Business-oriented CAx Integration with Semantic Technologies Revisited

Claudia Hess[1], Willy Chen[1], Thomas Syldatke[2]
[1]SysTec-CAx GmbH, München, {claudia.hess,willy.chen}@systec-cax.de
[2]AUDI AG, Ingolstadt, thomas.syldatke@audi.de

Abstract: Product development is one of the core business processes for car manufacturers. Today it is heavily support by information systems enabling computer-aided technologies for design (CAD), engineering (CAE), and testing (CAT), which we refer to as CAx systems. In order to improve the alignment between business and IT we have created an enterprise architecture for the research and development department at AUDI AG. The description of the various relevant architecture domains is thereby based on ontologies. A web-based tool enables collaborative editing of these contents as well as complex analysis on various aspects relevant to IT management.

1 Introduction

Core challenges for today's automotive industry are manifold. Market requirements are constantly changing, e. g. due to new regulations or customer demands. The product complexity increases, especially due to an increasing number of electronics. Moreover, the model range grows – 50% more models can be expected in the next years – while the development cycle time is reduced. Research and development departments (R&D) of automotive manufacturers nowadays have a number of specialized IT applications in use. They implement especially the different computer-aided methodologies, such as design (CAD), engineering (CAE) and testing (CAT). These specialized systems are highly dependent and constantly evolve in order to better adapt to the changing business requirements. We have created an enterprise architecture for the R&D department at AUDI AG to improve the alignment of business and IT and to manage the complex dependencies within the heterogeneous IT landscape.

At AST 2008 [CSH08], we presented our intended approach for integrating CAx methodologies and IT systems from a business-oriented view. We aimed to use semantic technologies to represent, formalize, and interconnect the relevant knowledge. Implementing this approach, we focused on supporting the development of the CAx systems with a business focus. This paper shows how we built up an enterprise architecture using semantic technologies. To this end, it is organized as follows. Section 2 introduces the use case. Section 3 describes the ontologies that we developed. In section 4, we present our approach to ontology-aided IT management and the web-based application that is used to access the knowledge and which allows for easily updating the ontology in a collaborative manner. Finally, section 5 summarizes the approach and lessons learned.

2 Use Case

Computer-aided methodologies support the different steps in the product development process. They are especially used for the construction of new parts and prototypes, their validation and their release for the serial production. Using as much as possible virtual validation methodologies and combining them in an appropriate way with physical validation methodologies allows for optimizing the product development process. Computer-aided design (CAD), engineering (CAE) and testing (CAT), referred to as CAx, play thus a major role. They are supported by a number of specialized IT systems ranging from standard products, such as in CAE the tools for pre-processing, solving and post-processing, to individual software that is highly tailored to AUDI's needs. As tools may require the output of other tools as input, there are complex dependencies between the different information systems. The exchange of information between them is not straightforward because product models, data structures and technical platforms differ. Managing the dependencies and bridging the gaps between CAx systems is thus a major task for optimizing their interplay and, in the end, that between the CAx methodologies.

As business architects, we analyze project proposals for new CAx systems as well as for extensions of existing ones. We have to ensure their integration into the IT landscape and their fit with business requirements. Moreover, managing the IT landscape requires to identify issues in the existing infrastructure and to initiate change projects. These may improve the alignment of business and IT or increase the re-usability and flexibility. Methods and frameworks from Enterprise Architecture (EA) provide here a valuable starting point. An EA describes the basic artifacts of an enterprise and their dependencies. These artifacts range from business aspects, such as strategic aspects and organizational structures, to IT-related ones, such as software components and the IT infrastructure. The EA allows for the analysis of the enterprise's current IT infrastructure and the strategic planning of its future state. The Enterprise Architecture Management Method Library defines Enterprise Architecture Management (EAM) as "a continuous and self maintaining management function seeking to improve the alignment of business and IT in an (virtual) enterprise" [EAM].

Different frameworks such as the Zachman Framework [Zac87], TOGAF [The09] and the ARIS framework [Sch99] have been proposed for modeling an EA. Creating an EA for the R&D department at AUDI AG, we followed the methodology suggested by TOGAF. TOGAF, The Open Group Enterprise Architecture Framework, is an industry standard architecture framework that can be freely used to develop an EA. The core of TOGAF is its generic enterprise architecture method, the Architecture Development Method (ADM), which specifies the different steps required for building up an EA. Guidelines and frameworks complete the toolkit. TOGAF comprises the following enterprise architecture domains: The *business architecture* comprises the business strategy, governance, organization, and key business processes. Then, the *application architecture* provides a blueprint for the individual application systems. It contains the interactions between the systems and their relationships to business processes. Then, the *data architecture* structures the organization's logical and physical data assets and the associated data management resources. The *technical architecture* describes the hardware, software and network infrastructure.

3 Architecture Domain Ontologies

Following the TOGAF architecture methodology, we started developing ontologies for the four architecture domains. In a whole, they provide the essential information for the management of the different CAx systems in use. The ontologies for the application and the data architecture domain are presented in the following (see also figure 1). We show the main classes for each domain. In practice, they are refined by subclasses, attributes and relationships.

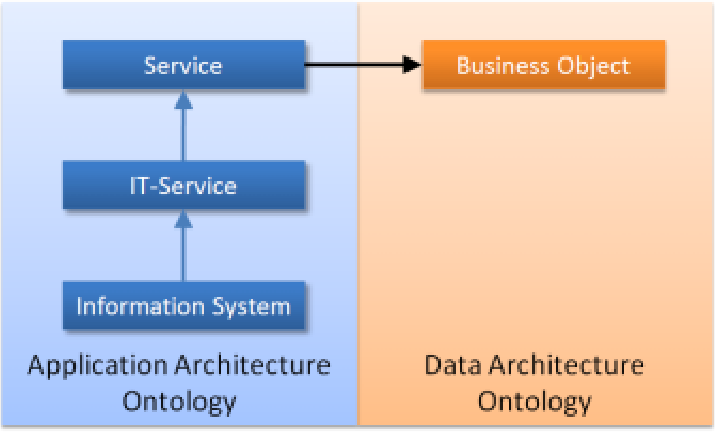

Figure 1: Top Level Classes

3.1 Application Architecture Ontology

The application ontology defines the structure for describing the software systems that are in use in AUDI's R&D department in a business-oriented way. Moreover, it provides a starting point for getting more detailed information on the actual implementations. Following the ideas of a service oriented architecture (SOA), an information system is described in terms of atomic IT-services and business services. Services are independent of other services and are highly reusable. They can be composed to more complex services. A service is self-contained and does not expose any internals to the service consumers.

The Application Architecture Ontology was inspired by the Service-Oriented Architecture Ontology [The08] (SOA Ontology) developed by The Open Group. This ontology aims at describing a service-oriented architecture with the help of an ontology. The SOA Ontology is currently available as a draft technical standard. It is provided as a OWL ontology. It defines classes for describing a SOA and relationships between these classes. The SOA Ontology contains both business and technical aspects. Moreover, it defines classes related to the development and management of a SOA.

Central classes and relationships of our ontology are:

- Business Service: A business service is, following the definition in the SOA ontology "a logical representation of a repeatable business activity that has a specified outcome". It is described from the domain user's point of view. Business services are on the one hand data services for creating and accessing business objects (e.g. create order, modify order) and on the other hand logic services which realize more complex business logics and rules by using one or more distinct business objects (e.g. check availability).
- IT Service: An IT service represents a self-contained functionality that corresponds to a business service. An interface describes its semantics. An IT service is ideally a Web Service. It may also be an EJB. However, in order to be able to represent the functionality by legacy systems, other IT service types may be considered such as DB views or Stored Procedures. We described IT services in a way that potential users of this IT service have enough information to use it. For example, a Web Service has to provide a link to its WSDL. Dependencies between different IT services are captured in the application architecture ontology, too: an IT service uses another IT service.
- Information system: An information system provides IT services. Systems thereby range from individual software to standard software and customized standard software. Standard software is ready-made software and is used as it is by the engineers in AUDI's R&D department. Individual software, in contrast, is specially developed to meet particular requirements. Customized standard software is such standard software that is tailored to fit AUDI's needs.

3.2 Data Architecture Ontology

The central class of our data architecture ontology is the business object. A business object is a description of a business entity that is independent of its implementation. It is important to define a business object precisely in order to ensure a common understanding. A business object can be described in detail by its attributes. For example, we have a business object *order* with attributes like an *order number*. Business objects may be structured hierarchically with general ones on the highest level, and more specific ones on the detailed levels. For instance, a business object *resource* may be further differentiated in resources required for performing a certain validation, such as a *crash test dummy* for performing a crash test. Business objects inherit the attributes of the hierarchically higher ones. For example, a specialized order for building up a prototype, also has an order number, inherited from the more general business object *order*. A further possibility to structure the data architecture are references between business objects. That means that dependencies between business objects are explicitly represented. An example for such a reference is that a client *creates* an order.

4 Ontology-aided IT-Management

In the previous section we presented ontologies which describe the application and the data architecture domain. These descriptions are adopted to the needs of AUDI and are used within different project stages that are relevant for governance in service-oriented architectures (cf. [Bis08]):

- Pre-project stage: During the initial project proposal and funding, the project initiator and an enterprise architect utilize the enterprise architecture to clearly define the scope of the envisioned project in terms of business objects and services that should be implemented as well as IT services to be reused. If the project deals with new business objects or requires more specialized services, the enterprise architect adds these information to the appropriate domain ontology.
- Project stage: The result of the project stage is, among others, a standardized IT specification based on UML 2.0, which covers all relevant aspects for the application architecture domain. Based on the XML metadata interchange (XMI) format, these information are synchronized with the application architecture ontology to reflect the most recent information about a specific IT system.
- Run-time: Enterprise architects are constantly monitoring the different architecture domains in order to identify optimization needs and if necessary trigger change projects. IT architects can analyze dependencies, e. g. which services and systems are affected by a modification of a certain service or have to be informed before the release of a new version.

Users at AUDI, such as enterprise and IT architects, project and requirement managers, can access the information from the application and data architecture ontology via a web-based application that provides means to search and visualize information but also edit and correct them like in a WIKI system. While the core information are represented as ontology facts, users can link documents to any of these ontology individuals to provide additional information about a topic. A semantic search engine enables the search for terms not only in the ontologies but also in all documents that are linked to ontology elements. A ranked result list is presented to the user. Users can generate reports for any element in the ontologies and use a graph visualization to analyze all known dependencies to other elements in the information model.

Most importantly, users can edit any A-Box element in a collaborative way using a wiki-style editor including features for showing the change history and comparing specific versions of an individual as well as simple rating mechanisms. The editor does not require any knowledge on ontological modeling so that business users can easily contribute information and add comments on existing facts. The described web-based tool has been set productive early 2009. Currently, the application has about 80 registered users, of which 25 are actively using and contributing to the knowledge base. The T-Box of all ontologies contains 80 classes, 104 object properties, and 22 data properties while the A-Box has about 25000 individuals and assertions.

5 Conclusion and Outlook

In this paper we have presented an ontology-aided approach to support the IT management in the research and development department at AUDI. The implementation of these ontologies is based on technologies that are currently being developed for realizing the vision of a Semantic Web. We use the Web Ontology Language (OWL) as ontology modeling language and provide a web-based application where users can edit the assertional boxes of the used ontologies without any detailed knowledge about ontology modeling. Most importantly, the use and update of the presented models is embedded into the IT management process, which ensures that the information are up to date and actively used. This is crucial for a high information quality.

While we are convinced that the architecture domain ontologies have to be defined for each enterprise separately, a common application architecture ontology e.g. based on the SOA Ontology by The Open Group could be an ideal starting point for enabling ontology-aided IT-management in other industrial environments. Depending on the available information about information systems, however, the degree of detailing needs to be adaptable.

We are currently working on addressing additional aspects of enterprise architecture management such as IT planning or business process alignment. Also, we are starting to implement IT governance guidelines using logical rules as provided by the Semantic Web technology stack.

References

[Bis08] Todd Biske. *SOA Governance*. Packt Publishing, 2008.

[CSH08] Willy Chen, Thomas Syldatke, and Claudia Hess. Business-oriented CAx-Integration with Semantic Technologies. In *Proceedings of Informatik 2008, Applications of Semantic Technologies 2008 (3rd International AST Workshop)*, 2008.

[EAM] EAMML. Enterprise Architecture Management Method Library (version 0.2). `http://wwwmatthes.in.tum.de/wikis/sebis/eamml`, last accessed January 19, 2010.

[Sch99] August Wilhelm Scheer. *ARIS – Business Process Frameworks*. Springer, 3 edition, 1999.

[The08] The Open Group. SOA Ontology Draft 2.0. `http://www.opengroup.org/projects/soa-ontology/uploads/40/16940/soa-ontology-200-draft.pdf`, last accessed January 19, 2010, 2008.

[The09] The Open Group. *TOGAF Version 9, Enterprise Edition*. 2009.

[Zac87] J.A. Zachman. A framework for information systems architecture. *IBM SYSTEMS JOURNAL*, 26(3), 1987.