An Enterprise Architecture Framework to organize Model Repositories

Knut Hinkelmann, Simon Nikles, Holger Wache, Daniela Wolff

FHNW University of Applied Sciences Northwestern Switzerland
Riggenbachstrasse 16, 4600 Olten, Switzerland
{knut.hinkelmann, simon.nikles, holger.wache, daniela.wolff}@fhnw.ch

Abstract. Models are a valuable knowledge asset for an enterprise. An enterprise model repository can improve sharing of enterprise knowledge and thus can exploit the use of the knowledge for various applications. In this work we present a framework for the organisation of enterprise models. The framework was derived from enterprise architecture frameworks. It distinguishes three dimensions: aspect, perspective, and modelling language family. For each of these dimensions we derive possible values. The framework can be used for enterprise repositories but also for knowledge exchange in a community as proposed by the Open Model Initiative.

Introduction

Modelling is defined as describing and representing all relevant aspects of a real-world/domain in a defined language. Modelling results in a model, which is a reproduction of the part of (possible) reality which contains the essential aspects to be investigated from a specific view point. For instance, an architect designs a building by creating a plan (model) which represents a real building.

By abstracting from irrelevant details they are a means to exchange information, to make simulations, support decision making. Thus, repositories of models are a valuable knowledge asset.

In companies we find various kinds of information models. For example, in Business Process Management (BPM) projects a large number of models are created representing knowledge about business processes or working environments. Model development is also a main task in software engineering, with data models, use case models or UML class diagrams being typical examples. This enumeration could be extended further. These models can differ in representation language and formality. Many models are graphical but we also find formal mathematical models or textual descriptions which can be regarded as informal models.

Models are often distributed over several media and repositories. This can have technical reasons. For example, models of BPM projects are often stored in BPM tools having their own model repository. There are also organizational reasons, as models are developed in a project and stored together with other project or product documents. This is often the case for software development.

In Hinkelmann et al. (2010) we propose an enterprise repository consisting of a linked models together with relations to real data and object. The Open Models Initiative (OMI) is an initiative that is not restricted to an enterprise but "intends to establish a community of people who focus on the creation, maintenance, modification, distribution, and analysis of models." (Karagiannis et al. 2008).

To exploit the knowledge represented in models it is important to organize the repository - being it restricted to a project, an enterprise or open like the OMI - in order to allow for effective and efficient storage and retrieval of models. Metadata are a common approach for describing resources in a repository that can be enhanced by knowledge organization methodologies like thesauri, taxonomies or even ontologies.

In the following we present a framework for organizing enterprise model repositories that is based on enterprise architecture frameworks. The framework builds on work in the plugIT project, where we used a similar approach to compare modelling languages.

It is our idea to use our proposed framework in order to support the Open Models Initiative (OMI). In the basics the OMI needs to store its open models in a repository – the OMI repository. The annotation of the models with terms from this framework simplifies the retrieval of the stored models and helps to compare models even if they are modelled in different modelling languages.

To motivate our approach we come back to the analogy to architects as it was already used by Zachman in his seminal paper (Zachmann 1988): There may be several different plans all about the building but focussing a specific viewpoint: A plan which contains all the plugs and wires in a building is used by an electrician for the electric installation, or a plan that shows the water taps. From this, Zachman derived a two-dimensional distinguishing perspectives and aspects to organize the models of a enterprise architecture.

In addition to these two dimensions we use a further one that refers to the compatibility of the modelling languages. Thus we have at least three dimensions (see **Fig. 1**):

The framework, however, can also be used as knowledge structure. The framework classifies models, modelling languages and tools according to several dimensions – depicted in

Perspectives - this dimension helps to clarify the role of the user and the application fields of the model

Aspects - this dimension is about what should be modelled and thus deals with the application fields and the modelling concepts

Language families - this dimension groups models according to their modelling languages which are based on a common philosophy.

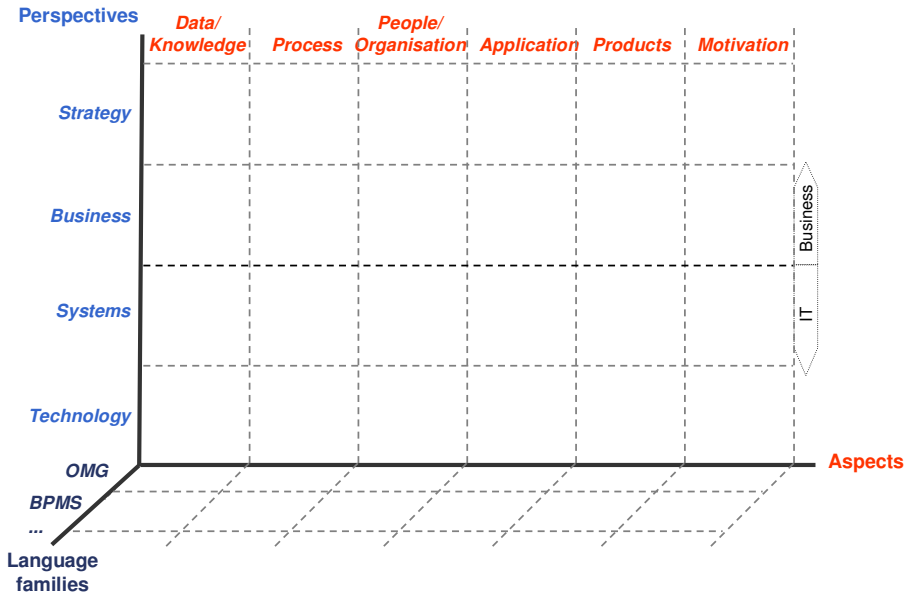

Fig. 1 The dimensions of the classification framework

In the following, we first motivate, why we selected these dimensions. Then we explain each of these dimensions before we synthesize a proposal for a framework.

Dimensions of the Modelling Organisation Framework

The model organization framework must allow for organizing knowledge models of any business-related topic. These topics correspond to the aspects of a enterprise architecture. Different stakeholders or roles have different viewpoints on each of these aspects. For the owner of an enterprise only those elements are of interests that correspond to the strategy and the business model while a programmer focuses on implementation details. This results in a two-dimensional structure where each aspect can be viewed from different perspectives.

There are various frameworks for describing the elements of an enterprise architecture like the Zachman Framework (Zachman 1987), the ARIS Architecture for Information Systems (Scheer 1999), the BPMS Framework (Karagiannis 1995, 1996) or The Open Group Architecture Framework TOGAF (Open Group 2009).

In the following we look at some of the frameworks in more details. **Fig. 2.** shows the two dimensions of the Zachman framework. The columns correspond to the different aspects and the rows correspond to the perspectives. The cells of the structure show examples of models and the main concepts to be represented in these models.

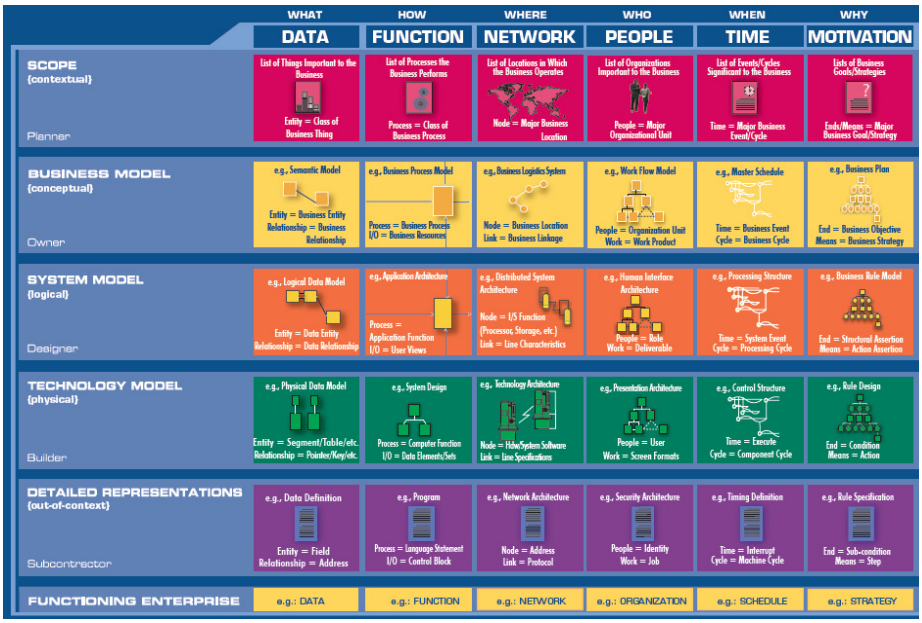

Fig. 2. Perspectives and aspects of the Zachman framework (Zachman 2009)

These two dimensions can also be found in the business process management frameworks. The ARIS Architecture for Integrated Information Systems (Scheer 1999) distinguishes between views and descriptive levels, which correspond to aspects and perspectives, respectively. In **Fig. 3.** the aspects (views) are represented as boxes and the perspectives (descriptive levels) as levels in these boxes. PROMET (Österle 1995) has three perspectives and several aspects.

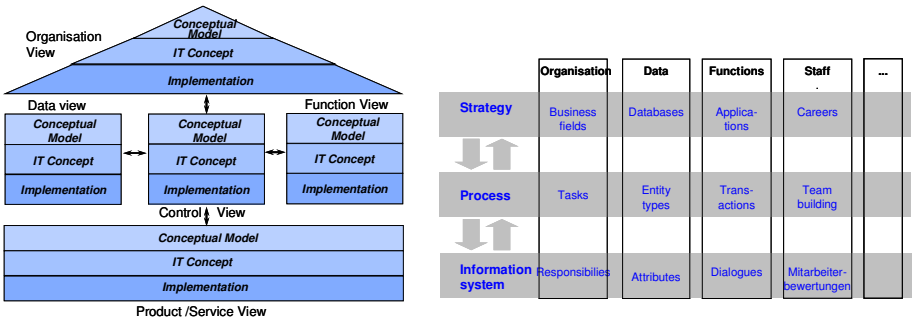

Fig. 3. Perspectives and aspects of the BPM Frameworks ARIS (Scheer 1999) and PROMET (Österle 1995)

For the BPMS approach (**Fig. 4.**) the perspectives correspond to the sub processes of business process management while the aspects depend on the application scenario. For the business process management of ADONIS® the aspects correspond to the core elements.

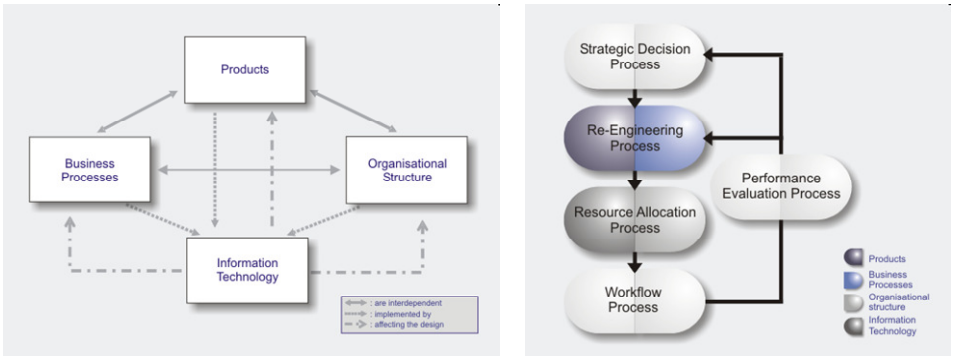

Fig. 4. Aspects and perspectives of BPMS

The Perspective Dimension

The perspective dimension helps to clarify the role of the stakeholders or roles for which the model is intended. From the different perspectives, each representation is different in nature, in content and in semantics from the others. The various frameworks differ in the number and roles of the different perspectives. In the following we will first give a short overview of the perspectives found in the various frameworks and then present the perspectives we chose here.

- Derived from the analogy to classical architecture, the Zachman framework (Zachman 1987) distinguishes five perspectives:
  - Scope: the perspective of the planner
  - Business Model: the perspective of the business owner
  - System Model: The perspective of the designer
  - Technology Model: The perspective of the builder
  - Detailed Representation: The perspective of the subcontractors
- The Open Group Architecture Framework TOGAF (Open Group 2009) has also a distinction that corresponds to the perspectives; it distinguishes
  - Business Architecture
  - Data/information Architecture
  - Application Architecture
  - Technology (IT) Architecture

- The description levels of the ARIS framework represent phase of development and correspond to these perspectives:
    - Conceptual model: Here the business view of modelling is realized.
    - IT concept: In this phase the design tasks for the respective views are performed (e.g. module design or relational schema)
    - Implementation: realization with IT (e.g. program flow, physical network implementation, …)
- In the BPMS approach (Karagiannis 1995) the perspectives correspond to the sub processes of business process management:
    - Strategic Decision: The perspective of the business owner
    - Re-Engineering: the perspective of the process owner and designer
    - Resource allocation: the perspective of process implementation taking into account IT related issues
    - Workflow: the perspective of the process execution
    - Performance evaluation: the perspective of the controller
- The Model-driven architecture of the OMG distinguishes between three levels which also correspond to perspectives. On each level OMG defines specifications and modelling languages describing different aspects of the enterprise with increasing technological details (OMG 2003):
    - CIM Computation-Independent Model
    - PIM Platform-Independent Model
    - PSM Platform-Specific Model
- The PROMET approach also distinguishes three perspectives. Compared to the OMG and ARIS framework, however, the focus is more on the business level: Strategy, Business Processes, Information Systems

Our framework is intended to support all these perspectives. However, as these frameworks use different terminology we need to harmonize it. After introducing our perspectives now we will show in Table 1 how the perspectives of different frameworks can be mapped onto our perspectives.

As enterprise architectures and models can be used for the alignment of business and IT, the distinction of the business and IT perspectives is important. We decided to further divide the business perspective into strategy and business, because the strategy of a company determines the business and both business and IT have to be compliant with the strategy. On the IT level we distinguish between system and technology models, where the last one contains more technology and implementation details. Thus, in our classification scheme we distinguish four different perspectives, two for business and two for IT:

Strategy - the perspective of the business owners. They outline the major objective of the company and how to achieve them in general.

Business - the perspective of the business professional. Strategic objectives have to be deployed in the daily business. Business engineers for example model at this layer the processes and describe how a product of the company has to be produced.

Systems - the perspective of the systems engineer. On this level, software components, servers, workflow models etc are represented. It is independent of a specific platform, programming language, operating system etc.

Technology - the perspective of the IT professional; it roughly corresponds to the platform specific model of OMG.

The following table relates the perspectives of the different frameworks to the four perspectives we will use in this paper.

plugIT	*Zachman*	*BPMS*	*ARIS*	*TOGAF*	*OMG*	*PROMET*
Strategy	Scope	Strategy	Conceptual Model			Strategy
Business	Business Model	Re-Engineering		Business Architecture	CIM	Processes
System	System Model	Resource Allocation	IT Concept	Information/ Application Architecture	PIM	Information Systems
Technology	Technology Model	Workflow	Implemen-tation	Technology Architecture	PSM	
	Detailed Model	Performance Evaluation				

Table 1. Relating the perspective dimension of several frameworks

It should be kept in mind that the representations from the different perspectives do not correspond to different levels of details – level of detail is an independent variable, varying within one representation. As a consequence, for example, the perspective "detailed model" of Zachman is not seen as an own perspective in our model but is merged with the technology perspective. Further the performance evaluation of BPMS takes into account performance figures from business and IT and thus is not considered a perspective of its own.

The Aspects Dimension

Aspects describe the application fields of the modelling languages. Similar to the perspectives, there is no generally agreed upon set of aspects in the various frameworks for enterprise architectures. For our framework we use the following aspects:

Data/Knowledge - this aspect describes the data, information and knowledge being used. Zachman only called this aspect data, but since there is no explicit knowledge aspect and while knowledge builds on data and information we added knowledge to this dimension.

Process - processes coordinate the tasks of a company and explain at different abstraction levels how and in which order tasks have to be performed. It corresponds to Zachman's Function perspective.

People/Organisation - people act in an organisational environment which is described by this aspect.

Applications - the models for this aspect describe IT systems, applications and their connections in a network environment. The aspect includes Zachman's Network.

Products - this aspect describes the features of products and services of an enterprise. This is important because products and services of an enterprise determine the processes and business model and thus should be represented explicitly. The product aspect is not available in Zachman's framework but can be found in business process management frameworks like ARIS and BPMS.

Motivation - enterprises do not - or should not - act randomly. When an enterprise executes a business process or applies a business rule, it should be able to say why. This is modelled in the motivation perspective.

The aspects of the other frameworks can be mapped onto our aspects. However, compared to Zachman there are two main differences. First we do not use time as an aspect of its own. Although time is an important metadata, models of time are part of an enterprise repository. We use time as a metadata as it is used also for Dublin Core or in combination with processes (order of activities, cycle time, working time). Instead we added the product aspect that can also be found in ARIS and BPMS. It is important to represent products, because their production, delivery, storage etc. is the main task of enterprises and thus should be represented explicitly.

Worth mentioning is also the renaming of the data dimension to data/knowledge. This is due to the fact that knowledge is not explicitly mentioned in any of the frameworks. Although it is often argued that knowledge is related to humans, the organizational structure (where people are modeled) does not cover any topics or skills of the people. As already shown by Newell (1982) knowledge should be separated from its representation and consequently should not be unified with the source (in this case the human). Newell therefore introduced the knowledge level. As for modelling knowledge similar models are used as for data modelling and to avoid the distinction of information, data and knowledge we combined these aspects into one dimension.

Modelling Languages and the Language Families Dimension

Standardisation bodies, communities and tool vendors offer modelling languages based on a common "philosophy" or methodology. Most often, these languages support references or mapping to models representing several aspects or perspectives. Important providers and developers of modelling languages are standardisation bodies:

- Object Management Group (OMG; http://www.omg.org/) - the OMG is the most important standardisation organisation. It develops, for example, the Unified modelling language UML which is a language family mainly dealing with various aspects for the IT level (OMG 2009b). A second family of languages deals more with the business level. This family includes for example the Business Process Modelling Notation BPMN (OMG 2008) and Semantics of Business Vocabulary and Business Rules (SBVR) which defines a terminology for documenting the business vocabularies, business facts and business rules (OMG 2009).
- W3C (http://www.w3.org) - the World Wide Web consortium defined standards both for modelling data and knowledge (e.g. XML, RDF, OWL) as well semantic representation of processes, services and rules with the objective to access and distribute them via internet.
- OASIS provides standards for modelling processes and Web Services like the Business Process Execution Language BPEL and the Web Service Description Languages WSDL (OASIS 2007).

A second source of language developers are vendors of modelling tools:

- Vendors of business process management tools like IDS Scheer with the ARIS tool (http://www.ids-scheer.com/en/ARIS/) or BOC with the Management Office (http://www.boc-eu.com/) offer modelling languages for several aspects and perspectives.
- Vendors like ILOG or Fair Isaac concentrate on rule modelling and offer several representation formalisms with their Business Rules Management systems.

This list covers standardisation bodies as well as relevant tool providers but is by far not complete.

The term modelling language is not as clearly defined as it can be regarded as model type or are a group of model. For instance, the OMG calls the Unified Modelling Language UML a language, but it consists of several diagrams. On the other hand, the Web Ontology Language OWL and the Semantic Web Rule Language are both called language although they are closely related as OWL concepts are constituents of SWRL rules. To clarify the terminology we distinguish between modelling languages and language families.

- A language ***family*** is a set of modelling languages that based on a common "philosophy" or methodology. Often there are relations, references and dependencies between languages of the same family. In this sense, UML is a language family. Roughly speaking, the set of languages offered by a standardisation body or vendor that is based on a common methodology or that have a common meta-meta-model are language families.
- A modelling ***language*** consists of modelling elements, relations and attributes that can be used together. It thus corresponds to a model type or diagram type. For example, languages OWL, EPC, BPEL or WSDL are modelling languages in this sense. The different diagrams of UML like Class diagrams, object diagrams or activity diagrams are also modelling languages in this sense.

We use the language families as a dimension in our framework. This allows us to examine together all those modelling languages that have the same paradigm. This is important for practical reasons: In a project one often decides for a specific modelling paradigm or a specific tool to model with. Then it is important to be able to retrieve and reuse exactly those models that are in line with this decision. The modelling language can also be added as metadata for models.

Relations between the dimensions

The different dimensions are orthogonal. **Fig. 5.** illustrates the topics of different models. Although all every combination is labelled, there does not necessarily exist a model. For instance, there is no model for product at Information system level. Also, there is not a direct corresponding model for relating the IT aspect at the strategic perspective layer.

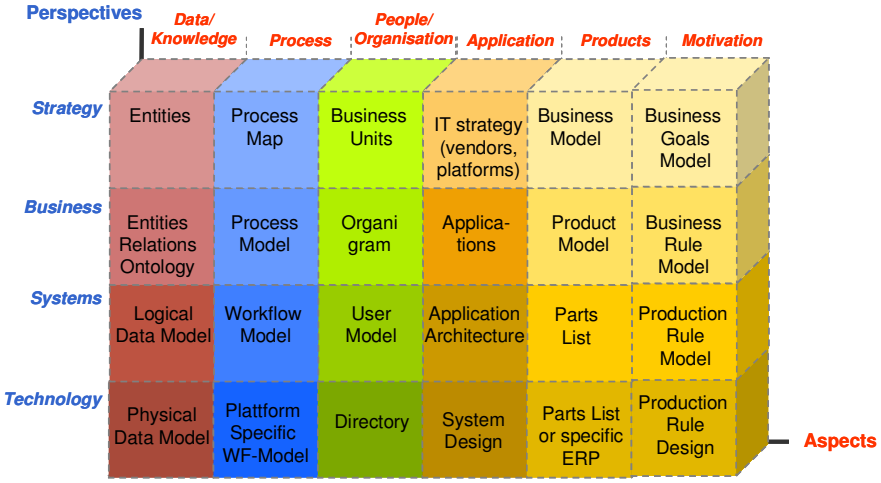

Fig. 5. Models associated to perspectives and aspects

How to use the Framework

To store and retrieve resources in a repository – for example models in a repository of the Open Model Initiative (OMI) – it is common to use metadata for describing the resources. Dublin Core (DCMI 2008) is a well-known metadata standard consisting of 15 elements in standard Dublin Core which are extended and refined for qualified Dublin core. Examples of Dublin Core elements are creator, subject, title, type or data (which can be refined to created, modified, valid, issued etc).

The dimensions of our framework can be regarded either as additional metadata elements specific for enterprise models or as encoding schemes for subject (perspective and aspect) or type (language family). Please note that it is allowed to use the other frameworks like the Zachman framework by mapping their perspective and aspects to our framework.

When a model is stored in the repository (like the OMI repository), a value is given for each of these elements. Assume we store a model for the sales process which is represented in BPMN, we would store at least the following metadata:

creator:	Knut Hinkelmann
aspect:	Process
perspective:	Business
language family:	OMG
modelling language:	BPMN
created:	June 24, 2010

A class diagram for a CRM system could have the following metadata:

creator:	Daniela Wolff
aspect:	Data/Knowledge
perspective:	System
language family:	UML
modelling language:	Class Diagram
created:	June 28, 2010

In addition to representing the three dimensions of the framework as metadata elements we could also use them as a basis for a graphical interface to browse a model repository: By clicking for example on a specific aspect, all models representing these aspects are listed. By clicking on a specific cube all models for the intersection are shown. For example, if a user clicks on the cube for "Process - Business" all process models from the business perspective are listed.

A user interface following this principle has already been implemented in the modelling language wiki system of the plugIT project[1]. This wiki contains a state of the art for modelling languages. The graphical interface is used to browse the modelling languages by selecting the perspectives or aspects the language should be suitable for.

[1] http://plug-it.org/plugITwiki/index.php5/Main_Page

As an entry point the framework is visualized as a clickable image map corresponding to **Fig. 5** where each perspective, aspect and cube is clickable what leads to a list of all corresponding models as shown in **Fig. 6.**

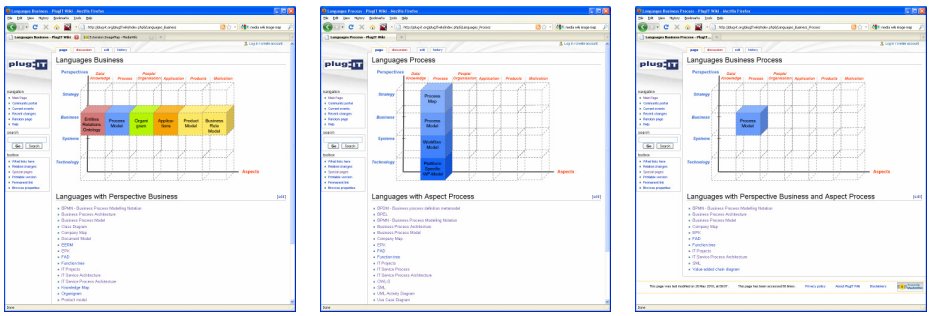

Fig. 6. Screenshot of the classification framework

The functionality of a Semantic Media Wiki (SMW) is used, to support stronger relationships between pieces of information (Barrett 2009). The models can be annotated with aspects and perspectives and retrieved with intricate queries, such as "Get all models of the aspect process with the perspective business".

In future the SMW may be extended by not only classifying the models but also providing further information to models, e.g. links to modelling tools allowing viewing or editing this kind of model, links to the specification of the modelling language, links to best practices how to use this model, or access to the models stored in the OKM repository. Even the last point offers a very nice entry point into the repository of the OMI.

Conclusion

In order to exploit the knowledge asset contained in enterprise models, storage and retrieval of the models should follow general, easy to understand and agreed-upon principles.

In particular graphical models are a general means to represent knowledge with the intention to make it understand. An enterprise model repository can improve sharing of enterprise knowledge in a company. It also exploits the use of the knowledge for various applications. For example, in the plugIT project, an IT-Socket is developed that uses enterprise models in order to support the alignment of business and IT (Woitsch et al. 2009). The Next Generation Modelling Framework, also developed in plugIT, allows for collaborative modelling integrating both different model repositories and different modelling tools.

In this work we presented a framework for organisation enterprise models. The framework was derived from enterprise architecture frameworks. It can be used for enterprise repositories but also for knowledge exchange in a community as proposed by the Open Model Initiative.

Acknowledgement

This work was done in the project plugIT - Business and IT Alignment using a Model-Based Plug-in Framework, funded by the European Union, Intelligent Content and Semantics, 231430

References

Barrett, D.J. (2009), *MediaWiki*, O'Reilly Media Inc. Sebastopol.

DCMI - Dublin Core Metadata Initiative (2008): *Dublin Core Metadata Element Set*, Version 1.1, http://dublincore.org/documents/dces/, retrieved July 5, 2010.

Hinkelmann, K., Merelli, E. and Thönssen, B. (2010): *The Role of Content and Context in Enterprise Repositories*. 2nd International Workshop on Advanced Enterprise Architecture and Repositories - AER 2010.

Karagiannis, D. (1995): *BPMS: Business Process Management Systems*. ACM SIGOIS Bulletin, Vol. 16, No 1, August 1995.

Karagianns, D.; Grossmann, W.; Hoefferer, P. (2008). *Open Models Initiative - A Feasibility Study*. University of Vienna, Department of Knoweldge and Business Engineering, http://cms.dke.univie.ac.at/uploads/media/Open_Models_Feasibility_Study_SEPT_2008.pdf , retrieved July 9, 2010.

Newell, A. (1982). The knowledge level. Artificial Intelligence, Vol. 18, No. 1, 87-127

OASIS (2007): *Web Services Business Process Execution Language* Version 2.0, http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf, retrieved: April 25, 2009.

OMG (2003): MDA Guide Version 1.0.1, http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf, retrieved: July 5, 2010.

OMG (2008): *Semantics of Business Vocabulary and Business Rules (SBVR)*, v1.0, http://www.omg.org/docs/formal/08-01-02.pdf, retrieved: April 15, 2009.

OMG (2009a): *Business Process Modelling Notation*. Version 1.2 http://www.omg.org/docs/formal/09-01-03.pdf, Retrieved: 14.05.2009.

OMG (2009b): *OMG Unified Modelling Language™ (OMG UML), Infrastructure*, Version 2.2, http://www.omg.org/docs/formal/09-02-04.pdf, retrieved: April 15, 2009.

Open Group (2009): *TOGAF Version 9*. http://www.opengroup.org/togaf/

Österle, H., *Business in the information age: heading for new processes*. Springer-Verlag 1995.

Scheer, A.-W. (1999): *ARIS. Business Process Frameworks*. Springer-Verlag.

Woitsch, R., Karagiannis, D., Plexousakis, D., Hinkelmann K. (2009): *Business and IT alignment: the IT-Socket*. Elektrotechnik & Informationstechnik, Springer-Verlag, 126/7

Zachman, J.A. (1987): *A framework for information systems architecture*. IBM Systems Journal, Vol. 26, No. 3.