Metamodelling Platforms

Invited Paper

Appeared in:

Karagiannis, D.; Kühn, H.: Metamodelling Platforms. In: Bauknecht, K.; Min Tjoa, A.; Quirchmayer, G. (Eds.): Proceedings of the Third International Conference EC-Web 2002 – Dexa 2002, Aix-en-Provence, France, September 2-6, 2002, LNCS 2455, Springer-Verlag, Berlin, Heidelberg, p. 182.

Dimitris Karagiannis
University of Vienna
Institute for Computer Science and
Business Informatics
Department Knowledge Engineering
Brünner Str. 72
A – 1210 Vienna
Austria
dk@dke.univie.ac.at
http://www.dke.univie.ac.at

Harald Kühn
University of Vienna
Institute for Computer Science and
Business Informatics
Department Knowledge Engineering
Brünner Str. 72
A – 1210 Vienna
Austria
hkuehn@dke.univie.ac.at
http://www.dke.univie.ac.at

Full version of: Karagiannis, D.; Kühn, H.: Metamodelling Platforms. LNCS 2455, Springer-Verlag, 2002, p. 182.

Metamodelling Platforms

Dimitris Karagiannis and Harald Kühn

University of Vienna, Department Knowledge Engineering, Brünnerstr. 72,
A-1210 Vienna, Austria
{dk, hkuehn}@dke.univie.ac.at

Abstract. The state-of-the-art in the area of modelling of organisations is based on fixed metamodels. Due to rapid changing business requirements the complexity in developing applications which deliver business solutions is continually growing. To manage this complexity, environments providing flexible metamodelling capabilities instead of fixed metamodels has shown to be helpful. The main characteristic of such environments is that the formalism of modelling - the metamodel - can be freely defined and therefore be adapted to the problem under consideration. This paper gives an introduction into metamodelling concepts and presents a generic architecture for metamodelling platforms. Three best practice examples from industry projects applying metamodelling concepts in the area of business process modelling for e-business, e-learning, and knowledge management are presented. Finally, an outlook to future developments and research directions in the area of metamodelling is given.

1 Introduction

Due to rapid changing business requirements such as faster time to market, shorter product lifecycles, increased interdependencies between business partners, and tighter integration of the underlying information systems, the complexity in developing applications which deliver business solutions is continually growing. Therefore, the elements of an enterprise are managed more and more model-based.

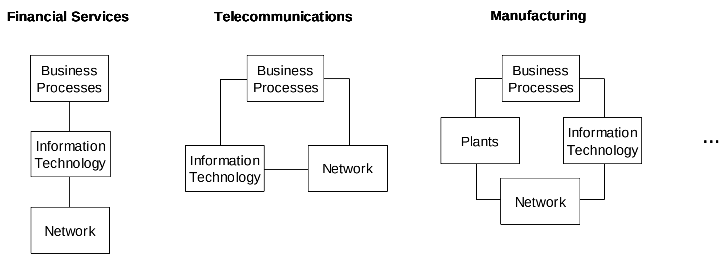

Fig. 1: Branch-specific business architectures

The *state-of-the-art* in the area of modelling of organisations is based on *fixed metamodels*. Product models are created by using product modelling environments, process models are created in business process modelling tools and organisational models are realised in personnel management tools. Web service models link these business models to information technology. They are created by using standardised languages and common ontologies. Information technology is modelled in tools supporting notions such as workflow or object-orientation. The models of the company's strategy, goals and the appropriate measurements are described and monitored by using tools supporting management concepts such as Balanced Scorecard.

Additionally, business architectures depend highly on the branches under consideration. E.g. as the network is a supporting element for doing business in financial services or manufacturing, in the telecommunication industries the network is the essential part of the business model (see figure 1). Branch specific solutions can be seen for example in the enterprise resource planning market, where all major manufacturers offer solutions for different lines of businesses. This causes additional requirements for modelling platforms, such as *integration* mechanisms for different views and aspects under consideration. Other major requirements to an enterprise modelling platform are *flexibility, adaptability, and openness*, to integrate models based on different modelling paradigms such as decision support models, descriptive models, or predictive models. These requirements have to be fulfilled by environments providing flexible metamodelling capabilities. The main characteristic of such environments is that the formalism of modelling - the metamodel - can be freely defined. Platforms based on metamodelling concepts should support the following topics:

1. Engineering the business models & their web services
2. Designing and realizing the corresponding information technology
3. Evaluating the used corporation resources and assets

This raises research issues on how to design, manage, distribute and use flexible metamodels on a syntactic as well as on a semantic level and how to integrate, run and maintain a metamodelling platform in a corporation's environment.

The remainder of the paper is organised as follows. Chapter 2 gives an introduction to general metamodelling concepts. In chapter 3 technologies for metamodelling are presented. In chapter 4 examples of metamodelling in the areas of business process modelling for e-business, e-learning, and knowledge management, are given. Finally, chapter 5 gives an outlook to future developments and research directions.

2 Metamodelling Concepts

Modelling methods consists of two components: a modelling technique, which is divided in a modelling language and a modelling procedure, and mechanisms & algorithms (shorten: mechanisms) working on the models described by the modelling language (see figure 2). The *modelling language* contains the elements, with which a model can be described. A modelling language itself is described by its syntax, semantics, and notation. The *modelling procedure* describes the steps applying the modelling language to create results, i.e. models. In this paper we define a metamodel as a model of a modelling language. Applying language theory for levelling languages, the result is a hierarchy of languages, meta-languages etc. The hierarchy of the corresponding models, metamodels etc. is described in section 2.1. Section 2.2 gives a short overview of the definition of syntax, semantics and notation of modelling languages and section 2.3 describes different roles in metamodelling.

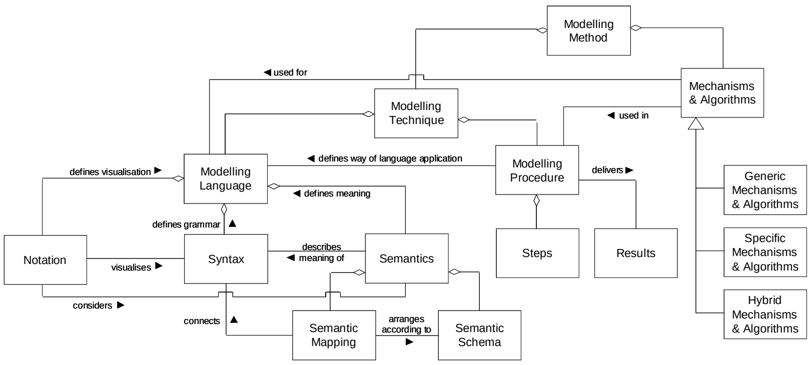

Fig. 2: Components of modelling methods

2.1 Modelling Hierarchy

The creation of a metamodel is also done by using a modelling language. This modelling language is called the *metamodelling language*. The model defining the metamodelling language is the meta-metamodel or *meta^2-model* [8, 23].

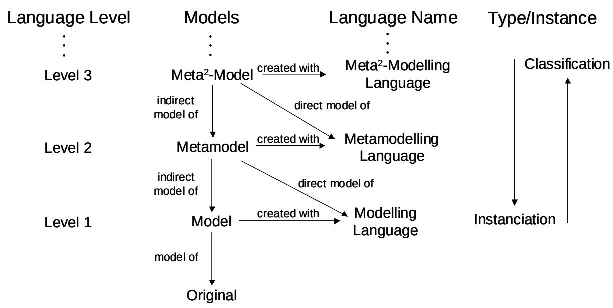

Fig. 3: Metamodelling based on language levels

Building language levels is not limited to a certain level. To “finish” the modelling hierarchy, it is important to find a useful level of abstraction. To use concepts such as “thing”, “property” and “relation” may be helpful, but lack of semantics especially if the language of the “finishing” level should provide the foundation for implementing the lower levels. In practice a four layer metamodel architecture is widely used such as shown in figure 3 [e.g. 5, 9, 15, 18, 19]. The lowest level is the original, from which a model is build on the second level. Often the lowest level is seen as runtime data, but we prefer to use the expression “original” because its not always runtime data from which a model is build. The highest level in the four layer architecture is the meta^2-level, which describes the concepts for building metamodels.

2.2 Syntax, Semantics, Notation, and Mechanisms

A (graphical) modelling language is described by its syntax, semantics, and notation.

The *syntax* describes the elements and rules for creating models and is described by a grammar. For modelling languages two major approaches exist to describe their syntax: graph grammars [21] or metamodels [8]. Often, UML class diagrams are used to describe the metamodel of the syntax. For syntactical rules, which cannot be fully expressed by class diagrams, additional constraint languages are used such as OCL [17, p. 6-1ff] or AdoScript [4, p. 589ff].

The *semantics* describes the meaning of a modelling language and consists of a semantic domain and the semantic mapping. The semantic domain describes the meaning by using ontologies, mathematical expressions etc. The semantic mapping connects the syntactical constructs with their meaning defined in the semantic domain ("semantic schema"). To formulate semantic definitions approaches such as denotational semantics, operational semantics, axiomatic semantics or algebraic semantics are used [8]. Sometimes, only (informal) textual descriptions are used to define the semantics, e.g. in the definition of the UML [17, p. xxviii].

The *notation* describes the visualisation of a modelling language. Static approaches define the symbols for visualizing the syntactical constructs e.g. using pixel-based graphics or vector graphics, but they do not consider the state of the modelling constructs during modelling. Dynamic approaches consider the model state by splitting the notation in a representation part and a control part. The representation part maps to the static approach. The control part defines rules to query the model state and to influence the representation depending on the model state [4, p. 105ff].

Mechanisms provide the functionality to use and evaluate the models built by using the modelling language. Mechanisms can be classified into generic, specific, and hybrid. *Generic mechanisms* are implemented on the meta^2-model, so they can be used for all metamodels based on the meta^2-model. *Specific mechanisms* are implemented for a particular metamodel. *Hybrid mechanisms* are implemented on the meta^2-model, but are adapted to particular metamodels, e.g. to improve usability [15, p. 85f].

2.3 Roles in Metamodelling

Considering the elements of a modelling method described in figure 2 and the components of metamodelling platforms shown in figure 4, different roles in administering and using such platforms can be distinguished.

The *method engineer* is responsible for a consistent and properly defined modelling method. Additional to his technical skills, the method engineer often has professional skills in an application domain. Application domains can be divided into verticals such as financial services, telecommunications, public administration, and manufacturing and horizontals such as business process modelling, application development, workflow management, and knowledge management.

The *language engineer* defines the modelling language. He is responsible for an adequate definition of the syntax, semantics, and notation.

The *process engineer* is responsible for the definition of the modelling procedure. Often the process engineer is an expert in applying modelling languages and has considerable experiences in project management and project execution.

The *tool engineer* configures the mechanisms of a metamodelling platform for particular metamodels. If additional mechanisms are needed, he is the responsible for implementing these mechanisms.

The *infrastructure engineer* provides the necessary IT infrastructure to run a metamodelling platform and to integrate the platform into existing infrastructures.

The *method user* applies the modelling method by using the platform. He creates models by using the modelling language, following the modelling procedure and applying the available mechanisms.

3 Metamodelling Technologies

Section 3.1 presents a generic architecture for metamodelling platforms. In section 3.2 a brief overview of existing metamodelling approaches and platforms is given. Section 3.3 describes the BPMS lifecycle as a framework for metamodelling.

3.1 Metamodelling Architecture

To support the topics mentioned in chapter 1, metamodelling platforms should be realised on a component-based, distributable, and scalable architecture. Figure 4 shows a generic architecture for metamodelling platforms.

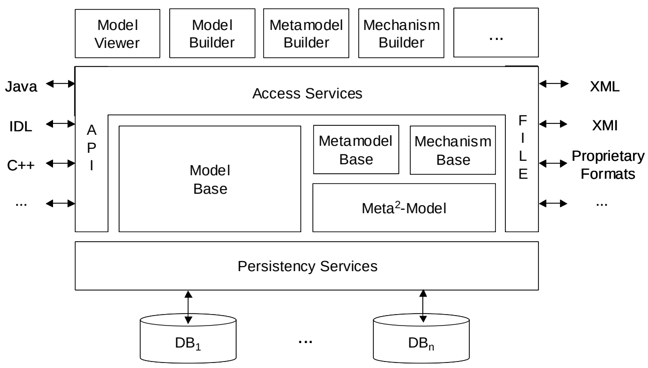

Fig. 4: Generic architecture of metamodelling platforms

The storage of all model and metamodel information is managed by *persistency services*. These services provide transparency of concrete storage types such as specific databases, files systems etc. Furthermore the persistency services enable the distribution of parts of models and metamodels.

The *meta²-model* provides the basic concepts to create metamodels and mechanisms. Typical concepts are "classes", "relations", "attributes", "modeltypes", "scripts" etc. The meta²-model is the central part of the architecture, as it provides the conceptual foundation and is connected with all other parts.

The *metamodel base* contains all information about the metamodels currently managed by the modeling platform. Changes in the metamodel base are delegated to the model base accordingly, to keep the models and their corresponding metamodels consistent.

The *mechanism base* contains information about functionalities to be applied to models and metamodels. These functionalities can be either stored directly in the mechanism base or outside of the metamodelling platform. If they are stored outside, the mechanism base holds only information how to find the appropriate mechanisms e.g. by using external name services.

The *model base* contains all models based on the metamodels. The model base communicates with the metamodel base to track metamodel changes and to forward them to the corresponding models.

Access services provide file-based and online interfaces to the different types of bases. According to access rights the appropriate information from the bases can be queried or even changed.

On top of the access services, different *viewer and builder components* support the usage and maintenance of the metamodelling platform such as model builder, metamodel builder, and mechanism builder.

3.2 Metamodelling Approaches

There exist various metamodelling approaches, different in richness of concepts and ranging from conceptual proposals to already implemented products. In the following, some of them will be illustrated briefly.

Atkinson proposes a modelling hierarchy aligned with the MOF hierarchy [1]. The focus is modelling in the area of distributed object systems. Atkinson stresses the dichotomy of "class" and "instance" which occurs changing the language level and proposes requirements for metamodelling approaches.

Frank proposes within his *MEMO* approach ("multi perspective enterprise modeling") a three level modelling hierarchy. Based on this hierarchy a modelling framework with the same named is suggested [6].

The *Resource Description Framework (RDF)* provides a modelling hierarchy for semantic networks. The foundation of RDF is build by three object types ("resource", "property" and "statement") for representing named properties and property values [16].

The *CASE Data Interchange Format (CDIF)* is based on a four level model architecture [5]. CDIF is a standard designed for the exchange of CASE models between tools of different tool providers. CDIF is not be further developed but major parts of the concepts influence the design of other metamodelling approaches such as the Meta Object Facility (MOF).

The *MOF* is a infrastructure for managing meta information [18]. Conceptually, MOF can be divided into two major parts: (a) the definition and maintenance of meta information based on a four level modelling hierarchy and (b) specifications of interfaces to access the metainformation within a distributed environment.

The *General Modeling Environment (GME)* is based on a four level modeling architecture. In [22] general metamodelling requirements and a approach of model integrated computing (MIC) is proposed.

The *MétaGen* system distinguish a "user metamodel" and a "implementation metamodel". Based on transformation rules between these metamodels, the system development should be more aligned with the requirements definition [20].

Kühn et al. propose a four level modelling hierarchy [15]. Their approach was implemented by BOC in the commercial meta-business process management tool *ADONIS* [10].

Another commercial product is the metaCASE tool *MetaEdit+* from MetaCase Consulting [13]. MetaEdit+ is a configurable CASE tool, based on a metamodelling approach.

3.3 Metamodelling lifecycle according to BPMS

The Business Process Management Systems paradigm (BPMS paradigm) is a general framework for Business Engineering [10, 11]. The lifecycle of the framework consists of five subprocesses. Each subprocess has different requirements and needs from the models and metamodels managed in a metamodelling platform.

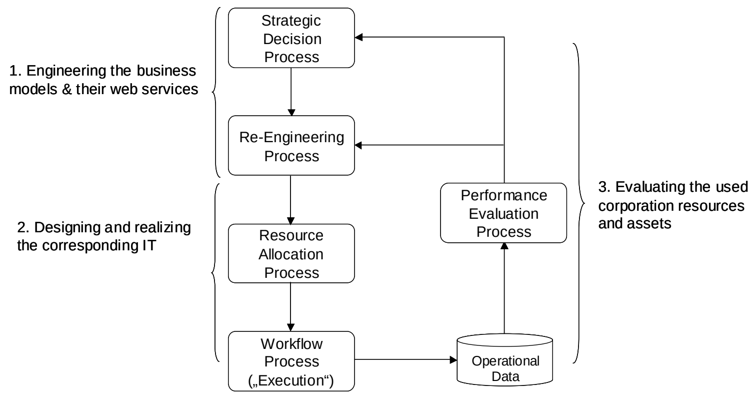

Fig. 5: The BPMS lifecycle

In the *Strategic Decision Process* the corporation's goals, the business models and the business requirements are defined. These influence the selection of the modelling method and the underlying modelling language, procedures and mechanisms. E.g. if models should deliver quantitative information about the business performance, then the metamodels should contain concepts such as “time”, “cost”, “volume”, “frequency” etc.

In the *Re-Engineering Process* the web services supporting the business models are designed and the necessary business applications are identified. The results of the Re-Engineering Process form the requirements to be implemented in the Resource Allocation Process.

Depending on the target systems and execution environments used in the *Workflow Process*, different modelling languages in the *Resource Allocation Process* are used. E.g. is the execution environment an ERP system such as SAP R/3, the modelling language should contain event driven process chains (EPC) to represent reference models helpful for customising purposes. Should the target system be developed in an object-oriented way, then the modelling language should contain diagram types from the UML etc.

The operational data produced by the web services and business applications during runtime are input for the *Performance Evaluation Process*. There, adequate mechanisms must be available to evaluate the used resources and assets, and proposals for business improvement have to be gained using the models and mechanisms.

4 Application Scenarios

This chapter provides examples of applying metamodelling concepts to support organisational and application development. The examples mainly come from EU funded projects all realized with industrial partners. Section 4.1 describes a scenario of business process modelling for e-business. Section 4.2 sketches a best practice approach in the area of e-learning, and in section 4.3 the PROMOTE approach for process oriented knowledge management is described.

4.1 Business Process Modelling for E-Business

In the modelling of business processes for e-business we distinguish two general steps: the first step is the design and optimisation of the business processes of the domain under consideration (section 4.1.1). In the second step the specific type of e-business has to be taken into account, which influences the overall business process design (section 4.1.2). Both steps will be explained by examples from the insurance sector.

4.1.1 Design and Optimisation of Application Domain

Business process modelling and optimisation is a continually task to streamline and steadily improve the way a corporation is working. The ESPRIT project *REFINE*, which finished 1997, was designed to *help the insurance sector to perform ”business process re-engineering”* (BPR) [24]. Core business processes, like underwriting and claims handling have been transformed within REFINE directly at the insurance sites within real pilots. The main idea of REFINE was developing and refining a generic methodology within pilot scenarios. The pilot scenarios were:

a. Exchange of standard EDI messages between the insurer and information providers for credit insurance.
b. Redemption of life insurance and data processing from sales to local offices.
c. Decentralised policy issuance for motor insurance.

For refining the methodology and the used modelling languages, metamodelling was used. For each participating insurance company an individual business process modelling language was designed. The specific languages contain modelling concepts fitting well with each of the scenario mentioned above such as information objects describing EDI messages, constructs for modelling distributed organisations, and performance measurements depending on the company under consideration.

Another major advantage was the possibility to use an organisation's terminology within the modelling languages by configuring the metamodels accordingly. This saved time and costs because of less training effort and improved acceptance because of direct identification with the terminology used.

4.1.2 Modelling of Specific E-Business Types

The type of e-businesses such as B2B, B2A, B2C, B2E etc. has to be considered in the business process design. Using Internet technologies many companies develop new business models to realize a tighter integration of their business partners and customers. Insurance companies for example develop such business models either to reduce their costs of administration or to establish new channels of distribution networks.

The following example describes e-business type specific modelling within a *B2B sales and distribution platform for insurance partners* ("insurance portal"). The main objective of the Internet platform is to support insurance agents to reduce the cycle time and costs of administration, which arose from the interaction with insurance companies. Additionally, agents should have more time to offer best advice to the consumer. The modelling framework applied in this project is *E-BPMS*, which was implemented by using the metamodelling tool ADONIS. In the following, the application of the framework will be described shortly, more details can be found in [2].

At the beginning of the project the business strategy was mapped to a *business model* and questions such as the following had to be answered in the strategic level:

a) Which are the processes and services (products) to be realized on the platform?
b) Which are the appropriate business partners to develop and run the platform?
c) Corresponds the business plan of the project with the business plans of each participant?

On the business level process categories are modeled in a *process map*. Process categories are refined by *business process models*. Business processes can be divided into the following types:

- insurance *core service processes*, e.g. application processes, claims management,
- *value adding processes*, e.g. cash management processes, event management,
- *development processes*, e.g. business and software development based on the core elements "products", "processes", "organisational units" and "information technology",

- *business operations processes*, e.g. process integration of business partners and
- *additional services*, e.g. legal advisor services, training and learning.

After finishing the requirement definition, the next level is the implementation level. The platform consists of a core service application and static HTML pages developed in a content management system. The *interaction process model* is part of this level to structure the navigation through the e-business application. Complex application modules are modeled in an object-oriented modeling language or are integrated as components (e.g. portal technology, security components) without modeling the details. Well-defined entry points are modeled in the flow of interactions. For each action in the interaction process model particular user interface elements can be assigned. In this way user interfaces can be designed and inspected before considerable amounts of money are invested in the implementation. The so established site map (swimlanes in the interaction model) can be imported in a content management system for the generation of site templates.

Based on the final application, tests are executed. To manage the test and production environment, the *IT infrastructure model* is used. Administrative processes specify the way releases are uploaded in the production environment. These processes and the responsible actors modeled in the business process models are linked with the corresponding environment in the IT infrastructure model. Additionally, the infrastructure model is used to manage the complexity of IS operations. To evaluate the interaction process and the IT infrastructure simulation algorithms can be applied. Results are cycle times, response times, resources allocated by the platform application etc.

The execution level of the B2B insurance platform is influenced by short release cycles - especially driven by short term content as news and events and by high fluctuation of platform users. Business operation processes such as content management processes, user management, and first and second level support, are documented by exporting all required information in a process based operating instructions manual. This manual is online available for the responsible operators and support agents.

4.2 E-Learning

The *ADVISOR* project, which was finished in the year 2000, dealt with *new ways of learning and training methods* in the field of business process re-engineering in the insurance sector and was the successor of REFINE [24]. Frequent changes in business processes, resulting from new products and the adaptation of existing products to new market situations, require tool-based methods in order to provide individuals and teams quickly with the appropriate information for their tasks. In addition, measures for (re-)training staff should be derivable as quickly as possible. In order to capitalise on employees' knowledge, creativity and experience, they should be enabled to provide input to their company's knowledge in a systematic and motivating manner. Starting from these business needs, three main issues were addressed in the ADVISOR project:

a. Improved access for employees to company and performance related information,
b. rapid, semi-automatic production of training materials, and
c. knowledge acquisition for organisational learning.

The first objective of ADVISOR was to provide methods and tools which allow for a holistic approach to information access, training and learning by closely coupling business re-engineering measures with training/learning measures. The second objective was to improve upon the psychological and organisational measures which are necessary to change the attitude towards continuous learning and to lead to better acceptance of new technology and processes. Both objectives were realised on three levels of learning: individual, team, and organisation.

In order to realise these objectives, the project built upon existing business process management methods and tools, which were extended by metamodelling in order to specify information and training needs for employees and to capture employees' experiences with business procedures and training measures. Extensive trial studies with and formative evaluation of the extended technology in the insurance companies accomplished the second main objective.

4.3 Knowledge Management

There is a significant gap between the importance of knowledge management and the realisation on all levels in an organisation: There are many surveys that show that knowledge management is recognized as a management task with high priority. When looking at concrete projects and initiatives, however, knowledge management receives much less attraction. Lack of time is a main reason that knowledge workers mention when asked why they do not support knowledge management.

To overcome these barriers the *PROMOTE* project [12], which will be finished in autumn 2002 with two industrial trial cases, provides solutions to two critical challenges of knowledge management:

a. *integration with the operational business:* knowledge management tasks are associated with knowledge-intensive activities in business processes
b. *providing access to available knowledge:* explicit graphical knowledge structures help to get an immediate overview of available knowledge.

PROMOTE is a process-oriented and metamodelling-based approach to knowledge management using the concept of a flexible organisational memory information system to store relevant information and provide pointers to people with relevant know how. Within the project a modelling language was designed to deal with the above-mentioned content and context characteristics. Amongst others, the modelling language contains:

- *Topic maps* which are semantic networks consisting of knowledge objects (topics) and relations between them. A graphical representation of topic maps helps a knowledge seeker to navigate in the organisational memory. If, for instance, he is looking for knowledge about cancer, a medical topic map shows all the related topics like smoking etc. Thus the knowledge seeker gets hints about relevant knowledge he/she did not think of.

- *Skill models* relate topic maps to people and represent the skill status of a particular person with respect to topics in a topic map.
- *Process models* represent the work context. Knowledge objects and people can be associated to knowledge-intensive activities via so-called knowledge processes.

Using a metamodelling approach, time and implementation costs were saved. Additionally, the resulting modelling languages are highly applicable and accepted because of repeated adaptations after several quality reviews.

5 Summary, Future Developments, and Research Directions

Metamodelling concepts and metamodelling platforms are getting more and more an *integral part of business engineering* strategies and approaches. Prominent examples are the international standards UML and MOF, which are both based on a four level metamodelling approach [17, 18]. In addition, this trend is underpinned by metamodelling products already available such as ADONIS or MetaEdit+ [4, 10, 13]. The *major advantages* from out experiences using flexible metamodel approaches instead of approaches using fixed metamodels are considerable savings in time and costs in application development, increased quality of delivered solutions, and enhanced acceptance because of directly mapping the domain under consideration.

Nevertheless, metamodelling is still a very challenging field for innovative future developments and essential research activities. Some of the developments and research directions we are expecting are:

- *Integration and interoperability:* The integration of heterogeneous systems to interoperable systems is part of enterprise application integration (EAI) efforts. In addition to technical integrations, the systems have to be integrated on a semantically level [25]. Coordinated metamodels, integration of ontologies, and enterprise model integration (EMI) give rise to further research.
- *Semantic Web:* The vision stated by Berners-Lee [3] aims at developing languages for expressing information in the WWW in a machine understandable form. Currently, most information in the Web is for human consumption. Promising efforts such as RDF are based on metamodelling concepts [19].
- *Model-driven Business Engineering:* Managing organisations and developing large enterprise applications causes complex interdependencies between different parts of organisations and applications. Often these parts are managed and realized by using different technologies and, if any used, different modeling environments. Chaining models for business, development and evaluation ("straight through business engineering") to measure and control business decisions based on operational data generated by business applications is of vital research interest [14].
- *Combination of modelling paradigms:* Modelling paradigms used in the IS development field are mostly descriptive. Other paradigms such as decision support models and predictive models are often used focusing on the business do-

main. We expect strong interest in combining these approaches by metamodelling to form new possibilities in enterprise management and development.

- *Language Engineering:* The definition of "good" modelling languages and their implementation in helpful software support still need a lot of experience and knowledge. To capture these experiences, patterns could be an appropriate formalism [7]. E.g. the current definition of semantics of modelling languages is either informal, and therefore often error prone and not directly understandable by machines, or formal, i.e. very time-consuming and expensive. In this area we are expecting improvements by interdisciplinary research.

References

1. Atkinson, C.: Metamodeling for Distributed Object Environments. In: Proceedings of the 1st International EDOC Workshop (1997)
2. Bayer, F.; Kühn, H.; Junginger, S.; Petzmann, A.: E-BPMS: Ein Modellierungs-Framework für E-Business-Anwendungen. In: Bauknecht, K.; Brauer, W.; Mück, Th. (Eds.): Proceedings of the Informatik 2001 - Wirtschaft und Wissenschaft in der Network Economy: Visionen und Wirklichkeit. University of Vienna, Austria (2001) 922-927
3. Berners-Lee, T.: Semantic Web Road Map. W3C Design Issues. http://www.w3c.org/DesignIssues/Semantic.html, Access 2002-08-05
4. BOC: ADONIS 3.6 Manual IV: Method Definition and Administration. BOC Information Technologies Consulting GmbH, http://www.boc-eu.com (2002)
5. Electronic Industries Association - CDIF Technical Commitee: CDIF CASE Data Interchange Format - Overview. EIA-IS-106 (1994)
6. Frank, U.: Multi-Perspective Enterprise Modeling (MEMO) - Conceptual Framework and Modeling Languages. In: Proceedings of the 35th Hawaii International Conference on System Sciences, Hawaii (2002)
7. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns – Elements of Reusable Object-Oriented Software. Addison-Wesley (1994)
8. Geisler, R.; Klar, M.; Pons, C.: Dimensions and Dichotomy in Metamodeling. Technical Report 98-5, Technical University Berlin (1998). http://tfs.cs.tu-berlin.de/~espress/doc/own/GKP98.ps, Access 2002-08-05
9. International Organization for Standardization: Information Technology – Information Resource Dictionary System (IRDS) Framework. ISO/IES 10027 (1990)
10. Junginger, S., Kühn, H., Strobl, R., Karagiannis, D.: Ein Geschäftsprozessmanagement-Werkzeug der nächsten Generation – ADONIS: Konzeption und Anwendung. In: WIRTSCHAFTSINFORMATIK 42/5 (2000) 392-401
11. Karagiannis, D., Junginger, S., Strobl, R.: Introduction to Business Process Management Systems Concepts. In: Scholz-Reiter, B., Stickel, E. (eds.): Business Process Modelling. Springer-Verlag, Berlin Heidelberg New York (1996) 81-106
12. Karagiannis, D., Prackwieser, C., Telesko, R.: The PROMOTE project: Process oriented knowledge management. In: Goncalves, R., Steiger-Garcao, A., Scherer, R. (eds.): Proceedings of the 3rd European Conference on Product and Process Modeling (2000) 217-224
13. Kelly, S., Lyytinen, K., Rossi, M.: MetaEdit+ - A Fully Configurable Multi-User and Multi-Tool CASE and CAME Environment. In: Constantopoulos, P., Mylopoulos, J., Vassiliou, Y. (eds.): Advanced Information System Engineering, Proceedings of the 8th International Conference CAiSE'96. Springer-Verlag, Berlin Heidelberg New York (1996) 1-21

14. Kühn, H., Junginger, S., Bayer, F.: How Business Models Influence the Development of E-Business Applications. In: Standford-Smith, B., Kidd, P. (eds.): Proceedings of the eBusiness and eWork Conference 2000. IOS Press (2000) 1024-1030
15. Kühn, H., Junginger, S., Karagiannis, D., Petersen, C.: Metamodellierung im Geschäftsprozeßmanagement: Konzepte, Erfahrungen und Potentiale. In: Desel, J., Pohl, K., Schürr, A. (eds.): Proceedings of Modellierung'99. Teubner-Verlag, Stuttgart Leipzig (1999) 75-90
16. Lassila, O., Swick, R. R.: Resource Description Framework (RDF) Model and Syntax Specification. REC-rdf-syntax-19990222, W3C (1999)
17. Object Management Group: OMG Unified Modeling Language Specification, Version 1.4 (2001). http://www.omg.org/cgi-bin/doc/?formal/01-09-67.pdf, Access 2002-08-05
18. Object Management Group: Meta Object Facility (MOF) Specification, Version 1.4 (2002). http://www.omg.org/cgi-bin/doc/?formal/02-04-03.pdf, Access 2002-08-05
19. Pan, J. Z., Horrocks, I.: Metamodeling Architecture of Web Ontology Languages. In: Proceedings of the Semantic Web Working Symposium (2001) 131-149
20. Revault, N. M., Sahraoui, H.A., Blain, G., Perrot, J.-F.: A Metamodeling Technique: The MétaGen System. In: Proceeding of the Tools Europe'95 Conference (1995) 127-139
21. Rozenberg, G. (ed.): Handbook of Graph Grammars and Computing by Graph Transformation - Volume 1: Foundations. World Scientific Pub. Co. (1997)
22. Sprinkle, J., Karsai, G., Ledeczi, A., Nordstrom, G.: The New Metamodeling Generation. In: Proceedings of the 8th Annual IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (2001)
23. Strahringer, S.: Metamodellierung als Instrument des Methodenvergleichs. Eine Evaluierung am Beispiel objektorientierter Analysemethoden. Shaker-Verlag, Aachen (1996)
24. Strobl, R., Karagiannis, D., Heidenfeld, M.: How an SME can spin of from a university with support from ESPRIT. In: Proceedings of the EMMSEC'98, Bordeaux (1998)
25. Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hübner, S.: Ontology-Based Integration of Information – A Survey of Existing Approaches. In: Stuckenschmidt, H. (ed.): Proceedings of the Workshop „Ontologies and Information Sharing" IJCAI'2001 (2001) 108-117