Integration and Knowledge Management Platform for Concurrent Engineering

Axel Hahn

University of Oldenburg, Escherweg 2, 26121 Oldenburg, Germany, hahn@wi-ol.de

Abstract
In this paper an integration approach is introduced that supports the interrelation of the distributed product model across tools boundaries using knowledge representation concepts like the Resource Description Framework (RDF). The product model stays “inside” the models of the used tools. This reduces the integration costs. The interrelated product model defines a network representing knowledge about the product/system that is under development. This network can be interpreted as a semantic network. With this approach product data can be linked to knowledge management tools, which are also based on a semantic network knowledge representation. A system architecture for cooperative development of the knowledge oriented product model is introduced combining Product Lifecycle and Knowledge Management technologies.

Keywords
Collaborative Workspaces, Information Management, Knowledge Management, Product Data Management

1 Introduction

Modern products become more and more complex. The management of the complexity of today’s products and their design processes has risen as a major challenge in product and system development. Following a suitable design methodology (e.g. [Pahl, Beitz, 1996]) engineering knowledge is used to model the product. Detailed knowledge about the actual development is required to handle the product model complexity, to reuse design modules and parts and to ensure the product quality. This knowledge is reused during the entire product lifecycle.

Products are developed in interdisciplinary development teams. One key to an effective cooperative product development process is the efficient exchange of product data and knowledge between the developers. Product Lifecycle Management (PLM) and Knowledge Management (KM) systems were developed to provide the right information and knowledge at the engineering workplace at the right time. PLM Systems help the engineer to handle the product and process complexity during the product lifecycle. KM systems maintain explicit knowledge reused for the product development. Both kinds of systems provide collaboration services (e.g. electronic whiteboards, news forums, etc.) to support the non-formal information and knowledge exchange. Most PLM and KM systems follow a coarse-grained approach by using a file-based information management. Information in the product models or inside documents describing some knowledge cannot be interrelated and accessed on point.

In product or system development processes a large number of Computer Aided Engineering (CAE) tools are used. This increases the complexity of the development process as well. The digital model of the whole product is distributed in the different model schemes and data storage facilities of the tools used. To avoid high integration cost a loosely coupled integration approach is introduced in this paper. This paper introduces an integration approach that allows the fine-grained interrelation of product models from the different tools with low integration efforts. By using standardized knowledge representation technologies
like RDF the captured knowledge is reusable by agents, for reasoning, for search and navigation etc.
In the next chapter a short wrap-up about KM and PLM systems is given extended by an overview about integration technologies, which are used in the engineering domain. Chapter 3 introduces the fine-grained integration approach and how Semantic Web technologies are used for the implementation within a platform architecture depicted in chapter 4.

2 Knowledge and Product Lifecycle Management

KM and PLM systems are important tools to manage the complexity of the product models and development processes.

2.1 Knowledge Management

Knowledge has become an important production factor for modern industrial companies. In virtually all business processes the accessibility of knowledge leads to an improvement in time and quality of the process results. This is especially relevant in the product development / system engineering domain. Knowledge management systems should provide the necessary information at the engineer’s workplace.

In most cases this knowledge base consists of documents (MS-Office, PDF, HTML, …) administrated in a classification hierarchy and searchable by keywords or full text retrieval. Because the knowledge representation level is on the granularity of files a fine-grained access to dedicated information (e.g. a formula, a diagram or a part entity) and their interrelation is difficult. Often knowledge management systems are poorly integrated with the tools which are used in the product development process e.g. CAE Tools, PLM Systems etc. Most commercial Knowledge Management systems are generic and not specified for the product development domain. Recent developments in the knowledge management domain can be found in [Damiani, Jain, Howlett 2002].

2.2 Product Life Cycle Management Systems

Product Life Cycle Management systems maintain all product information along the product life cycle. These company wide used systems manage the different structures of the product (Bill of material, sell structure, etc.). They offer basic information management capabilities like file, user, process and configuration management combined with viewers, product structure editors and many other functionality. A good overview about status and future developments of PLM systems is given by [Abramovici, Sieg 2002].

These tools maintain data on file level. The PLM system has normally no access to the information stored in the files. Therefore the PLM system cannot react on dedicated changes inside the managed files and therefore cannot guarantee the consistency between the files. Extending the PLM System in order to access the file content would have the disadvantage of high integration efforts.

2.3 Integration Technologies

To give an overview about tool integration a three-layer model is helpful: Layer one provides the integration on communication level. Layer two covers the integration on the semantic level. On the third layer processes and methodologies are integrated.

On communication level several integration concepts are developed (e.g. file exchange, database integration, inter-process communication). Inter-process communication (Web Services [W3C 2002], RPC, CORBA [Mowbray, Zahavi 1995]) allows cooperative usage of the tools in concurrent engineering.
Each CAE tool uses its individual model schema. The most common and well-used integration approach on the model level is the development of a common schema standard for the information exchange (e.g. STEP [ISO 1994]). This reduces the number of needed import / export filters. Because of the richness of the exchanged product information standards like STEP become very complex. Due to the variety and dynamic evolution of the tools, which are used in the engineering domain, standards are not able to cover all aspects of the model that can be handled by the tools.

On the process level the concepts of the different engineering disciplines have to be supported by modern workflow management components e.g. provided by Product Life Cycle Management tools. Knowledge engineering is not yet fully covered by existing development methodologies.

3 Knowledge based Product Model Integration

As already addressed in modern development processes the management of product models is often file-based and therefore coarse-grained. The integration of product information into a single representation in an information registry e.g. using STEP causes huge integration efforts and is not suitable for quick and flexible tool integration.

In this research approach the product model stays under the control of the dedicated tool using the tool specific data model schema. This significantly reduces the integration efforts. Semantic web technologies and management tools are used to link the model entities of this distributed model. Two integration concepts are used: file-based and online. In the file-based approach the information are stored in a XML representation enriched with meta-information expressed in RDF (Resource Description Framework, http://www.w3c.org/RDF). In the other approach tools can provide the information online with an interface implemented as a Web Service.

The concept introduced in this paper supports the fine-grained integration of CAE-Tools with PLM and Knowledge Management Systems to offer best access to product information and knowledge at the engineering workplace.

3.1 Fine-grained Model Integration

The proposed loose tool integration approach manages the interrelations of the different product model entities distributed over the used CAE Tools described by using the data models of these tools. The engineer can use this to store a lot of useful information which help to understand the product model, describe constraints and other dependencies and help to manage the product model consistency. Obviously this is not a suitable approach for scenarios like the exchange of the product model between a CAD and a Finite Element Analysis.

The fine-grained integration approach is very helpful to support concurrent engineering. The stored interrelation information helps to understand the work done earlier or by others and to manage the product model consistency and the development process (e.g. by identifying modules that have to be adapted to changes done on related modules).

The linkage of product model entities by references loosely integrates the CAE tools. This is a very suitable tool integration approach for co-operation in current engineering. It has strengths in the area of multidisciplinary engineering and integration of the tools along the engineering process phases. This approach leads to an integrated and distributed product model.

In the following the way of working with this integrated product model is described by an example from the system-engineering domain: One of the first activities in the engineering process is the identification and the specification of the requirements. This can be done by
using a word processor or by dedicated tools like DOORS or reqtify. After capturing the requirements they are analysed and a rough system design is developed by a modularisation of the system and a functional description of the different modules and components. For the functional specification the engineer uses tools like Rational Rose, Rhapsody or Statemate Magnum.

If an engineer models a functional component or designs a special functionality fulfilling a specific requirement he can now express the relationships between the entities of the functional model and entities of the requirement model as illustrated in Figure 1. How this is done depends on the depth of tools which are integrated into the engineering environment. Ideally the engineer just drags and drops the entities between the different tools. Then the engineer can specify the types of the modelled interrelations.

By doing this work the engineer generates an interconnected product model spread over the requirement tool and functional modelling tool. The resulting integrated product model stores very useful information about the design rationales of the developed system.

![Figure 1: Networked requirements and functions blocks](image)

Now another engineer can see why specific designs are done in the way they are by getting to know which requirements were part of the design rationales of the initial developer. This information can also be used for a lot of other applications e.g. the generation of the design documentation. Another usage is the automatic validation of the design system. The model interrelations can help to recognize requirements that are not taken into account for the design. Additionally IT components can identify which system areas have to be checked if a dedicated requirement changes.

The Semantic Web technology is chosen to implement the product model interrelations because of its capabilities of easy integration in XML files and generic semantic expression. In the proposed fine-grained integration approach the product data stays inside the tools and will not be exchanged between the tools. The idea is to connect the product model entities instead. The different tools are used to model the different parts of the product models. Theses parts may represent product modules, dedicated models (requirements, functions, CAD etc.). These parts of the product model of a product are not independent. Examples for product module interrelations are signal interfaces in system engineering or the tolerances of a mechanical connection between two geometric parts.

This concept can be followed all along the development process. The interrelated model represents a lot of product knowledge, which helps engineers to keep the product model up-to-date during the design process. If interrelated entities change, the related interrelations will be adapted as well. The product knowledge can also be used by expert systems to do model
analysis as described in the example. The entity interrelations can also express constraints. Constraints also help to support the product model consistency.

The linked product/system model entities define a network which represents knowledge about the entity relationships. This network allows an intelligent navigation through the network and represents engineering knowledge beyond the subparts of the model handled by the engineering tools. Knowledge management tools can be used to manage und to use this network. The RDF and RDF Schema technology helps to link this interrelation knowledge with explicit knowledge handled by the knowledge engineering system. PLM systems are integrated into this approach by using their process and document management capabilities.

The product data can stay in the proprietary file formats of the used engineering tools. This reduces the integration efforts. The data entities can be interrelated across the system boundaries and beyond the data model capabilities.

To access and review the information the CAE tools or viewers are used. There is no information loss generated by the exchange of information using standards, which does not cover tool specific model features.

3.2 Semantic Web Technologies

Driven by the development of the Internet and the huge amount of available information the technologies of the Semantic Web have emerged. As already mentioned in the last subchapter the Semantic Web technology is very suitable for the implementation of the proposed concepts. The Semantic Web technology implements a navigation map for the Internet by offering intelligent navigation between information resources based on the relatively old technology of Semantic Nets.

The development of the Semantic Web followed two ideas: To invent a high level navigation and information retrieval method for the Internet and to describe information resources that can be found in the Internet. Knowledge-based systems use the Semantic Web Technology to model specific knowledge (e.g. Ontologies [Gruber 1993], Constraints [DAML+OIL 2001]) and interrelate it to dedicated information entities.

In an ontology engineering process the semantic concepts (topics) of a dedicated domain are defined. This ontology is used to describe and link information resources.

The most important technological base for the Semantic Web is the Resource Description Framework (RDF) and RDF Schema (RDFS) [W3C 2003].

RDF is a very easy way to express interrelations between information resources with XML (a semantic map) or to define topics by referencing them to information resources. Furthermore RDF can be used in XML files (or HTML) to annotate existing information entities by adding meta information. A simple example for this is the linkage of a tolerance in a CAD model with an entry in a requirement management tool. RDF expresses tupelos like T(Subject, Predicate, Object). The object can also be another subject. For the example this is notated as T("the drive controller", “fulfils”, “requirement1”) and expressed in XML using RDF:

```xml
<?xml version=1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3c.org/1999/02/22/rdf-syntax-ns#"
  xmlns:s="http://myschema.de">
  <rdf:Description
    about=http://wi-ol.de/part/drivecontroller>
    <s:fulfils>
      requirement1
    </s:fulfils>
  </rdf:Description>
</rdf:RDF>
```
RDFS is a technology used to express an ontology for the elements of the tupelo. Therefore RDFS is capable of describing classes for subjects, predicates and objects and to define their relations (e.g. inheritance), axioms and constraints.

With this technology, the user can define a domain ontology as well as describe and link the information resources by using semantic specifications. This makes the orientation in the network very easy. The user can navigate through the distributed model with the help of his semantic map. The Semantic Web is a very powerful method to capture development knowledge and to make it usable by the user and by knowledge engineering tools.

3.3 Product Life Cycle Integration

In a lot of development teams PLM systems are in use. The capabilities of these systems can be extended by the introduced knowledge-oriented model interrelation approach. As depicted in Figure 2 the PLM system handles the so-called macro model of the product. The macro model can be product structures, part classifications, sales structures etc. The PLM system manages also the documents/files related to the different parts of the product model. These files store the micro model of the product. The micro models contain the requirements, functional specifications, CAD models, NC programs etc.

The interrelations between the elements of the micro model cannot be handled by the PLM systems. Because the consistency of the micro and macro models is very important a lot of efforts are spent in the integration of this models.

As depicted in the figure the introduced fine-grained model integration approach helps to maintain the interrelation between the entities of the different micro model files. The concept supports the linkage between micro model and macro model entities.

The PLM system can be used to manage the process and macro model management as usual. By using the RDF technology the product model management capabilities of the PLM system are extended. If a file is changed the entity relations can be analysed to decide which other files have to be adapted to the recent changes of the product model to ensure the consistency of the design.
4 Integration Platform Architecture

A platform architecture is developed in order to integrate the distributed product model entities by using Semantic Webs. This platform offers basic services to define and manage the knowledge network extending the product model maintained by the different CAE tools.

The component-oriented architecture of the platform is based on works on Knowledge Management system architectures [Gronau, Laskowski 2001] reusing PLM System functionality. The inter component communication is based on Web Services (SOAP).

The platform follows a three layer functional architecture depicted in Figure 3. The lowest layer consists of repository and integration components. These components are used to store the semantic information inside relational or XML based database management systems, RDF Files or as meta information stored with the tool specific XML representation. An access component homogenizes the access to this information and provides a transparent mapping of logical references (expressed by XLink [DeRose et al. 2000]) to physical addresses.

Integration components are used to communicate with the integrated tools. This covers the retrieval of meta information and the addressing of specific model entities.

The second layer contains basic knowledge management components. These components are a Topic and Ontology Server which manage and navigate through the semantic network, a Catalogue Server which maintains and navigates through development knowledge organized in information catalogues (product components, functions, solution elements, ...). A Rule Solver component supports knowledge processing.

The third layer provides the user applications of the platform. These are collaborative components for the management of the documents, a knowledge editor and navigation and retrieval components. If it is possible the developer adds meta-information with the GUI of the tool used. In the other case this can be done with the knowledge editor. The third layer also offers knowledge administration components to maintain the ontologies and knowledge catalogues.

Now it is feasible to develop additional engineering tools that provide functionality on top of the semantic network. These tools can easily access semantic specification of the interrelated information. Examples are tools for the analysis of the developed models e.g. a requirement-tracking tool that checks the fulfillment of requirements during the development process.
Other tools are intelligent knowledge retrieval and catalogue search functionalities that examine the actual development context and models.

5 Conclusion

In this paper a concept and architecture for fine-grained product model and tool integration is described. By using Semantic Web technologies product model entities from different parts of the product model handled by different CAE tools can be interrelated. Additionally the product model entities can be linked to external knowledge resources. The interrelated product model stores valuable knowledge about the product under development.

The platform manages the fine-grained interrelations in the product model. Document and process management is done by integrated PLM system functionality. Because the model information stay inside the tools by using the tool specific model schema the integration efforts are clearly reduced.

By using this platform all needed information and knowledge can be provided at the developers’ workplace. The Semantic Web technology helps the engineers to understand and to manage the product model. RDF and XML technologies help to reuse the knowledge stored in semantic network also by other tools and knowledge management systems.

References


DAML+OIL Reference Description http://www.w3.org/TR/daml+oil-reference, 2001


Staab, S.; Erdmann, M.; Mädche, A.: Ontologies in RDF(S), ETAI Journal, 6, 2001
