

# Business-oriented CAx Integration with Semantic Technologies

Willy Chen<sup>1</sup>, Thomas Syldatke<sup>2</sup>, Claudia Hess<sup>1</sup>

<sup>1</sup>Achievo Inproware GmbH, München, {willy.chen,claudia.hess}@achievo.de

<sup>2</sup>AUDI AG, Ingolstadt, thomas.syldatke@audi.de

## 1 Introduction

Increasing the efficiency and effectiveness of the product development process is an important strategic task for the AUDI AG to deal with the challenges of today's automotive industry. The flexible but fully integrated use of virtual and physical validation methodologies opens up possibilities for optimizing the development process of new parts or prototypes. These methods are today strongly supported by different Computer-aided technologies (summarized as CAx), which are implemented by a number of specialized IT systems. This work presents an approach to support the management in integrating these different tools, especially from a more business-driven view. We use semantic technologies to represent, formalize, and interconnect the business and IT knowledge relevant to CAx supported validation methods. To this aim, the paper is organized as follows. Section 2 introduces the use case. A CAx architecture and its technical implementation are presented in sections 3 and 4. We conclude with a discussion and an outlook on future activities.

## 2 Use Case

The development process of a car is driven by a huge number of different requirements, either from the market, from the car manufacturer (e.g., to reflect the corporate identity), or from regulations. Within the product planning and concept development phase at AUDI, the specific requirements for a new car model are translated to technical specifications. In the successive process steps, the actual parts are then constructed, validated, and released. All these steps, in particular the validation, are strongly supported by CAx methodologies, namely:

- Computer Aided Design (CAD): This branch provides methodologies for the virtual design and validation of the parts of a car, such as digital mockup (DMU) methods or parametric design methods.
- Computer Aided Engineering (CAE): The main task of CAE is to provide method-

ologies for simulating the behavior of a car with respect to its functions and features - e.g., finite element analysis (FEA) for crash simulation, computational fluid dynamics for thermal management, and multi body simulation (MBS) for driving dynamics.

- Computer Aided Testing (CAT): Methodologies for performing physical tests of cars are provided by CAT. Examples are the vehicle management, job, and testing control, and test result analysis.

All CAx methods are implemented by appropriate information systems. Their individual focuses within the product development process have led to a great variety of add-on tools that are in use today. As each of them is highly specialized, product models, data structures and technical platforms differ across the CAx tools. This heterogeneous IT infrastructure complicates the exchange of information between different CAx systems which, however, is a prerequisite to flexibly choose the optimal validation approach. Moreover, the actual choice has to take into account the business aspects ranging back to the initial requirements.

An example illustrates the interplay of the CAx methodologies: A requirement regarding passenger safety should be validated and the project manager has to identify the relevant CAx methods. Crash tests – either virtual simulations or physical tests – therefore need to be carried out. While in early phases, a physical crash test might not be feasible because real parts are still under development, virtual methods are nowadays not yet sufficient for simulating all physical aspects. Time and cost efforts are another aspect. Virtual models can be reused, whereas physical ones fit only for a knacker's yard after a crash test. However, in order to prove compliance with regulations, physical tests are still indispensable today. Having chosen the appropriate combination of virtual and physical crash tests, the results obtained have to be jointly analyzed. This, however, is not straightforward, because the validation methods are supported by different CAx systems. Moreover, some methods might still be missing and have to be developed.

Finding the optimal combination for a validation need is thus a highly complex task, where we have to deal with business knowledge, IT systems, and in particular, with all the interdependencies between them. Automated means might help to better deal with this challenge. Decisions could be made more transparent and traceable. Key questions for such intelligent systems include:

- Which CAx methods are the most efficient ones in a particular process context to validate specific functions of the car?
- Which combination of CAx methods is most suitable for carrying out a specific validation task with respect to time and budget?
- Which CAx systems are implementing the required CAx methods?
- How can one exchange information between specific CAx systems?

### 3 Solution Approach

Addressing the given use case raises two main questions. First, how can we represent and integrate knowledge from different domains uniformly and make it machine-processable? Secondly, how can we make the dependencies between the various aspects of CAx integration transparent and traceable? Semantic technologies, which were originally developed for knowledge representation and reasoning in the Semantic Web, provide promising means to deal with exactly these issues. Ontologies can be used to formalize different kinds of knowledge in a uniform and interoperable manner. Complex relations and policies can be captured as rules. Reasoners finally allow for an automated discovery of implicit dependencies in such formal models.

Since the problem domain is highly complex and hardly fits into a single model, we have developed a layered architecture. Its divide and conquer approach makes the overall complexity of the envisioned knowledge base manageable and facilitates its development. The layers comprise the core business aspects as described in the use case. Ontologies represent the information within the layers as well as the dependencies between subsequent layers. Figure 1 depicts this approach.



Figure 1: CAx Architecture

The layers represent the following business aspects:

- **Requirements:** The product development starts with defining the requirements for a product.
- **Functions and Features:** The requirements for a car are translated to an appropriate set of functions (e.g., braking, wiping) and characteristics – i.e. concrete requirements for specific functions (e.g., braking within a specific distance, compliance with regulation FMVSS 104<sup>1</sup>). These functions and characteristics can be uniformly

<sup>1</sup>U.S. Department of Transportation, Federal Motor Vehicle Safety Standards and Regulations, Standard No. 104 - Windshield Wiping and Washing Systems - Passenger Cars, Multipurpose Passenger Vehicles, Trucks, and Buses

defined for all car models.

- **Processes:** As mentioned before, the state within the development process is a important aspect for choosing a specific validation method. We therefore need an adequate representation of this knowledge.
- **Validation Methodologies:** The core task of the reference architecture is represented by this layer, covering detailed information about the various validation methods as well as qualitative and quantitative criteria for supporting the management decisions.
- **Services and Systems:** Following the idea of the Service Oriented Architecture (SOA), this layer covers the actual CAx systems and their functionality described as atomic services.
- **Business Objects:** In order to provide the required information for exchanging data, an implementation independent representation of the data is given. If required, rules may capture the semantic relationship between different data sources.
- **Data:** This layer depicts the data that is already available in the CAx systems. Typically, it resides in relational database systems such as Oracle RDMBS.

The requirements as well as the actual data are used as cornerstones of our architecture. Describing the dependencies between the layers – for example representing the knowledge, which validation method is implemented in which system – completes the knowledge base for CAx integration. We may thereby reuse first results from the ongoing research in the areas of Semantic Business Process Management, Semantic Web Services and Semantic Data Integration for the creation of specific layers of our architecture. By applying reasoners, we are able to discover dependencies across multiple layers and make them visible to the business expert.

The CAx integration addresses an important business aspect within AUDI AG that could affect the productive processes and decisions. We therefore need to ensure high-quality results when creating the CAx architecture. Due to the great amount of relevant knowledge and limited development capacities, we need methodologies for supporting the efficient development and constant maintenance of the created CAx architecture. The reuse of existing approaches to deal with such aspects is a key to accomplish this task today. The use of standards guarantees the reusability and a long service life of the developed knowledge base.

## **4 Implementation**

The Web Ontology Language (OWL) and the Rule Interchange Format (RIF) are considered core technologies for the Semantic Web today (cf. <http://www.w3.org/2001/sw/layerCake.svg>). While OWL is already widely accepted as a standard for ontologies, the work on RIF is still in progress. The lack of an accepted rule standard is reflected by the number of existing proposals and languages such as SWRL, WRL, and F-Logic.

Also, reasoning with large data sets (ABox reasoning) is still a big issue for the practical use of OWL today.

To deal with these challenges we decided to use lightweight ontologies and rules. They are based on subsets of OWL, namely OWL Lite<sup>-</sup> [dBPF04], and the latest Basic Logic Dialect proposal from the Rule Interchange Format (RIF) working group at the W3C. The proposed combination of ontologies and rules falls into plain Datalog, for which sound, complete, and efficient evaluation strategies are well known. This guarantees for a high-performance and scalable foundation today while remaining open and flexible to future developments.

In order to decouple the development and the maintenance of the knowledge base from the actual representation language, we adopt the Model-driven architecture (MDA) proposed by the Object Management Group (OMG). It provides means to create, maintain, and exchange formal models and is easily applicable to ontologies and rules. An important rationale is hereby the possibility to reuse experiences from MDA with respect to the management of models and the availability of frameworks supporting it – e.g., Unified Modeling Language (UML) for model development and XML Metadata Interchange (XMI) for model exchange. Moreover, the OMG has recently adopted the Ontology Definition Metamodel (ODM) as a standard for ontology development.

Based on an integrated metamodel that reflects the used combination of ontologies and rules we have build a management framework using the Eclipse platform and the Eclipse Modeling Framework (EMF) [CS08]. It provides a uniform foundation, for example, for verifying and validating the quality of the knowledge base. It allows us to support a number of existing representation languages and reasoners such as OWL and SWRL using KAON2<sup>2</sup>, Datalog using IRIS<sup>3</sup>, and F-Logic using Ontobroker<sup>4</sup>. This flexibility guarantees a high reusability and interoperability of the developed knowledge base. Also, we can ensure a high-quality knowledge base by testing it on different inferencing platforms.

While a number of technical issues such as quality and security assurance or expressiveness and reasoning scalability still remain fundamental for the future success of the intended solution, the most challenging problem is the question how to acquire the relevant knowledge. While there are a huge number of different documents available for all aspects of the product development process, the most important knowledge is often hidden in the experts' heads. It is therefore essential to offer those people appropriate interfaces to voluntarily contribute their knowledge to the knowledge base. We are convinced that approaches in the context of Web 2.0 [O'R05] are promising ways to deal with this issue.

Until today we have done an initial creation of the presented architecture. We extracted a number of relevant information from existing sources such as enterprise information systems or Excel files. We started interviews to collect expert knowledge, especially on the dependencies between the layers. Currently, we focus on a small number of functions of a car.

---

<sup>2</sup><http://kaon2.semanticweb.org/>

<sup>3</sup><http://iris-reasoner.org/>

<sup>4</sup><http://www.ontoprise.de>

## 5 Discussion and Outlook

In this paper we presented an approach for connecting various information sources to provide a knowledge base that enables a more business-oriented integration of CAx systems at AUDI. In particular, we introduced a layered architecture, which could be used to analyze and discover dependencies between originally defined product requirements and their representation in specific IT systems. This boosts the possibilities for selecting the most feasible validation methods – e.g. which combination of virtual and physical crash tests – to use in order to release some new part.

The application of semantic technologies – i.e. ontologies and rules – thereby allows for a uniform representation of the relevant knowledge. Furthermore, there is no need to change or modify any existing system or data source. In fact, we make them interoperable by adding semantic layers. We opted for using standards, or at least highly promising proposals, such as provided by the W3C. The adoption of the Model-driven Architecture enables an industrial strength knowledge base management relying on a number of existing solutions. This guarantees for a future safe investment on the developed knowledge base and its high quality.

The main advantage of the presented approach based on semantic technologies is the ability to deal with complex and heterogeneous knowledge sources and integrate them in a uniform manner. This allows us to discover interdependencies within and across the various aspects of the product development process and reveal them to the business user. To achieve this increased transparency we can rely on machine-based reasoning. This approach provides an important means to deal with the steadily increasing complexity of the automotive development.

Based on the presented CAx architecture, we can further use formal methods to assess and evaluate the quality of the existing CAx systems and their use within the product development process. This might also be useful for discovering potential improvements. Also, the CAx architecture contains information relevant for the requirements analysis and the system design phase of new applications in the context of the product development process. In particular, we can provide information about the available services, their interfaces, and their dependencies. Altogether, the CAx architecture provides a platform enabling a highly business-oriented governance of CAx systems.

## References

- [CS08] Willy Chen and Heiner Stuckenschmidt. Towards Industrial Strength Knowledge Bases for Product Lifecycle Management. In *16th European Conference on Information Systems, Galway, Ireland, 2008*.
- [dBPF04] Jos de Bruijn, Axel Polleres, and Dieter Fensel. OWL Lite<sup>-</sup>. WSML Deliverable D20v0.1, 2004.
- [O’R05] Tim O’Reilly. What Is Web 2.0. <http://oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>, September 2005.