Metadata Repositories for Virtual Engineering[*]

Ingolf Geist, Dipl.-Inf., Stephan Vornholt, Dipl.-Inf.

1. Introduction

Efficient and effective support of the product design process in general and Virtual Engineering (VE), in a particular case, requires the possibility of the automatic generation of problem-oriented models. Especially, it is necessary to extract simplified geometry, mechanical and electrical models from CAD data. The derived models are enriched with additional product data information and used as input of simulations. In the recent years many tools and algorithms emerged that allow automatic transformations between engineering domains. As transformations are complex operations, it is necessary to combine different approaches to a process of transformation steps that may be executed automatically or semi-automatically. Finally, outcomes of the combined processes are simulation results that describe the behavior of the investigated products.

During the process models and data are transformed, but also metadata, e.g., information about CAD and simulation models, product specifications, and simulation results, are transformed and transferred. Therefore, it is necessary to manage not only the data transformations in the VE process but also the flow of the metadata. Users, i.e., designers and analysts are interested in the history of a simulation result or a design decision besides other kinds of metadata about the VE data. Here, two central questions arise: where does the data come from and how was the data transformed? These classes of queries are denoted as data lineage queries (Tan 2007) and can be answered with help of a *Metadata Repository*. The requirements, tasks, and challenges of the introduction of a Metadata Repository into the VE process are the focus of this work. The remainder of the work is structured as follows. Section 2 describes an example Virtual Engineering process and its data and metadata. Section 3 summarizes the kinds of metadata that a Metadata Repository has to manage. An architecture proposal is given in Section 4 and the paper concludes with a summary and further research steps in Section 5.

2. Processes in Virtual Engineering

In this work we assume the Virtual Engineering process as the phases of Design and Simulation/Analysis in the product life cycle. Figure 1 shows a typical process that is based on the work of (Juhasz, Schmucker 2008). Starting point is a (intermediate) CAD model that contains the geometry, the structure as well as additional information like used materials and electrical properties. The goal of the process is the analysis of the behavior of a mechatronic product using a simulation.

The process comprises several parts. At first, (semi-)automatic methods are used to transform the CAD model into a multi-body system (step 1). Additional sources, e.g., product databases, contain information about electrical devices like engines which are added in the next step (2). Necessary parameter values are extracted using automatic data extraction steps (3) or are manually assigned (step 4). Based on this information, the initial model structure is translated into a simulation program, for example into a Modelica system (Modelica Association, 2007).

---

[*] This work is supported by European Commission: European Regional Development Fund.

Thereby, basic models are reused that describe the behavior of systems parts, e.g., body, joint, engine, collision models. The results are new complex model components that can be reused in other systems. Furthermore, a simulation program is created that is used in behavior experiments and for visualizations. After the model derivation workflow has been carried out, the experiment workflow starts. Experiments use derived and enriched simulation programs as well as a set of parameter value variants that are extracted either from the original CAD model or from additional sources. The simulation program is parameterized (step 4) and executed by a simulation system, e.g., Dymola (step 5). The results are tested (step 6) against requirements, and parameter values are either adjusted or the design in the CAD model is modified to meet the requirements. If the requirements are met, other tests can be carried out. Figure 1 also shows several data formats and data models that are used during the VE process and which have to be managed as well as to be transformed into each other, too.

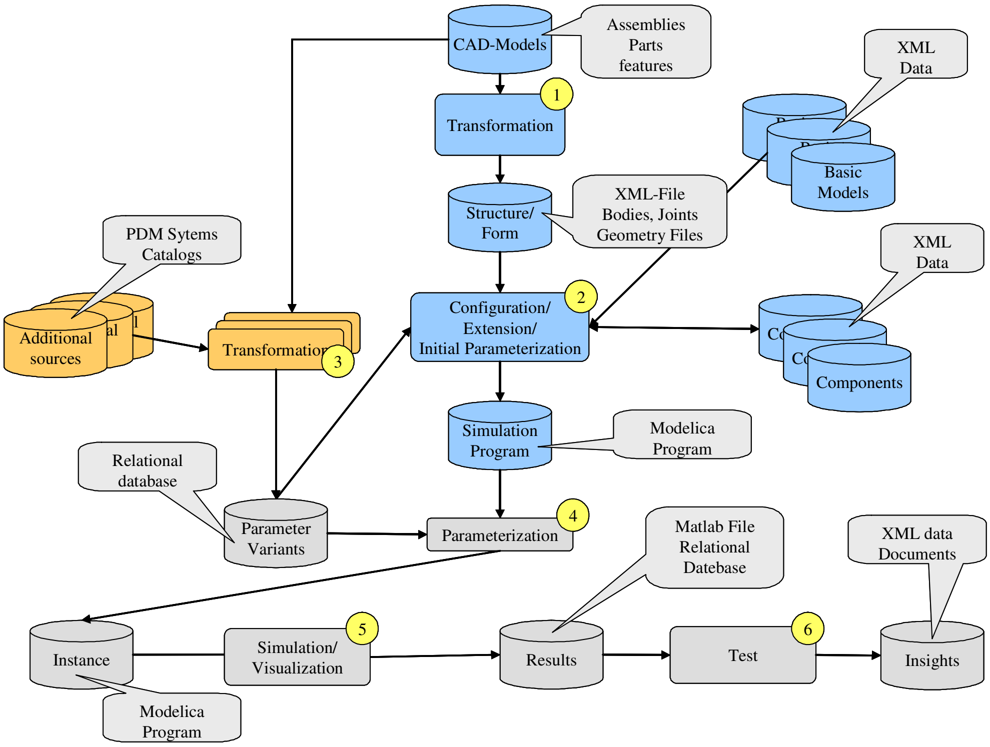

Figure 1: Exemplary VE process illustrating the semi-automatic derivation of a mechatronic simulation from a CAD model and additional sources

The results of a workflow are used in different ways, including: (i) immediate usage, the process is mostly automatic and the designer will use the results directly; (ii) the models are created by simulation experts using the CAD model, they parameterize models semi-automatically and test many variants for a given structure. The results and insights are returned to the designer; (iii) the created parameterized models are reused in other, more complex simulation models, e.g., an industrial robot is reused in a production line simulation; (iv) the visualization information is post-processed and used in a Virtual Reality environment.

The process and the use cases require a management of the metadata within the system besides the development of intelligent data transformation systems, in order to allow the documentation and verification of VE processes and their results, the retrieval of past solutions, and the support of systems and humans to discover domain-spanning correspondences. Furthermore, metadata about experts and designer improves the social interaction between different teams.

3. Metadata for Virtual Engineering

After the introduction of an exemplary VE process, we now present categories of metadata that are essential in virtual product development processes. In general, metadata in VE processes can be distinguished into following categories:

- Structural metadata,
- Workflow metadata,
- Descriptive metadata,
- Lineage metadata and
- Technical metadata.

Technical metadata describes the characteristics of the systems, data access, used operation systems, versions of programs, etc. We will discuss the other four parts in the remaining section in detail.

Structural metadata

Structural metadata is classified into the description of tool- and domain-specific data models, description of the structures of local data as well as integration information. The information can be distinguished into different levels:

- description of the local data models in every domain using a common notation and
- description of the structure of local data and models of every domain using the local data models.

In the first level, the constructs of the particular domain- and tool-specific data models are specified. In that way, all local data models can be described in a common language. In the second level, the structures of the data in the different sources are stored. For example, data models are the object-oriented model of Modelica for simulation environments or CAD data model consisting of part, assemblies, and features. In contrast, second level structural metadata describes the structure of actual models. For example, class, parameter, and port names of a Modelica model or the assembly hierarchy of a CAD model with part names and parameter values.
Integration metadata represents the correspondences between different domains and models which describe semantic dependencies between structural elements, e.g., the material parameter of a CAD model and the mass parameter of a body model. As correspondences are created with help of complex algorithms or manual assignment, it is essential to store them in the metadata repository, e.g., as mapping tables (Bettaieb, Noel 2007).

Workflow metadata

A workflow model allows the explicit definition of activities comprising a VE process (Bose, Frew 2005). In the context of VE, workflows are used to track the data and its transformations made by different systems and one individual or a small group of individuals. Workflows are represented by directed acyclic graphs (DAG) or other network types. Representatives are Petri nets or state charts for instance (Bose, Frew 2005). In Virtual Engineering, workflows have to be adaptable because new insights during development process require different models of different accuracy, different original data sources, etc. However, parts of workflows can be reused and recombined. For example, consider the VE process in Figure 1.

The experiment workflow can be reused in later experiments even if varying model transformation and parameterization steps are preceding the experiments.

Descriptive Metadata

Descriptive metadata helps users and systems to find, to document, and to connect resources. The metadata is defined as free text or terms of a pre-defined ontology or taxonomy (Li et al. 2008). Furthermore, it can be structured using different attributes or be unstructured as text. Descriptive metadata can be connected to every resource in the VE process, i.e., to models, structural and workflow metadata, instances, parameter values, and design results, as well as to systems and transformation steps. In that way, the complete VE process can be documented. Typically, metadata is either stored in a Metadata Repository or as annotation together with the data (Bhagwat et al. 2005). A metadata repository has to ensure that descriptive metadata is kept connected to the corresponding resources. An example of descriptive metadata is the documentation of the meaning of a certain parameter value in form of free, unstructured text.

Data lineage metadata

Data lineage comprises all information about a process that was used to develop a product, i.e., where does the original data originate from, which transformation processes have been used, which other data was involved in result construction. Thus, data lineage information explains and documents the history of design decisions based on the collected metadata about transformations and data. Lineage information is distinguished into fine-grained data provenance and coarse-grained process descriptions (Tan 2007). VE requires both kinds of lineage data and requires the combination both kinds of lineage information to audit the VE process, which is an open research problem. Data extraction from product databases for parameterization requires a detailed analysis which values are extracted by which data operations. I.e., fine-grained control of data provenance is necessary. Coarse-grained lineage data describes the transformation steps that include complex algorithms, e.g., CAD to multi-body system transformation. Here, data values cannot be traced in detail.

4. Architecture

In this section we propose an architecture that introduces a metadata repository which is illustrated in Figure 2. The architecture is based on the requirements to scientific data management presented in (Bose, Frew 2005) and on a lightweight data management solution for product development presented in (Ding et al. 2007)

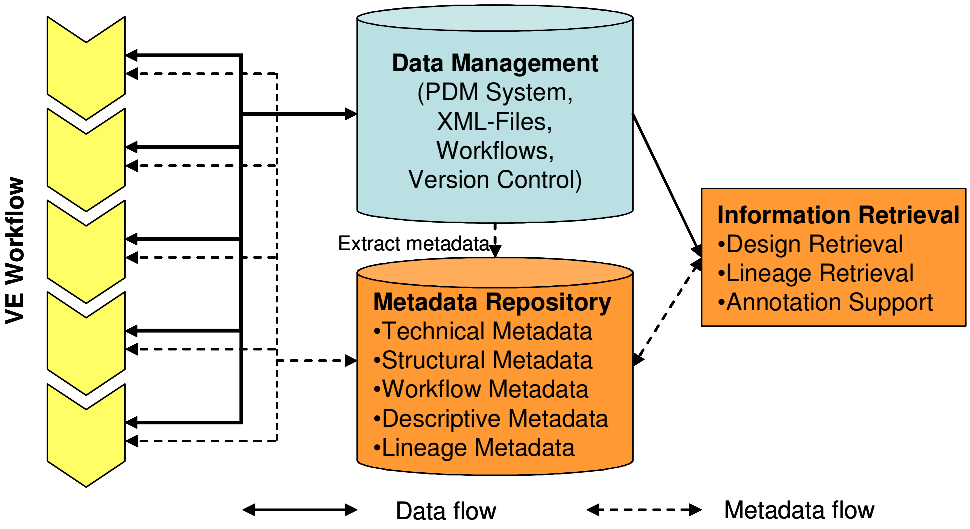

Figure 2: Proposed Architecture

and (Bettaieb, Noel 2007). The models and knowledge that have been created and used during the VE process are stored using a data management solution, for example PDM systems or version control systems are usable. It is planned to use domain-specific XML languages (Modelica XML, Ansys XML, X3D, PLM XML, etc.) for storing development results in the particular engineering domains. Equivalently, XML languages exist also for workflow modeling (PDL). Additional XML languages for simulations results have to be developed. Thus, a lightweight VE data repository is built on top of an XML database.

The Metadata Repository manages all kinds of information that were described in Section 3. The system will be based on an XML database. Structural and workflow data as well as technical metadata is extracted from the data repository. Domain-specific data models are described using XML schemata. Integration data is managed as mapping tables and organized in integrated components. Furthermore, an ontology of every domain is managed to describe resources. A first version of an ontology is modeled based on the structure of standard libraries e.g., the Modelica standard library. As a top-down developed taxonomy or ontology is rigid and slowly evolving, annotations are supported as second kind of descriptive metadata by the metadata system. Thus, the dynamic nature (changing development teams, using of other engineering domains, fast development progress, new requirements, etc.) of the VE process is supported easily.

All data and meta information are used and updated by the tools within the VE process. They will be accessed through a Web service interface. Furthermore, a metadata repository Web application allows the retrieval of design and metadata in a domain spanning way. In order to represent the iterative nature of the product development process, the metadata repository has to support different versions of metadata, too.

5. Conclusions

We identified in this paper that systems supporting VE processes have to include Metadata Repositories. Metadata Repositories store important information about the VE process and transformation and creation of data and metadata. The meta information can be distinguished into five types. The metadata is especially important to retrieve existing designs and to verify and to document designs and simulation results. Furthermore, we proposed an architecture taking into account a metadata management during the virtual product development. The next steps are the specification of the data schema of a metadata repository and the evaluation of a prototype in an ongoing research project in the automotive area. Furthermore, we want to investigate how fine-grained data provenance and coarse-grained process information can be combined in VE processes.

6. References

Bettaieb, S., Noel, F., 2007, A generic architecture to synchronise design models issued from heterogeneous business tools: towards more interoperability between design expertises. Engineering with Computers,online first.

Bhagwat, D., Chiticariu, L., Tan, W. C., Vijayvargiya, G., 2005, An annotation management system for relational databases. VLDB J., 14(4):373–396, 2005.

Bose, R., Frew, J., 2005, Lineage retrieval for scientific data processing: a survey. ACM Comput. Surv., 37(1):1–28.

Ding, L., Ball, A., Matthews, J., McMahon, C., Patel, M., 2007, Product Representation in Lighweight Formats for Product Lifecycle Management (PLM). In 4th Int. Conf. on Digital Enterprise Technology, September 2007.

Juhasz, T., Schmucker, U., 2008, Automatic Model Conversion to Modelica for Dymola-based Mechatronic Simulation. Modelica 2008, March 3rd-4th, 2008, pages 719 – 726. The Modelica Association.

Li, Z., Raskin, V., Ramani, K., 2008, Developing Engineering Ontology for Information Retrieval. Journal of Computing and Information Science in Engineering, 8, March 2008.

Modelica Association, 2007, Modelica - A Unified Object-Oriented Language for Physical Systems Modeling – Language Specification – Version 3.0, September 2007.

Tan, W. C., 2007, Provenance in Databases: Past, Current, and Future. IEEE Data Eng. Bull., 30(4):3–12.