Computers in Industry 79 (2016) 77–86

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

A new paradigm for the continuous alignment of business and IT: Combining enterprise architecture modelling and enterprise ontology

Knut Hinkelmann [a,b,*], Aurona Gerber [c,b], Dimitris Karagiannis [d], Barbara Thoenssen [a], Alta van der Merwe [b], Robert Woitsch [e]

[a] School of Business, FHNW University of Applied Sciences and Arts Northwestern Switzerland, 4600 Olten, Switzerland
[b] Department of Informatics, University of Pretoria, Pretoria, South Africa
[c] CSIR Meraka, Pretoria, South Africa
[d] Knowledge Engineering Research Group, University of Vienna, Vienna, Austria
[e] BOC Asset Management GmbH, Vienna, Austria

ARTICLE INFO

Article history:
Received 27 June 2015
Accepted 16 July 2015
Available online 25 August 2015

Keywords:
Business-IT alignment
Enterprise engineering
Enterprise architecture
Enterprise ontology
Metamodelling

ABSTRACT

The paper deals with Next Generation Enterprise Information Systems in the context of Enterprise Engineering. The continuous alignment of business and IT in a rapidly changing environment is a grand challenge for today's enterprises. The ability to react timeously to continuous and unexpected change is called agility and is an essential quality of the modern enterprise. Being agile has consequences for the engineering of enterprises and enterprise information systems. In this paper a new paradigm for next generation enterprise information systems is proposed, which shifts the development approach of model-driven engineering to continuous alignment of business and IT for the agile enterprise. It is based on a metamodelling approach, which supports both human-interpretable graphical enterprise architecture and machine-interpretable enterprise ontologies. Furthermore, next generation enterprise information systems are described, which embed modelling tools and algorithms for model analysis.

1. Introduction

The paper deals with Next Generation Enterprise Information Systems in the context of Enterprise Engineering (EE). Giachetti [1] defines Enterprise Engineering as "the body of knowledge principles and practices to design an enterprise" where an enterprise is a "complex socio-technical system that comprises interdependent resources of people, information, and technology that must interact with each other and their environment in support of a common mission".

The ability of keeping up with continuous and unexpected change is an essential quality of modern enterprises and will become a necessity for existence. Dove [2] calls this characteristic *agility* and defines it as "the ability of an organization to thrive in a continuously changing, unpredictable business environment." The concept of the agile enterprise emerged in the early 1990s [3]. An agile enterprise rapidly adapts to changing business challenges and opportunities and it continuously improves to optimize costs, quality and speed of delivery. It enables top management to quickly implement new strategies and control key business parameters to gain a competitive advantage [4], which means that enterprise engineering is an ongoing activity. An enterprise is not designed just once, but an enterprise is, to varying degrees, redesigned many times [1]. Implemented business processes and information systems have to be continuously adapted. As changes may be triggered from the business as well as from developments in the technology, a *continuous alignment of business and IT* is needed.

The pace of change is continuously accelerating and managing the change is increasingly beyond the control of companies. The rate of technological progress increased throughout history. For example, in the car industry new models are developed within few months instead of years. In the banking industry, the time to market for new financial products is a few weeks instead of months [5]. Each new product or service requires new or adapted processes and information systems to produce the products and to deliver the services. Reduced time to market increases the demand for

* Corresponding author at: FHNW University of Applied Sciences and Arts Northwestern, School of Business, Olten, Switzerland. Tel.: +41 62 957 23 01.
E-mail addresses: knut.hinkelmann@fhnw.ch (K. Hinkelmann), AGerber@csir.co.za (A. Gerber), dk@dke.univie.ac.at (D. Karagiannis), barbara.thoenssen@fhnw.ch (B. Thoenssen), alta.vdm@up.ac.za (A. van der Merwe), robert.woitsch@boc-eu.com (R. Woitsch).

http://dx.doi.org/10.1016/j.compind.2015.07.009
0166-3615/

changes of business processes and information systems. Considering the multiyear nature of many enterprise engineering initiatives, the architecture at the start of a development might not be appropriate anymore when the new business processes and information systems are rolled out.

The grand challenge for today's enterprises, which is deal with in this research, is the *continuous alignment of business and IT in a rapidly changing environment.* According to Gartner [6] enterprises are facing a new era of enterprise IT, the 'digitalization' era, "a period characterized by deep innovation beyond process optimization, exploitation of a broader universe of digital technology and information, more-integrated business and IT innovation, and a need for much faster and more agile capability".

In order to deal with this grand challenge an approach is proposed, which uses model-based engineering as visualized in Fig. 1.1. The approach builds on the principles of model-driven enterprise engineering [7] and is supplemented with two innovative and challenging developments:

- Shift the paradigm of model-driven engineering from *development* to *continuous adaptation.* In contrast to software development it is unusual for enterprise engineering to follow a greenfield approach and start from scratch. Instead, typically a 'running' enterprise is adapted. The challenge is to react on change in the business (e.g. due to an altered business strategy) and IT (e.g. due to innovative technology) alike and to continuously keep business and IT aligned. Models are used for designing and adapting enterprises and enterprise information systems before they are changed in reality.
- Support machine interpretable and human interpretable models: McCauley [8] defines an agile organization as "one that can sense opportunity or threat, prioritize its potential responses and act efficiently and effectively". In order to support in sensing, prioritizing and acting, the models should not only be passive storage of knowledge intended for human use but model processing in this context also demands automated operations on models that retrieve and interpret information for decision making. The focusing on machine interpretable knowledge is called knowledge engineering (KE) [9] and is distinguished from knowledge management (KM), which is focusing on human interpretable knowledge. The challenge is to keep both representations consistent.

To meet these challenges a metamodel approach for next generation information systems is proposed, which builds on the knowledge engineering for business process management presented in [9]. These are the main characteristic of the approach:

- Graphical notations are provided, which can easily be understood by humans.
- Semantic lifting makes the semantics of metamodels explicit [10,11] such that the analysis, adaptation and evaluation of models can be done by a machine. The semantics of the metamodel is specified by an ontology.

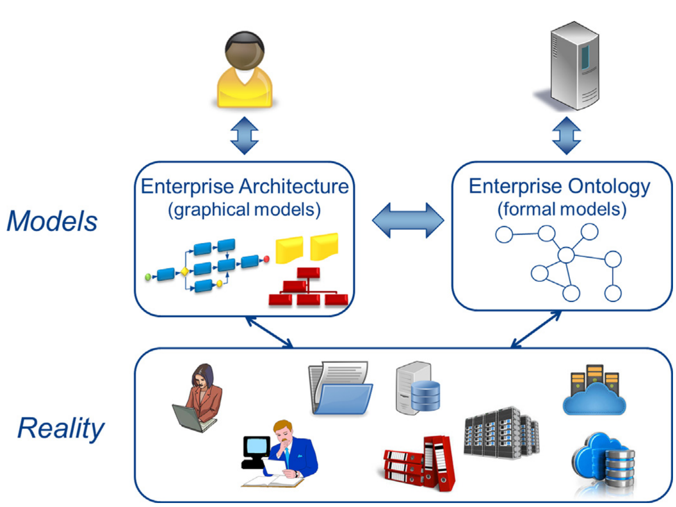


Fig. 1.1. The modelling approach for continuous business-IT alignment.

The next section provides some background information as well as more detail on metamodelling. Solutions, which are already available to realize the proposed approach for the next generation enterprise information systems, are discussed. Challenges that still need to be solved in order to fully realize this approach are highlighted. In Section 3 elements of modelling methods are explained. Then in Section 4 the modelling method for continuous business-IT alignment is presented. Finally in Section 5 the contribution is summarized and an outlook on future work is given.

2. Background

In this section background information in relevant topics for continuous alignment of business and IT is provided. First different types of agility are discussed followed by an explanation of business–IT alignment in the context of an agile enterprise. Lastly background on enterprise modelling is provided and showing how it supports the alignment of business and IT.

2.1. Enterprise agility

Cummins [4] divides agility into four dimensions: dynamism, adaptability, flexibility and awareness.

- *Dynamism* is defined as the ability to change the process definition of an enterprise. The need to change a process definition may result from process improvements to process innovation or process reengineering.
- *Adaptability* is the ability of an enterprise to react to exceptional circumstances or unexpected events during the performance of a process instance, which may or may not be foreseen.
- *Flexibility* is the ability to deal with a fair degree of uncertainty.
- *Awareness* is the ability to detect opportunities and risks.

Reichert and Weber [12] also distinguish between four types of agility[1] needs in Process Aware Information Systems (PAIS) namely variability, looseness, evolution and adaptation. *Evolution* represents the ability of the process implementations to change. Since business processes can evolve over time, it is not sufficient to implement them once and then to never touch the PAIS again. Evolution is equivalent to the dynamism in [4] and in this paper the focus is on evolution/dynamism as well as awareness.

2.2. Complexity and change

In order to identify the need for changes, an organization has to continuously monitor itself and be prepared to react quickly to threads and opportunities. However, the challenge to react quickly is increased by the complexity of today's IT. According to Dietz [13] the most dominant problem identified in scientific as well as in popular science on enterprise management, is complexity and how it can be managed. He claims that because of the complexity of

[1] Reichert and Weber call it "flexibility" but here the term "agility" is used in order to be consistent in naming.

enterprises a conceptual model is needed that "only shows the essence of the operation of an enterprise" and therefore "the model abstracts from all realization and implementation" [13]. Hence, Chen et al. [14] consider enterprise architecture as the foundation of enterprise systems engineering with the goal to support stakeholders of an enterprise to manage system engineering and changes. Zachman regards enterprise architecture as the determinant of survival in the Information Age in order to deal with increased complexity and change of enterprises [15].

2.3. Enterprise architecture (EA)

There are various definitions of enterprise architecture (EA). A definition that is in line with the ISO/IEC/IEEE 42010 standard [16] defines an enterprise architecture as "fundamental concepts or properties of an enterprise in its environment embodied in its elements, relationships, and in the principles of its design and evolution." An enterprise architecture is typically described using models. An architecture description is a work product used to express architecture. The description of the enterprise architecture is a helpful and necessary tool to understand complexity and manage change [17].

Due to the complexity of an enterprise architecture description, many frameworks were developed to assist in this task. A framework is a logical structure for classifying and organizing complex information. An architecture framework is defined by the ISO/IEC/IEEE 42010 standard as "conventions, principles and practices for the description of architectures established within a specific domain of application and/or community of stakeholders" [16].

There is a huge variety of EA frameworks (EAF). Matthes [18] points out that to date more than 50 frameworks are available. In his compendium Matthes gives a detailed description of 34 EA frameworks, based on clearly structured and well defined criteria. Here two EA frameworks are briefly mentioned, which are widely used. The purpose is to show that although the content is comparable the structure of the frameworks can differ.

The Zachman framework is of particular interest because according to [18] it is widespread with an approximate market share amounts between 22% and 25% and builds the basis for many other frameworks. The Zachman Framework is a two dimensional matrix [19]. Rows depict different perspectives of the role a stakeholder may take (named planner, owner, designer, builder and subcontractor), and columns represent the various aspects that should be considered. They are "different abstractions from or different ways to describe the real world" ([20] p. 592). The aspects (rows) are named based on the fundamentals of communication. The interrogatives What (data), How (function), When (time), Who (people), Where (network), and Why (motivation) build the basis for the concise description of complex ideas [19].

TOGAF is another well-known EA framework [21]. The overall enterprise architecture as composed of a set of closely inter-related architectures: Business Architecture, Information Systems Architecture (comprising Data Architecture and Application Architecture), and Technology (IT) Architecture [22].

2.4. Enterprise architecture descriptions

Zachman gives no advice on how the enterprise architecture description should look: intersections of perspectives and aspects can be represented in models of various model types, like a data model or a process model. Those model types can in turn be represented in various languages. OMG has developed several specialized modelling languages for enterprise architecture modelling, for example Business Process Model and Notation (BPMN) [23], Case Management Model and Notation (CMMN) [24], and the Business Motivation Model (BMM) [25]. The purpose of these graphical modelling languages is to support communication between human stakeholders. They are not intended for machine interpretation—although there does exist execution engines for BPMN.

The ArchiMate Standard [22] introduces an integrated language for describing enterprise architectures. ArchiMate fits into the TOGAF framework as it provides concepts for creating a model that correlates to its three architectures (layers). According to [26] ArchiMate can be used to describe all aspects of the EA in a coherent way, while tailoring the content for a specific audience.

ArchiMate provides a graphical representation of its language elements based on UML class diagram but customized and limited to a small set of modelling constructs in the interest of simplicity of learning and use. The standard claims that architecture descriptions "are formal descriptions of an information system, organized in a way that supports reasoning about the structural and behavioural properties of the system and its evolution." [22]. However, the ArchiMate language has one shortcoming: it is intended for human interpretation and not suitable for *automatic* reasoning for two reasons. It is too coarse grained as it only contains basic concepts and relationships that serve general enterprise architecture modelling purposes [28].

2.5. The enterprise engineering knowledge space

Models are representing part of reality or a vision in an agreed modelling language. The term "knowledge space" is used to name what is represented in a model. The actual knowledge space represented in models is specified according to the four dimensions *form, content, interpretation*, and *use* (see Fig. 2.1).

- The *form* represents the syntax and semantic.
- The *content* is seen as the domain in which knowledge engineering is applied. In the area of Enterprise Engineering the content comprises the enterprise architecture, which can be the as-is architecture or a planned to-be architecture. The model is then the enterprise architecture description.
- Depending on the intended *use* only a subset of the knowledge space's content might be of interest. Views and viewpoints are a means to specify which part of an architecture description is of relevance for one or more stakeholders to address specific concerns.
- The representation of knowledge is either focused on machine *interpretation* or on human *interpretation*. In this context *enterprise architecture* is typically represented by the means of graphical models which typically are cognitively more adequate for *human interpretation* and *enterprise ontologies*, on the other hand, are formal representations which can be *interpreted by machines*.

In the following sections the human-interpretable, graphical modelling is referred to as enterprise architecture and to

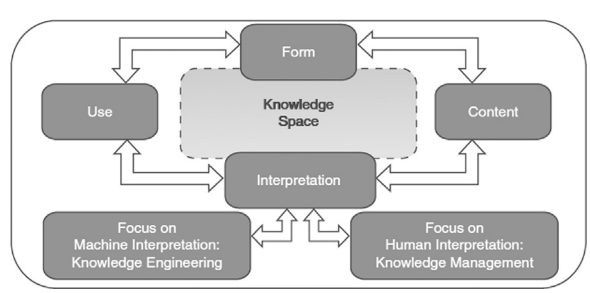


Fig. 2.1. The four dimension of a knowledge space [9].

machine-interpretable, formal representations as enterprise ontology. A modelling method for Model-Driven Enterprise Engineering is proposed that allows for describing the knowledge space in both a human-interpretable form (enterprise architecture models) and machine-interpretable form (enterprise ontology).

2.6. Enterprise ontology (EO)

Because of the complexity of enterprise architecture, machine intelligibility of enterprise architecture descriptions is considered essential for agile enterprises. A machine-understandable and interpretable architecture description would allow to answer questions like "which processes are affected by the replacement of an application?", "which roles are involved in the process?", "why did we decide to customize this specific application?"

As shown by [29,30] an enterprise ontology (EO) could meet this request. Describing enterprise architecture as an ontology started in the 1990s with TOVE [31], The Edinburgh Enterprise Ontology [32] and the organizational memory [33]. In contrary to EA enterprise ontologies are concrete representations of (generalized) enterprise architectures developed to be re-used in enterprises [34], adopted and enhanced to an enterprise's specific needs. Den Haan [35] has used an enterprise ontology to realize a Model-Driven Enterprise Engineering. ArchiMEO is an example of an enterprise ontology based on the ArchiMate standard. It contains the concepts of ArchiMate 2.0 and extends them by more generic concepts to express more specific elements, for example activities or types of business actors.

The advantage of having an ontological representation of an enterprise architecture that is machine understandable and hence allows for automation was proved in two research projects. One building an early warning system for risks in the supply chain [36]; the other linking enterprise architecture description with operational databases to provide an integrated view and management of enterprise entities spread over various data stores, represented in different ways and levels of granularity [37].

There are a variety of representation formalisms for ontologies which allow for machine interpretations. Recent approaches like RDFS and OWL were developed in the context of the semantic web [38]. There is no 'right' language to formally describe an enterprise ontology. The "choice of the language to use in a system or analysis will ultimately depend on what types of facts and conclusions are most important for the application" [39]. It is not the purpose of this paper to propose the appropriate ontology representation formalism. This is left to future research.

3. A model-based approach for enterprise engineering

In the previous chapter the background of human-interpretable and machine-interpretable enterprise modelling was provided. In this section the basis for integrating these two modelling approaches is presented. According to [41] a modelling method consists of a modelling technique, which is further divided into a modelling language and a modelling procedure, as well as modelling mechanisms and algorithms [41]. The components of a modelling method and their relations are visualized in Fig. 3.1. Each of the main components (modelling language, modelling procedure and mechanism and algorithms) is accented with a different colour. In this section each of the three main constituents in the context of enterprise engineering are explained.

3.1. Modelling language

A modelling language is defined by *syntax*, *semantics*, and *notation* that provide the necessary modelling primitives in order to build the model. The concepts that describe the modelling

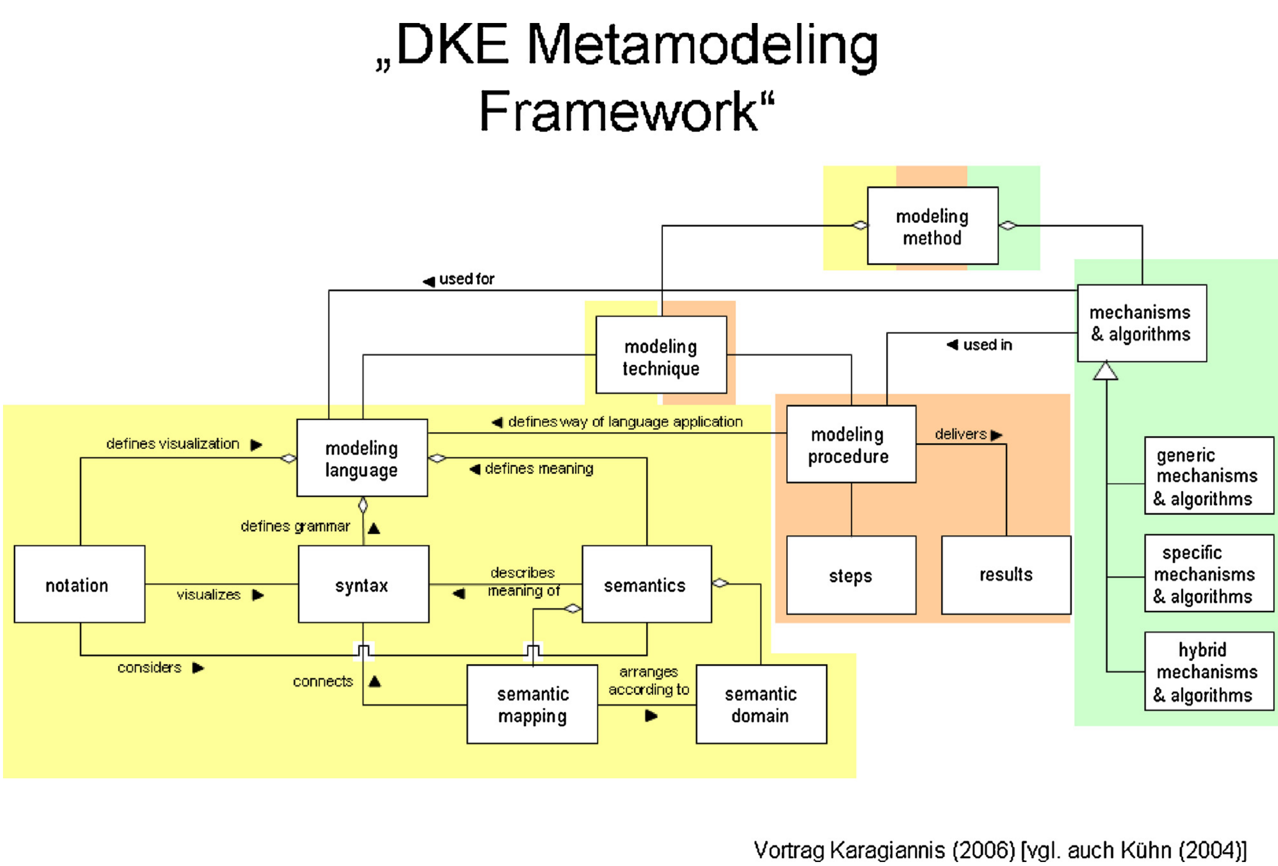


Fig. 3.1. Components of modelling methods [41].

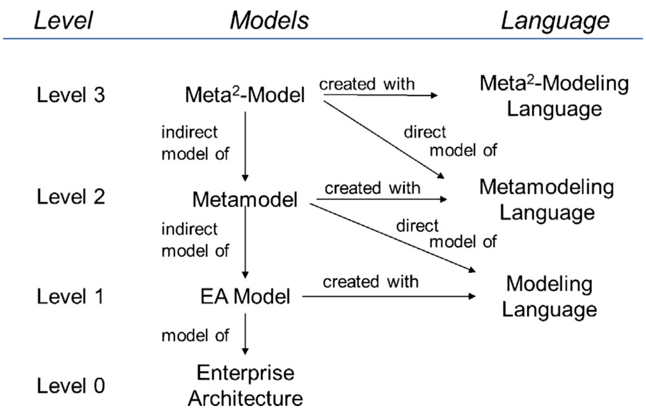


Fig. 3.2. Layered model stack for Enterprise Architecture (adapted from [41]).

language are defined in the metamodel language, which itself has to be represented in a modelling language. Fig. 3.2 shows a layered model stack, which was proposed by Strahringer [42] and adapted by Karagiannis (e.g. [9,41]). The stack could be extended indefinitely but typically four layers are sufficient.

A prominent metamodelling framework is the Meta Object Facility (MOF), an OMG standard for Model-driven Engineering [43]. MOF is based on the UML infrastructure and thus it is a candidate for object-oriented enterprise modelling. MOF metamodels for modelling languages like BPMN, CMMN or BMM are typically modelled as UML class diagrams. ArchiMate is also a modelling language created with UML as metamodelling language. MOF has also been used to define metamodels for Ontology languages like OWL and RDFS [44]. MOF expresses abstract syntax and semantics but does not support the definition of the graphical representation, i.e. the notation or concrete syntax. Thus, MOF is applicable to define metamodels for a machine-interpretable modelling language, i.e. an *enterprise ontology*.

ADOxx® is a meta-metamodelling framework for defining graphical modelling languages. It has been researched at the University of Vienna (see for example [45–47]) and implemented in the commercial tool ADONIS®. The ADOxx® meta-metamodel provides the basic metamodelling classes that are necessary to define graphical modelling languages such as class, attribute, and relation. It also introduces several concepts for the enterprise architecture modelling, such as model types, views, and predefined classes for directed graphs for business processes and undirected graphs for organizational structure.

The ADOxx® meta-metamodelling integrates concrete syntax and abstract syntax. The definition of the classes, attributes and relations defines the semantics and the abstract syntax of the modelling language (see left part of Fig. 3.3). The concrete syntax corresponds to the graphical notation for the modelling elements (see right part of Fig. 3.3). Each class has an attribute *GraphRep*. The value of this attribute is a script which defines the notation. Due to the expressiveness of the GraphRep script language, ADOxx® is a good fit to define metamodels for human-interpretable *enterprise architecture* modelling.

Hence for integrating enterprise architecture and enterprise ontology, the challenge is to integrate metamodels derived from frameworks like MOF – which are used for machine interpretable modelling languages – with metamodels derived from frameworks like ADOxx®—which are used for human interpretable graphical languages. Some approaches are shown in Section 4.

3.2. Modelling procedures

The modelling procedures depend on the use of the knowledge represented in the models (see Section 2.5). They support different tasks of enterprise engineering for example business process management, business-IT alignment, risk management, decision management, business analytics and supply chain management. ADOxx® is the metamodel framework of choice for the open models

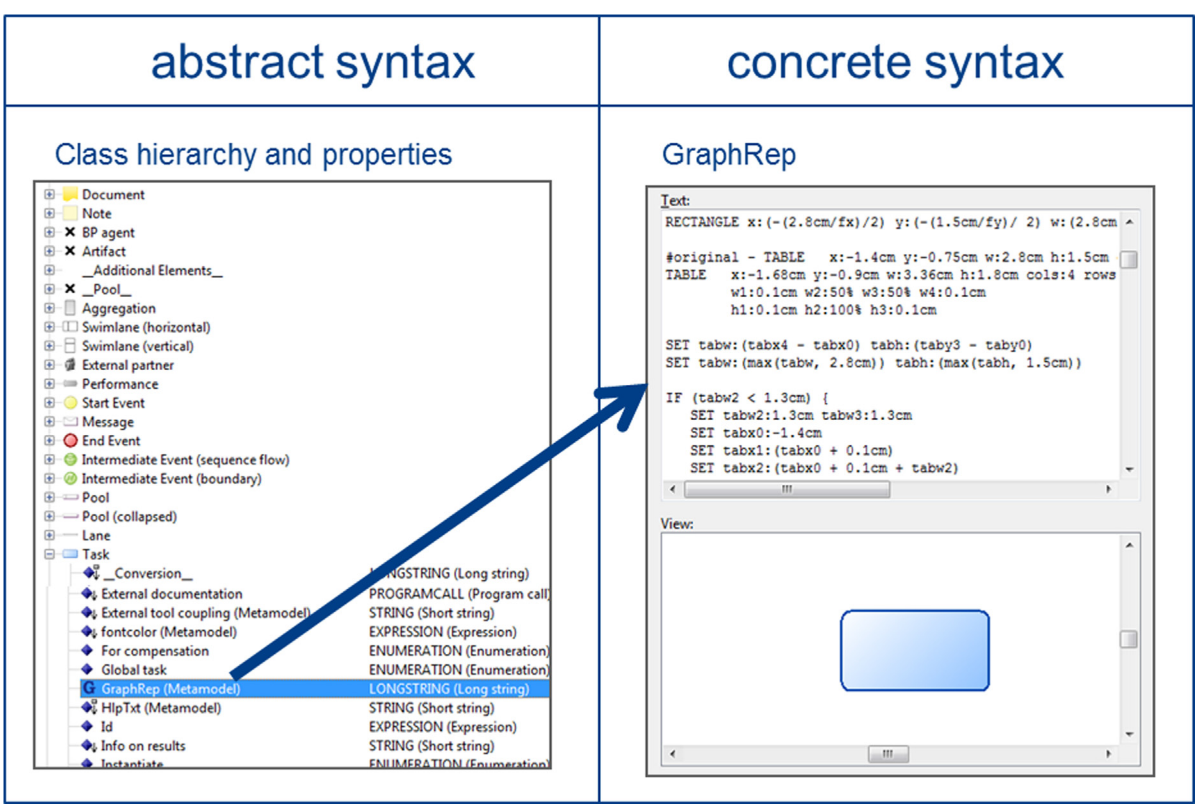


Fig. 3.3. Abstract and concrete syntax of a task in ADOxx®.

initiative (OMi, www.openmodels.at), a community of practice for the creation of modelling methods [48]. In a recent booklet of the OMi, 25 modelling methods for a variety of application domains are described [49]. In recent projects a modelling languages and procedures have been developed for strategic alignment of business and IT [50] and for integrating enterprise risk management with business motivation, business processes and business rules [51].

Since modelling is a human task, it typically starts with graphical models, which are cognitively more adequate than formal methods for most stakeholders. The graphical models are used as a means for communication between the stakeholders involved in enterprise design.

A challenge is to extend these modelling procedures with phases of machine-interpretation, which has been realized for some specialized procedures only. For business process management there are procedures that realize the execution of graphically generated process models. The idea of model-driven engineering is to transform models on higher abstraction levels into lower level models until the model can be made executable. In [37] a procedure to generate meta data from enterprise architecture model is described.

Another challenge is to keep the connection between graphical and machine-interpretable models. If for example an information system, which implements a process, is changed this change should be mirrored back to the graphical model in order to keep both models consistent.

3.3. Mechanisms and algorithms

Machine-interpretation of the models is implemented in modelling mechanisms and algorithms, which realize the model processing operations. To automate these operations, the modelling language should have a well-defined semantics and syntax. The concrete syntax is exploited for supporting the modeler in modelling design, e.g. by visualizing particular aspects of the model. Mechanisms and algorithms process the abstract syntax.

The challenges to engineer the agile enterprise demand for mechanisms and algorithms that can analyse the models in order to detect potential risks and to seize opportunities. The continuous alignment of business and IT can be supported by mechanisms, to identify information systems, which are affected by a change of the business. The other way round one might need mechanisms to identify business processes, which are affected by modifications in the IT.

4. A modelling method for continuous business-it alignment

In this section the model-based approach for enterprise engineering is presented. As already argued in the beginning, engineering the agile enterprise is an ongoing endeavor of design and redesign, which requires a continuous alignment of Business and IT. In the rest of this section the elements of the modelling method consisting of the modelling procedure, the modelling language and the mechanisms and algorithms are described (see Fig. 3.1).

4.1. Modelling procedure for continuous business-it alignment

The approach consists of four phases and is a variant of the Plan-Do-Check-Act cycle (Fig. 4.1):

(1) Establish/adjust goals: strategic and operative goals both for business and IT and their relations.
(2) (Re-)Engineer the enterprise: modelling resp. adapting the business, application and technology architectures as well as their relationships.
(3) Implement the enterprise architecture and run the enterprise.
(4) Monitor the running of the enterprise and recognize adaptation needs.

If in the monitoring phase a need for adaptation is recognized the cycle starts again. Enterprise models are particularly used for the identification of adaptation needs (Phase 4) and implementing changes (Phase 2).

4.2. Metamodelling and enterprise ontologies

In this section the modelling approach is described, which combines human-interpretable graphical enterprise architecture models and machine-interpretable formal models. The challenge is to keep both representations consistent.

A modelling language consists of notation, syntax and semantics [41] (see Fig. 3.1). Höfferer [52] discusses the relationship between metamodels and ontologies for defining modelling languages. Metamodels and ontologies are different but complementary concepts. Ontologies basically provide the semantics of the modelling language constructs [10,53] as well as the semantics of model instances. Metamodels provide the syntax of a modelling language; they define all available modelling constructs as well as valid ways to combine them. Some semantics is also included in language constructs. Therefore, there are two approaches to define the human-interpretable and the machine-interpretable modelling languages, including through Semantic lifting and Semantic Metamodels:

- *Semantic lifting:* The metamodels for the human-interpretable graphical enterprise architecture and the machine-interpretable enterprise ontology are strictly separated. Metamodels and ontologies are merged by transformation, which is called semantic lifting.
- *Semantic metamodels:* The semantics of all modelling concepts is expressed by an ontology. The ontology is extended by a metamodel, which defines the notation and syntax of the graphical modelling language. This has the advantage that the semantics is expressed only once.

In the following sections a description of these two approaches is given.

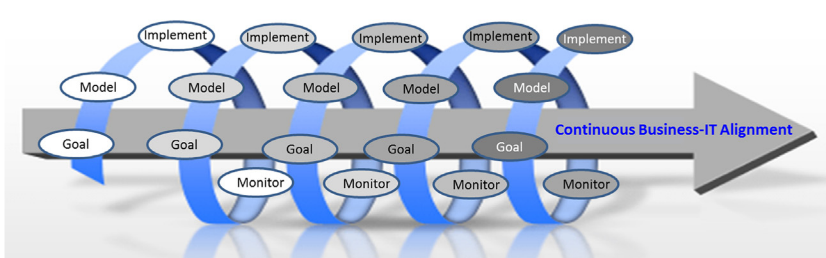


Fig. 4.1. Continuous business–IT Alignment.

4.2.1. Semantic lifting: Separating metamodels and ontologies

Fig. 4.2 shows the conceptual architecture for semantic lifting. Different models of the enterprise architecture are created corresponding to different metamodels, which define primarily syntactical but also some semantic aspects of model elements. The ontologies define the machine-interpretable semantics of the language concepts. Semantic lifting is closes related to semantic annotation, which combines the human-readable and machine-readable information (see [54] for a recent overview).

The ontologies are independent from the concepts for the graphical languages. The basis for interoperability is provided via linking model elements of the metamodels with ontology concepts. Since ontologies are, of course, also models, they need to use a language that is also defined by metamodels. Kühn has identified four kinds of merging patterns, which can be applied for integrating enterprise architecture and Enterprise Ontologies, as means to integrate different types of metamodels [55,56]:

- Reference pattern: defines links that relate exactly one element in the EA metamodel to exactly one element in the ontology metamodel.
- Extension pattern: specifies how the EA model can be extended by concepts of the EO. New concepts can be integrated.
- Transformation pattern: rules are responsible for creating parts of one or more EA framework metamodels in an ontology. This mechanism enables for example the generation of an ontology from a business process.
- Merge pattern: The merge pattern can be regarded as a specialization of the transformation pattern, where a merge rule generates a part of the ontology from two or more EA framework models.

The transformation between enterprise architecture and enterprise ontology, which makes the semantics of the graphical models explicit, is called lifting [10]. It has been implemented in ADOxx® [11].

In the project plugIT this approach was applied to enable a computer-supported business-IT alignment using semantic technologies [57]. Business people and IT providers externalise their knowledge via the use of graphical models. These models are then translated into instances of enterprise and domain ontologies to enable automated support of business and IT alignment.

The disadvantage of completely separating metamodels and ontologies is that they can have incompatible semantics. To overcome the problem, the LearnPAd project (http://www.learnpad.eu) initially agreed on a shared understanding of important terms before defining the metamodels and the ontologies. There is no way, however, to strictly enforce that the semantics of metamodels and ontologies are consistent.

4.2.2. Semantic metamodelling

In order to avoid the consistency problem between metamodels and ontologies a semantic metamodelling approach is proposed. The ontology defines the complete semantics of all the concepts. The ontology is extended by a specification of the graphical notation, which corresponds to the concrete syntax of the modelling language. The difference to the transformation approach is that the semantics is expressed only once for both human-interpretable enterprise architecture and machine-interpretable enterprise ontology. The semantic modelling can be regarded as a variant of the MOF metamodelling framework [43] where UML is replaced by an ontology language as a metamodelling language.

The graphical notation for each concept is defined separate from the semantic description (see Fig. 4.3). A mapping is defined between concept definition and graphical definition [58]. This is a difference to the approach of Fig. 1.1, where the graphical notation is part to the class definition. This semantic metamodelling approach has been prototypically implemented in the ATHENE

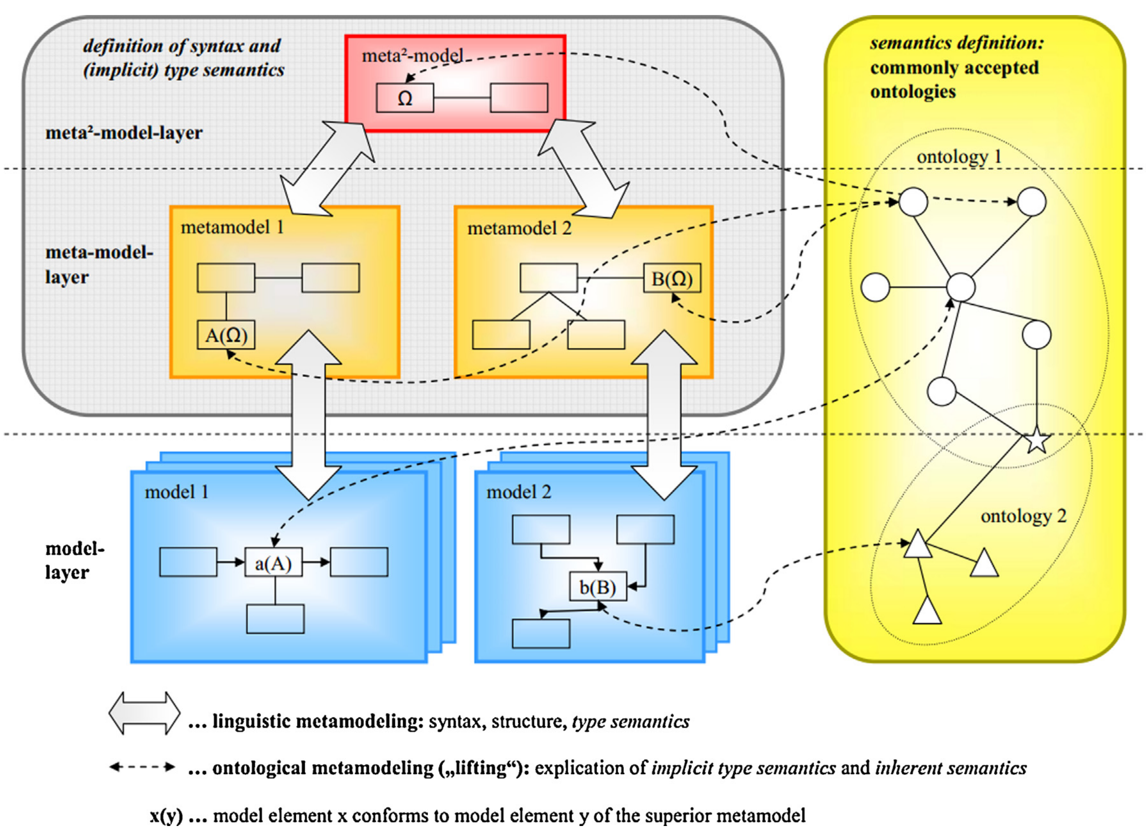


Fig. 4.2. Metamodels for human-interpretable and machine-interpretable models [52].

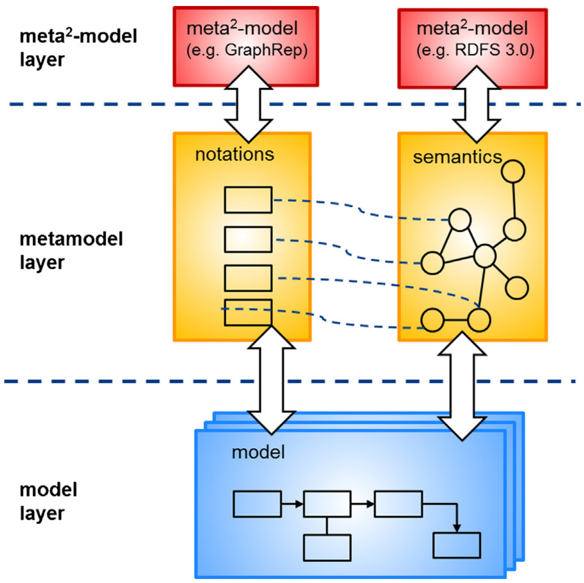


Fig. 4.3. Semantic metamodelling.

system [59]. The concepts of the modelling language are defined using the Resource Description Framework RDFS 3.0.

4.3. Mechanisms for identification of adaptation needs

Enterprise engineering should be a conscious, purposeful endeavor, and managers should regularly review their business processes and information systems from multiple perspectives to ascertain whether they are meeting enterprise needs [1]. This is typically a human task. Business analysts exploit the human-interpretable, graphical models in order to detect potential risks and to seize opportunities.

Because of the complexity of the enterprise and the enterprise models it cannot be assumed that business analysts are able to detect all required changes and are able to assess the consequences of all potential influencers. This is where machine-interpretable models and enterprise ontologies could contribute.

In a recent study it has been shown that adaptation needs of an enterprise architecture can be identified by observing content of information systems [60]. Several events have been recognized, which can be checked automatically in order to trigger adaptations of the enterprise architecture. This requires the enterprise architecture to be represented in a machine-interpretable way with a formal semantics, as it is defined by an enterprise ontology. The actual change of the enterprise architecture and of information systems still requires human judgment. Thus, a combination of machine-interpretable models to identify adaptation needs and human-interpretable graphical models to support a business analyst in making appropriate decisions on how to implement the changes offers new opportunities for continuous business-IT alignment.

A similar approach was used to improve contract management in the DokLife project [61]. Monitoring obligations and liabilities is time consuming and error prone. Whereas Contract Management Systems (CMS) deal well with time-triggered obligations like periodical payments, they fail to trigger obligations based on events, as this knowledge is out of the systems' scope. In the DokLife project an approach to fill the gap was introduced. Information about the obligations managed in a CMS is related to background knowledge modelled in an enterprise ontology, e.g. processes to be triggered, responsible roles and required resources. This allows to trigger processes based on pre-defined events.

In the APPRIS project is was shown how an enterprise ontology can be applied to analyse early warning indicators for supply risk management [36]. An inference engine regularly assesses data from various internal and external information sources in order to identify potential procurement risks. Risks depend on enterprise knowledge which is represented in the enterprise ontology. Results of risk identification and assessment are displayed in an easy-to-understand way for a human user to decide for appropriate actions. This means that the knowledge needs to be understood by both machines (for risk assessment) and humans (for deciding about actions). The combination of graphical and formal representations with a common semantics is an appropriate approach.

5. Conclusion

This paper proposed a new paradigm for next generation enterprise information systems, which shifts the development approach of model-driven engineering to *continuous adaptation* of the agile enterprise. Enterprise information systems are closely integrated with (1) model analysis tools which allow assessing influencers, to identify risks and to seize opportunities and (2) modelling tools for changing the enterprise. The proposed metamodelling approach for the implementation of these information systems supports both human-interpretable graphical models with machine-interpretable enterprise ontologies. It was shown that the integration is possible; it has been applied in several projects. It is still some time until commercial tools are available and business architects and IT architects are using this modelling paradigm.

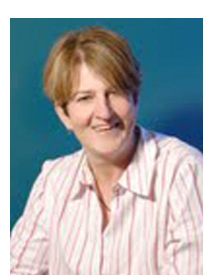
It is a future long-term challenge to involve business people not only in the adaptation of enterprise architecture but also into the implementation and adaptation of enterprise information systems. Evolving application flexibility via embedded modelling tools has been identified in a recent study as one of the 10 most important technology trends in business application architecture [62]. The authors predict that future business applications will incorporate business-oriented graphical modelling tools that enable rapid, code-free modifications to business applications, including process orchestration, business rules, notification, organizational structures, embedded business intelligence, and even the assembly of new functionality from existing functional elements. To automate the modification and adaption of applications – or at least to support the human user in adapting the current models – a formal semantics of the models is essential. The modelling approach presented in this paper provides a solid basis for this future challenge.

References

[1] R.E. Giachetti, Design of Enterprise Systems: Theory, Architecture, and Methods, CRC Press, Boca Raton, FL, 2011.
[2] R. Dove, Knowledge management, response ability, and the agile enterprise, J. Knowl. Manage. 3 (1) (1999) 18–35.
[3] R. Dove, Agile enterprise cornerstones: knowledge, values & response ability, in: Business Agility and Information Technology Diffusion, IFIP TC8 WG 8.6 International Working Conference, Atlanta, (2005), pp. 313–330, 180.
[4] F.A. Cummins, Building the Agile Enterprise with SOA, BPM and MBM, Morgan Kaufmann Publishers, Burlington, 2009.
[5] M. Op't Land, E. Proper, M. Waage, J. Cloo, C. Steghuis, Enterprise Architecture, Springer, Berlin Heidelberg, 2008.
[6] Gartner, Taming the Digital Dragon: The 2014 CIO Agenda, Gartner, Stamford, CT, 2014, pp. 1–12.
[7] M. Völter, T. Stahl, J. Bettin, A. Haase, S. Helsen, Model-Driven Software Development: Technology, Engineering, Management, John Wiley & Sons, West-Sussex, England, 2013.
[8] D. McCauley, Achieving agility, in: K.D. Swenson (Ed.), Mastering the Unpredictable, Meghan-Kiffer Press, Tampa, FL, 2010, pp. 257–275.
[9] D. Karagiannis, R. Woitsch, Knowledge engineering in business process management, in: Handbook on Business Process Management 2, Springer, Berlin Heidelberg, 2010, pp. 463–485.
[10] G. Kappel, E. Kapsammer, H. Kargl, G. Kramler, T. Reiter, W. Retschitzegger, W. Schwinger, M. Wimmer, Lifting metamodels to ontologies: a step to the semantic integration of modelling languages, in: O. Nierstrasz, J. Whittle, D. Harel, G. Reggio (Eds.), Model Driven Engineering Languages and

Systems, Proceedings of the ninth International Conference, MoDELS 2006, LNCS 4199, Springer-Verlag, Genova, Italy, 2006, pp. 528–542.
[11] V. Hrgovcic, D. Karagiannis, R. Woitsch, Conceptual modelling of the organisational aspects for distributed applications: the semantic lifting approach, in: COMPSACW 2013, 2013 IEEE 37th Annual Computer Software and Applications Conference Workshops, 2013, 145–150.
[12] M. Reichert, B. Weber, Enabling Flexibility in Process-Aware Information Systems, Springer, Berlin, Heidelberg, 2012 (no. c.).
[13] J.L.G. Dietz, Enterprise Ontology. Theory and Methodology, Springer-Verlag, Berlin Heidelberg, 2006.
[14] D. Chen, G. Doumeingts, F. Vernadat, Architectures for enterprise integration and interoperability: past, present and future, Comput. Ind. 59 (2008) 647–659.
[15] J.A. Zachman, John Zachman's Concise Definition of The Zachman Framework, Zachman International, Colorado Springs, CO, 2008, Available ⟨http://www.zachman.com/about-the-zachman-framework⟩. [Online].
[16] ISO/IEC, ISO/IEC/IEEE 42010 Systems and Software Engineering—Architecture Description, 2011.
[17] J.A. Zachman, The Zachman Framework For Enterprise Architecture: Primer for Enterprise Engineering and Manufacturing, Zachman International (electronic book), Colorado Springs, CO, 2003, Available through http://www.zachmaninternational.com.
[18] D. Matthes, Enterprise Architecture Frameworks Kompendium, Springer-Verlag, Heidelberg, 2011.
[19] J.A. Zachman, The Zachman Framework: The Official Concise Definition, Zachman International, Colorado Springs, CO, 2008, Available ⟨http://test.zachmaninternational.com/index.php/the-zachman-framework⟩. [Online].
[20] J.F. Sowa, J.A. Zachman, Extending and formalizing the framework for information systems architecture, IBM Syst. J. 31 (3) (1992).
[21] The Open Group, TOGAF 9 V91, 2011.
[22] The Open Group, ArchiMate 2.0 Specification, The Open Group, Berkshire, United Kingdom, 2012.
[23] OMG, Business Process Model and Notation (BPMN) Version 2.0, 2011.
[24] OMG, Case Management Model and Notation (CMMN), V 1.0, 2013.
[25] OMG, Business Motivation Model, Version 1.1, Object Management Group OMG, 2010.
[26] H. Franken, S. van Dijk, B. van Gils, Three Best Practices for Successful Implementation of Enterprise Architecture Using the TOGAF® Framework and the ArchiMate® Modelling Language, 2013 http://blog.opengroup.org/2013/02/08/.
[28] B. Thönssen, Automatic, Format-independent Generation of Metadata for Documents Based on Semantically Enriched Context Information, University of Camerino, Camerino, Italy, 2013.
[29] D. Kang, J. Lee, S. Choi, K. Kim, An ontology-based Enterprise Architecture, Expert Syst. Appl. 37 (2) (2010) 1456–1464.
[30] K. Hinkelmann, E. Merelli, B. Thönssen, The Role of Content and Context in Enterprise Repositories, in: Second International Workshop on Advanced Enterprise Architecture and Repositories—AER 2010, 2010 (no. Cmd).
[31] M.S. Fox, M. Barbuceanu, M. Grüninger, An organisation ontology for enterprise modelling: preliminary concepts for linking structure and behaviour, Comput. Ind. 29 (Jul (1–2)) (1996) 123–134.
[32] M. Uschold, M. King, S. Moralee, Y. Zorgios, The enterprise ontology, Knowl. Eng. Rev. 13 (1997) (Specia).
[33] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, M. Sintek, Toward a technology for organizational memories, IEEE Intell. Syst. 13 (3) (1998) 40–48.
[34] M. Uschold, M. Healy, K. Williamson, P. Clark, S. Woods, Ontology reuse and application, in: International Conference on Formal Ontology in Information Systems FOIS'98, 1998, p. 192, vol. 179.
[35] J. Den Haan, An Enterprise Ontology based approach to Model-Driven Engineering, TU Delft, Delft University of Technology, Delft, 2009.
[36] S. Emmenegger, K. Hinkelmann, E. Laurenzi, B. Thönssen, Towards a procedure for assessing supply chain risks using semantic technologies, in: A. Fred, J.L.G. Dietz, K. Liu, J. Filipe (Eds.), Knowledge Discovery, Knowledge Engineering and Knowledge Management, Springer, Berlin Heidelberg, 2013, pp. 393–409.
[37] K. Hinkelmann, M. Maise, B. Thönssen, Connecting Enterprise Architecture and Information Objects Using an Enterprise Ontology, in: IEEE International Conference on Enterprise Systems, 2013.
[38] D. Allemang, J. Hendler, Semantic Web for the Working Ontologist Effective Modelling in RDFS and OWL Second Edition, second ed., Morgan Kaufmann Publishers, Waltham, MA, 2011.
[39] R.J. Brachman, H.J. Levesque, Knowledge Representation and Reasoning, Morgan Kaufmann Publishers, San Francisco, CA, 2004.
[41] D. Karagiannis, H. Kühn, Metamodelling platforms, in: Proceedings of the Third International Conference EC-Web at DEXA 2002, 2002.
[42] S. Strahringer, Metamodellierung als Instrument des Methodenvergleichs: eine Evaluierung am Beispiel objektorientierter Analysemethoden, Shaker, Aachen, 1996.
[43] OMG, OMG Meta Object Facility (MOF) Core Specification Version 2.4.2, 2014.
[44] OMG, OMG Ontology Definition Metamodel Version 1, 2009, May.
[45] D. Karagiannis, S. Junginger, R. Strobl, Introduction to business process management systems concepts, in: B. Scholz-Reuter, E. Stickel (Eds.), Business Process Management, LNCS: Springer-Verlag, 1996.
[46] S. Junginger, H. Kühn, R. Strobl, D. Karagiannis, Ein Geschäftsprozessmanagement-Werkzeug der nächsten Generation—ADONIS: Konzeption und Anwendungen, Wirtschaftsinformatik 42 (5) (2000) 392–401.
[47] D. Karagiannis, P. Höfferer, Metamodels in action: an overview, in: ICSOFT 2006—First International Conference on Software and Data Technologies, 2006.
[48] D. Karagiannis, W. Grossmann, P. Höfferer, Open Model Initiative—A Feasibility Study, Sep., 2008.
[49] OMiLAB, Open Model Laboratory Booklet, University of Vienna, Vienna, 2014.
[50] K. Hinkelmann, A. Pasquini, Supporting Business and IT Alignment by Modelling Business and IT Strategy and its Relations to Enterprise Architecture, in: Second IEEE Conference on Enterprise Systems, 2014.
[51] N. Sudakova, K. Hinkelmann, Explicitly Modelling Relationships of Risks on Business Architecture, in: eChallenges e-2014 Conference Proceedings, 2014.
[52] P. Höfferer, Achieving Business Process Model Interoperability Using Metamodels and Ontologies, in: European Conference on Information Systems, 2007, 1620–1631.
[53] G. Kramler, G. Kappel, T. Reiter, E. Kapsammer, W. Retschitzegger, W. Schwinger, Towards a semantic infrastructure supporting model-based tool integration, in: GaMMa '06: Proceedings of the 2006 International Workshop on Global Integrated Model Management, 2006, pp. 43–46.
[54] Y. Liao, M. Lezoche, H. Panetto, N. Boudjlida, E.R. Loures, Semantic annotation for knowledge explicitation in a product lifecycle management context: a survey, Comput. Ind. 71 (2015) 24–34.
[55] H. Kühn, Methodenintegration im Business Engineering, University of Vienna, Vienna, 2004.
[56] H. Kühn, F. Bayer, S. Juninger, D. Karagiannis, Enterprise model integration, in: Proceedings of the fourth International Conference EC-Web 2003—Dexa 2003, 2003, pp. 379–392.
[57] R. Woitsch, D. Karagiannis, D. Plexousakis, K. Hinkelmann, Business and IT alignment: the IT-Socket, Elektrotechnik und Informationstechnik 126 (Jul (7–8)) (2009) 308–321.
[58] S. Nikles, S. Brander, Separating conceptual and visual aspects in meta-modelling, in: Workshop on Advanced Enterprise Architecture and Repositories, 2009.
[59] K. Hinkelmann, S. Nikles, B. Thönssen, L. von Arx, An ontology-based modelling tool for knowledge-intensive E-government services, in: MeTTeG07, Proceedings of the first International Conference on Methodologies, Technologies and Tools Enabling E-Government, 2007, pp. 43–56.
[60] M. Frei, Event and Data Sources for Enterprise Architecture Documentation, University of Applied Sciences and Arts Northwestern Switzerland, Switzerland, 2015.
[61] B. Thönssen, J. Lutz, Semantically enriched obligation management. An approach for improving the handling of obligations represented in contracts, in: Proceedings of fourth Conference on Knowledge Management and Information Sharing (KMIS2012), 2012.
[62] J. Hoppermann, P.D. Hamerman, G. Lawrie, The 10 Most Important Technology Trends in Business Application Architecture Today, Forrester Research, Cambridge, MA, Sep, 2013.

Prof. Knut Hinkelmann is dean of the Master of Science in Business Information Systems at the University of Applied Sciences and Arts Northwestern Switzerland FHNW and research associate at the University of Pretoria. At the University of Camerino he is permanent member of the PhD Committee. In 1988 he obtained a diploma in Computer Science from the University of Kaiserslautern and in 1995a PhD in Natural Sciences from the Computer Science Department of the same university. From 1990 until 1998 he was researcher and head of the Knowledge Management research group at the German Research Center for Artificial Intelligence (DFKI). From 1998 until 2000 he worked as product manager for Insiders Information Management GmbH. He joined the FHNW in August 2000 as a professor for Information Systems. From 2002 to 2008 he was dean of the Bachelor of Science in Business Information Technology. His research topics include modelling of knowledge-intensive processes, knowledge management and knowledge technologies. He has been supervisor and external examiner of many PhD theses and guest lecturer at the University of Vienna and University of Camerino. Furthermore he was CEO of the KIBG GmbH from 1996 until 1998; and from 2006 until 2012 he was Scientific Advisor of STEAG & Partner AG. He is member of IEEE and GI.



Prof Aurona Gerber completed an electronic engineering degree at the University of Pretoria in 1987. Aurona have been employed in different milieus', including a research institution, academic institutions and industry. During her employment in these different environments, she was involved in the development, research and academic side of ICT applications and information systems. She completed her PhD in 2007 in Semantic Web technologies, and is currently employed at the CSIR as principal researcher. In addition to research and development, Aurona is involved in supervision of Master and PhD students as well as teaching of post-graduate courses. Aurona Gerber serves on different programme committees for conferences, review panels for journals and is currently the chair of SMCS South Africa. Her research interest is within the domain of Ontology Engineering and Data Science, with a special interest in the use of models and meta-models to do advanced modelling.

Prof Dimitris Karagiannis read at the Technical University of Berlin, where he graduated with a PhD in Computer Science. He was a visiting scientist at research institutions in the US and Japan. From 1987 to 1992 he headed the Business Information Systems group at the Research Institute for Applied Management (FAW) in Ulm as a scientific director. Since 1993 he is full professor at the Faculty of Computer Science at the University of Vienna and head of the Research Group Knowledge Engineering. 2011 he was awarded an honorary professorship by the Babes-Bolayi University Cluj-Napoca in Romania. As head of the Knowledge Engineering group his main research areas are Business Process Management, Meta-Modelling, and Knowledge Management. Prof. Karagiannis has published several books and scientific papers in journals and conferences on Knowledge Databases, Expert Systems, Business Process Management, Workflow-Systems and Knowledge Management. In 1995 he established the Business Process Management Systems Approach (BPMS), which has been successfully implemented in several industrial and service companies. He is the founder and head of the supervisory board of the BOC Group (http://www.bocgroup.com). 2008 he was a founding-member of the Open Models Initiative and has created 2012 the Open Models Laboratory (http://www.omilab.org). In addition to his long-standing engagement in national and EU-funded research project, Prof. Karaginnis is acting since 2005 as a reviewer for the European Commission. He is a member of IEEE and ACM and serves on the steering committee of the Austrian Computer Society.

Prof Barbara Thönssen is a full professor and senior researcher with the Business Information Systems Department at the University of Applied Sciences and Arts Northwestern Switzerland (FHNW). She did her PhD at the University of Camerino in the Dipartmento di Matematica e Informatica, where her thesis focused on automatic, format-independent generation of metadata for documents based on semantically enriched context information. She started her professional work in the field of natural language processing, developing electronic dictionaries to be used for spelling checking and automatic indexing. She was leading projects in electronic archiving, document management and workflow management for a number of large Swiss banks. She was responsible for E-Government and electronic archiving solutions for the Zurich City Council. In 2004 she joined FHNW where she lectures on Business Information Systems and is responsible for their Certificate of Advanced Studies for Information and Records Management. She is currently engaged in several national and international research projects. Her current research focuses on bringing semantic technologies into practice.

Prof Alta van der Merwe is currently the Head of Department and associate professor in the Informatics Department within the School of Information Technology. Informatics is a diverse and young field where much of the research done is still using grounded theory to establish frameworks and methods to understand the field and to investigate the complexity of the use of technology in different domains. Informatics is also the field where technology and people 'meets'. In this unique field Prof Alta van der Merwe focuses on the design of socio-technical solutions with research activities in Enterprise Architecture, Data Science and different theories supporting the successful use of technology in the organization. Her interest also includes the design of systems using innovative and new approaches such as Crowd Sourcing and Content Awareness. Prof van der Merwe is the founder and past chair of the South African IEEE SMCS Chapter, specialist editor of the SAIEE journal (Software Engineering track) and co-founder and past chair of the Enterprise Architecture Research Forum (EARF). On International level she was involved in the proposal and acceptance of the IEEE Enterprise Engineering and Enterprise Architecture Technical Committee, where she still acts as co-chair.

Dr. Robert Woitsch holds a PhD in business informatics and is currently responsible as managing director for innovation management via European and National research projects at the consulting company BOC (www.boc-group.com) in Vienna. He deals with concept modelling and knowledge management projects since 2000 starting with the EU-funded project PROMOTE and has been working in more than twenty EU-projects in the domain of technology enhanced learning, knowledge management, business and IT alignment as well as business process management. Beside his participation in those EU projects, he coordinated the business and IT alignment project plugIT and is now coordinating the cloud project CloudSocket. Dr. Woitsch is involved in commercial KM projects for skill management and knowledge balances especially in the security domain and was a member of the Austrian Standardization Institute. He published about 40 papers and is involved as a reviewer for EU funded projects and acts as a member of program committees in KM-conferences. He recently manages the Open Innovation Community adoxx.org.