [6], EPC [7], IDEF [8] or LOVEM-E [9]. Also for goal modelling there are several approaches available, predominantly related to GORE (Goal-oriented requirements engineering). GORE has yielded a variety of modelling approaches including AGORA (Attributed Goal-Oriented Requirements Analysis) [10], GBRAM (Goal-Based Requirements Analysis Methodology) [11] and KAOS [12].
On the IT Modelling Level we distinguish between two areas:

- Workflow and Service Description for the discovery and the execution of services and workflows (here standards like BPEL [13], OWL-WS [14],WS-CDL [15], etc. will be considered) and
- the Software Architecture to document the platform and for the technical specification of services and components to be developed (here the de facto standard UML [16] will be used).

The integration of all these modelling languages comes with several challenges that may be classified according to their level:

1. Syntactic level: Deals with the representation of the models.
2. Semantic level: Concerns the question how different methods may be integrated.

3. Contextual level: Refers to the ambiguity if terms are used in different contexts.

The remainder of this chapter discusses the challenges in more detail and chapter 4 describes how they have been tackled in the BREIN Modelling Framework.

2.1 Syntactic Level

The syntactic level concerns models and their representation. Modelling languages differ on a syntactical level which means that they are stored in different formats and there is no unified way to access them.

Models may be stored as files or in repositories (such as relational or XML databases), and their format may not follow any standard representation. Proprietary formats and different data sources lead to many problems as they require the development of new interfaces for every representation.

An example are OWL ontologies that are stored in an XML database, business processes in a relational database, UML diagrams stored as files and an organisational model in a spreadsheet format. Concerning the database there are again different vendors, some of them implementing the SQL standard but also others that don't. Working with spreadsheets produces even bigger issues for retrieving the information stored within as there is no accepted (de facto) standard for accessing them.

2.2 Semantic Level

The semantic level concerns the modelling language and raises the question how different modelling languages may be integrated with each other. The integration of two modelling languages usually results in a gap, and the key question is how this gap may be bridged.

In particular there are substantial difficulties when it comes to bridge the gap between the business and the IT-view referred to as the Business-IT gap. One approach is to use business processes to derive requirements for IT services that will in return (in part) automate them. In this case the method that is used to model the business view has to be integrated with the one representing the technical world (e.g. a business process modelling language for the business view and UML [16] for the modelling of software artefacts).

Process-oriented requirements analysis [6] follows top down approach, starting with identifying processes in form of a process map. The next step refines the processes in a business process model decomposing them in a sequence of activities and decisions. This description may be further enhanced by the definition of the working environment, the involved resources etc. Finally, business use cases may be derived from the activities.

A use case in general is a specific way of utilizing a system using some part of functionality [16]. A business use case refers to business actors and their goals for interacting with the business.

In the UML the concept of use case exists, but rather as system use cases which represents the scope of an application. Therefore a single business use case may be mapped to several system use cases, each defining the interaction with a single application. Nevertheless the gap between business and technical view may be bridged via the use case concept.

2.3 Contextual Level

The contextual level refers to the fact that different stakeholder use the same term in different contexts. For instance. a business expert will interpret the term service in the business context referring to the non-material equivalent of a good. A technician might understand service as software that is deployed as a web service. Mentioning another example, the term business process might refer to concrete business process in a company e.g. transporting the passengers from the gate to the aircraft at the airport, or could be interpreted as a sequence of web service calls by a technician.

The problem is that business experts cannot define software, as they don't have the appropriate background. On the other hand for software developers it is difficult to understand how business works and to derive the requirements. Nevertheless a common understanding of the domain between all stakeholders is essential for a successful system implementation.

BREIN will try to implement a "business grid" analysing two concrete scenarios and deriving requirements from them. The above mentioned ambiguity is a mayor problem as the communication between business people and technician is crucial when implementing such a system.

3 The Concept of the BREIN Modelling Framework

This chapter explains modelling approaches relevant as a basis for the Modelling Framework.

The first step in defining the BREIN Modelling Framework was the identification of mayor technologies involved. As a basis the modelling approaches of ATHENA [17], EMI [18] (Enterprise Model Integration), MDD/MDA [19] (Model Driven Design/Model Driven Architecture) and the concepts of SOA [20] (Service Oriented Architectures) were analysed.

ATHENA has developed a viewpoint-based integration approach to model interoperability which shows the existence of multiple viewpoints comprising the viewpoints of business analysts, product developers, system architects, and software developers. It becomes evident that these viewpoints share some common objects or concepts that need to be integrated in order to sustain modelling integrity.

A quite similar attempt of defining a generic modelling framework is the EMI approach. Enterprise Model Integration stands for the integration of the different modelling methods used on the design graph. On the one hand, the design graph includes semi-formal business oriented models, which try to explain the business domain independent of the technology to be used. On the other hand, the design graph includes platform independent technical models which, in the case of processes, can be considered as the basis for process execution.

With respect to ATHENA the BREIN modelling framework adopts ideas like the view point orientated integration approach. Nevertheless we have a slightly different focus with respect to the modelling methods used. The concepts of EMI are necessary for the integration of meta-models of modelling languages describing different aspects of a company.

Considering the observations made in the analysis, the following separation of layers with their respective models, which are also defined in the MDD/MDA approach [19] by the OMG [21], have been derived:

1. CIM (Computation Independent Model): Models capturing the real world business, serving as the requirements.
2. PIM (Platform Independent Model): Workflow models on a platform-independent layer that may be derived from the upper layer.
3. PSM (Platform Specific Model): Executable models bound to a specific platform, which are a refinement of the PIM layer.

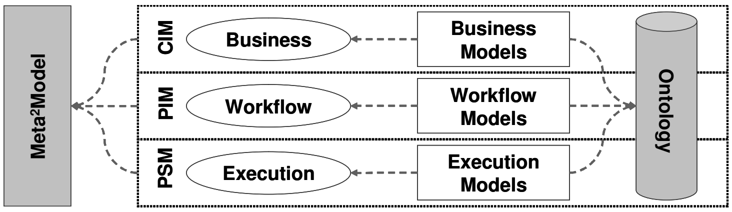

Fig. 2. Conceptual BREIN Modelling Framework [2]

The resulting conceptual BREIN Modelling Framework is depicted in Fig. 2.

4 Addressing the BREIN Modelling Challenges

In the following, we explain how in chapter 2 identified challenges will be tackled by the BREIN Modelling Framework.

4.1 Syntactic Level

These shortcomings are addressed by providing one generic model repository for the models of used by different methods. The meta-model approach allows (1) the use of several modelling languages and (2) makes the used methods exchangeable.

The generic model repository acts as a mediator that allows access to the models through one unified XML format. The functionality of the modelling framework is also exposed through web service interfaces. The framework allows the model interchange with external systems through the implementation of import and export mechanisms with standard formats including EPC [7], UML/XMI [16], BPEL [13] and OWL [22].

In the following, the details on how this framework has been implemented to realize this vision will be introduced.

A web based modelling environment was implemented to realise the BREIN Modelling Framework. The environment allows distributed modelling as all services are offered over the web.

The main aspect of the Modelling Framework is the focus on individual services that are linked together for providing required functionalities. This view is influenced by the concept of Service-Oriented Architecture [20] as it is today discussed in practice and industry. To implement such services the decoupling of certain parts of functionalities is necessary. A distinction is made between Utility Service, Basis Service and Component Service.

Basis Services are services that provide a benefit for all Component Services if used. This means that basis features like caching, logging, security can be used by the Web-Applications. The Basis Services are selected based on the OGSA framework [23] and compared with the J2EE and the .NET Basis services. This means that using the framework automatically enables the usage of such Basis Services.

Utility Services are individualised sets of features that run in a specific project configuration and follow the framework to be used by any service. Such Utility Services usually provide benefit to a group of services – e.g. ontology communication services that takes model information to compare with the ontology – so it is not classified as a Basis Service, as it can not be found in the above mentioned framework.

The term "Component Services" describes the third category. The Component Services are autonomous sets of functionalities that run on top of the Basis Services.

Examples for a Component Service would be the Import/Export Service, the Modelling Service, etc. The functionalities of Component Services are also offered via web service interface. Component Services are decoupled from the user interface. "Web-GUIs" for services reside on the highest layer and offer the functionality of the Component Service via a user interface.

4.2 Semantic Level

The BREIN Modelling Framework enables the integration of modelling languages with meta-model integration patterns [18]. The modelling languages are linked using a loose integration pattern. The modelling languages are coupled using a so called "transition layer" that contains concepts from both methods.

There are different types of integration, here we consider vertical and horizontal integration:

Vertical integration is a typical top-down or bottom-up approach where different levels of abstraction are integrated. For the top-down-integration the starting point are the elements of the higher level method. Method fragments of the lower layer are selected and integrated based on the requirements from the upper method. Another possibility is the bottom-up integration which is more common in reengineering attempts.

To conflate the Business Modelling Layer and the IT Modelling Layer vertical integration is needed and, in the case of BREIN that would be a top down integration approach. This means that business goals, strategies and business processes serve as starting points for application development. The meta-models from the business modelling level are integrated and refined with the meta-models from the IT modelling level.

Horizontal integration is used for the integration of method fragments at the same layer of abstraction, which means that we integrate meta-models with the same level of detail. This integration approach is used to integrate the methods in the Business Modelling Layer and the methods in the IT Modelling Layer, respectively.

4.3 Contextual Level

This challenge will be tackled through an integration of meta-models and ontologies. We propose an bootstrapping approach [24] where models serve as basis for the creation of ontologies and vice versa. The idea is to define a uniform terminology, which will be available in form of an ontology. This terminology will be used by modelling services to assist the user when creating models.

The BREIN modelling framework will facilitate the use of this terminology through two components:

- Ontology Generator: supports the generation of an ontology from business process models
- Modelchecker: Checking the consistency of the models according to the terminology

The following subsections deal with these two components of the Modelling Framework.

Ontology Generator. From the literature of meta-models and their relation to ontologies [25], [26] it is reasonable to use business process models, filtering them according to relevant concepts and transforming the relevant ones into an ontology.
As the ontology language (e.g. OWL [22]) and the business process modelling language may be interpreted as meta-models, the meta-modelling reference pattern according to [18] can be applied. For the model integration two dimensions have to be considered, first the direction of integration (vertical, horizontal or hybrid) and second the level of integration (loose, intermediate, and strong).
BREIN implements a vertical integration between business process models and ontologies. The integration can be also considered as loose as we consider the business process models and the ontology almost independent.

Basically there are two ontologies that can be applied:

- An ontology on a meta-model layer, in the case of BREIN, a list of concepts for business process modelling
- An ontology on a model layer, in the case of BREIN, a list of concepts of the scenarios.

The first approach deals with the transformation of the meta-model into an ontology. In BREIN this concerns the business process modelling meta model which is a specially adapted meta model following the E-BPMS [6] paradigm.

The meta-model concepts have been analysed according their usability and transformed into ontology concepts. The concepts have been re-named and aligned to fit into the BREIN ontology. The result is a top-down analysis of concepts that were used for the description of the BREIN scenarios.

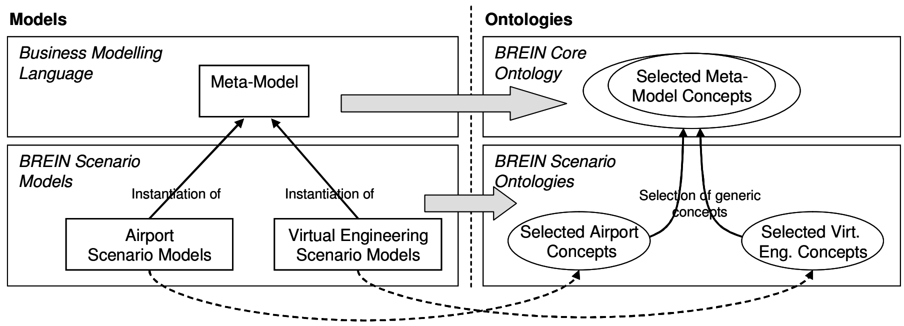

Fig. 3. Generation of ontologies from business models

The second approach deals with the transformation of concrete business process models into an ontology. Here the semi-formal description (models) of the two scenarios (Airport scenario and Virtual Engineering scenario) were the basis to derive the scenario specific concepts (see Fig. 3).

In parallel to this top down approach, a bottom-up approach by knowledge engineers captured a number of concepts which were integrated into the ontology.

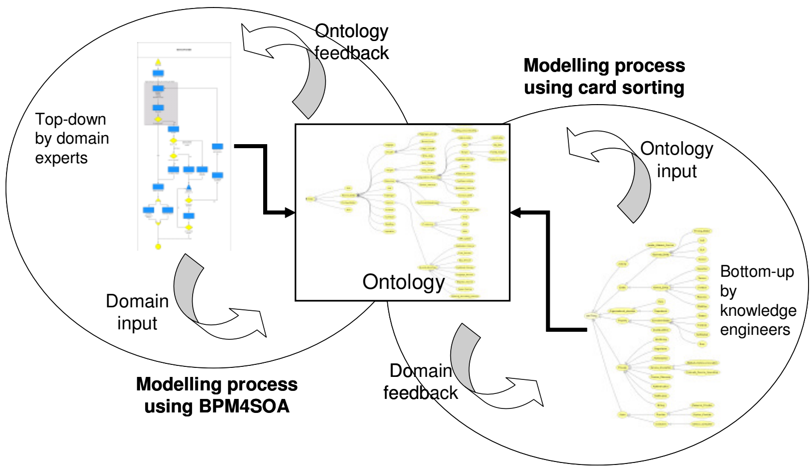

Fig. 4. Ontology generation approach

The combination of these approaches led to the definition of the ontologies in an iterative way (see Fig. 4).

Modelchecker. Once an terminology is defined and represented as ontology it may be used to support the user in creating consistent models. The idea is that the ontology does not replace models but it is rather considered complementary, as it defines the domain conceptually, contains the correct terminology and may serve as reference during modelling.

The Modelchecker also allows, apart from the basic mechanisms of syntax checking that Modelling tools usually offer, the semantic checks.

Through the use of the domain ontology following shortcomings that concern modelling can be addressed [27]:

- If the models are created by several users the used vocabulary differs. The Modelchecker compares the used terms with predefined ones and gives feedback if:

1. the used term already exists (additionally providing information on its definition or context),
2. the existing term has associated synonyms or
3. the term does not exist.

If the term does not exist it can be added to ontology so that is available for further use (missing concept) or the user may decide to use another term that already exists within the ontology. This approach couples ontologies and the modelling and feeds the ontology with new concepts during modelling through a bootstrapping process [24].

- Modelling is usually done at different levels of abstraction. Within one model the modeller is supposed to keep the same level of abstraction. The ontology can be used to check if there is a problem of granularity in the modelled process.
- The ontology can be used to assist the user via auto-completion functionalities when entering information.
- The information incorporated in the ontology can be used to perform a functionally enriched search. It is not only possible to search for exact text matches but the ontology may be used to find related concepts and to search for things that are close to the term specified (e.g. car is similar to vehicle, motor-bike is similar to car or truck, etc.).

There are some additional usage scenarios that are considered to be implemented in the future.

5 Conclusion

In this paper we first identified basic and BREIN related requirements for a Modelling Framework that should support all modelling related tasks within the project. Based on the requirements we developed the concept for the BREIN Modelling Framework and explained the related modelling challenges on three levels.

The BREIN Modelling Framework addresses these challenges. The aspects of ontology generation and the usage of ontologies by the modelling framework were discussed in more detail. The framework has already been implemented and is currently in use by the project consortium. As the project progresses the BREIN Modelling Framework will further evolve. This includes the implementation of the Best-Practise Roadmap, the development of more components, a exact definition of the Workflow and Service Description method and the creation of reference models for all related modelling languages.

Acknowledgments. This work has been supported by the BREIN project (http://www.gridsforbusiness.eu) and has been partly funded by the European Commission's IST activity of the 6th Framework Programme under contract number 034556. This paper expresses the opinions of the authors and not necessarily those of the European Commission. The European Commission is not liable for any use that may be made of the information contained in this paper.

References

1. BREIN project, http://www.gridsforbusiness.eu
2. Leukel, J. (Ed.): BREIN Deliverable 3.3.3, The BREIN Modelling Framework, http://www.gridsforbusiness.org, (2008)
3. Woitsch, R., Utz, W.: Roadmap to Akogrimo Convergence: A Sample of Process Oriented Knowledge Management with PROMOTE®, BPOKI'06 (Business Process Oriented Knowledge Infrastructures), at I-KNOW '06, Graz, Austria (2006)
4. Supply-Chain Council (Eds.): Supply Chain Operations Reference Model (SCOR®) Version 8.0, http://www.supply-chain.org, (2006)
5. OMG: BPMN 1.0, http://www.bpmn.org/
6. Bayer, F.; Junginger, S.; Kühn, H.: A Business Process-oriented Methodology for Developing E-Business Applications. In: Proceedings of the 7th European Concurrent Engineering Converence (ECEC 2000), Society for Computer Simulation (SCS) (2000)
7. van der Aalst, W.: Formalization and Verification of Event-driven Process Chains, Information & Software Technology (1999)
8. IDEF Integrated Definition Methods, http://www.idef.com/
9. IBM: Line of Visibility Enterprise Modeling (LOVEM), http://www-935.ibm.com/services/ us/index.wss/offering/gbs/a1006593
10. Kaiyia, H., Horai, H., Saeki, M.: AGORA: Attributed Goal-Oriented Requirements Analysis Method. In: Proceedings of 10th Anniversary IEEE Joint International Conference on Requirements Engineering. IEEE Computer Society (2002)
11. Anton, A.I., Carter, R.A., Dagnino, A., Dempster, J.H., Siege, D.F.: Deriving Goals from a Use Case Based Requirements Specification. In: Requirements Engineering Journal 6 (2001)
12. van Lamsweerde, A.: Requirements Engineering in the Year 00: A Research Perspective. In: Proceedings of the 22nd International Conference on Software Engineering (ICSE´00), ACM Press, Limerick (2000)
13. IBM: Business Process Execution Language for Web Services, http://www-128.ibm.com/developerworks/library/specification/ws-bpel/
14. Beco, S., Cantalupo, B., Giammarino, L., Matskanis, N., Surridge, M.: OWL-WS: a workflow ontology for dynamic grid service composition, e-Science and Grid Computing (2005)
15. W3C: Web Services Choreography Description Language, http://www.w3.org/TR/2004/WD-ws-cdl-10-20040427/

16. Object Management Group (OMG): Unified Modeling Language Superstructure Version 2.0, http://www.omg.org/docs/formal/05-07-04.pdf, (2005)
17. ATHENA Project, http://www.athena-ip.org/
18. Kühn, H., Bayer, F., Junginger, S., Karagiannis, D.: Enterprise Model Integration, Proceedings of the 4th International Conference ECWeb (2003)
19. Object Management Group: MDA Guide Version 1.0.1, http://www.omg.org/docs/omg/03-06-01.pdf
20. OASIS: OASIS SOA Reference Model, http://www.oasis-open.org/
21. OMG (Object Management Group), http://www.omg.org/
22. W3C: OWL Web Ontology Language Use Cases and Requirements, W3C Recommendation 10 February 2004, http://www.w3.org/TR/webont-req/
23. Open Grid Forum: The Open Grid Services Architecture, http://www.ggf.org/documents/GFD.80.pdf
24. Karkaletsis, V., Paliouras, G., Spyropoulos, C.: A Bootstrapping Approach to Knowledge Acquisition from Multimedia Content with Ontology Evolution, Adaptive Knowledge Representation & Reasoning (AKRR'05), Helsinki, Finland (2005)
25. Höfferer, P.: Achieving Business Process Model Interoperability using Metamodels and Ontologies. In Österle, Hubert; Schelp, Joachim & Winter, Robert (Eds.): Proceedings of the 15th European Conference on Information Systems (ECIS 2007), St. Gallen, Switzerland, University of St. Gallen (2007)
26. Woitsch R., Karagiannis D., Fill H.-G., Blazevic V.: Semantic Based Knowledge Flow Systems in European Home Textile: A Process Oriented Approach with PROMOTE. Proceedings of the I-Know (2007)
27. Leutgeb, A., Eichner, H., Utz, W., Woitsch, R., Fill, H-G.: Semantic and Adaptive E-Government Processes - A Field Report About Ontology Based Approaches in EU-Projects, In Proceedings of the Eastern European e|Gov Days 2007, Prague, Czech Republic (2007)