# Data-driven and Integrated Engineering for Virtual Prototypes

**Stephan Vornholt**
**Otto-von-Gurericke University Magdeburg**
**Magdeburg, Germany**
**vornholt@iti.cs.uni-magdeburg.de**

**and**

**Veit Köppen**
**Otto-von-Gurericke University Magdeburg**
**Magdeburg, Germany**
**koeppen@iti.cs.uni-magdeburg.de**

## ABSTRACT

The increasing usage of components in automotive systems is accompanied by complex dependencies of products as well as engineering and economic domains. Thus specialized analysis approaches and simulations are used to identify influences of domains and components on each other. Current research deals with database systems and data management approaches to integrate different engineering domains into a component oriented model. On the other side integrative management solutions made huge approaches in the field of computer controlled economic analysis and optimization. ERP systems and Data Warehouse approaches enable a wide range of data storage and analysis solutions. A challenging task is the interaction as well as the influence of different engineering and management disciplines on each other. In this paper a comprehensive solution for data driven and integrated engineering of prototypes in the automotive domain is presented. This approach enables early updates of concepts and structures as well as re-combination of virtual products including the planning process. Based on the description of challenges the field of domain expert integration, an integrative architecture is presented.

## 1. INTRODUCTION

The increasing usage of software based approaches in automotive systems is enabled by the support in the fields of computer applications and further research on control systems for complex dependencies. These allow tests and analyses on virtual prototypes before implementing and manufacturing real prototypes. A challenging task is the interaction of different components as well as the influence of different engineering and management disciplines on each other. On the engineering side the CAx techniques and concurrent virtual/digital engineering approaches enable complex product engineering, taking i.e., the phases design & construction and analysis & simulation into account. On the other side the data based management systems made huge approaches in the field of computer controlled part lists, cost control, and delivery optimization. The interaction of both research areas promises a faster and cheaper design phase as well as continuous open decisions. To support this goal an integration of both domains, management information systems and concurrent engineering, is necessary.

The early adaption of ideas and concepts is enhanced by virtual engineering (VE) in the field of mechanical engineering solutions, which lead to a huge amount of variants. Many influences have to be considered and tested including the market. Business questions have influences on the product re-design as well as the design process itself can influence business constraints. Thus, an active cross linking of tools and production partners is required. Models and strategies can change and make a re-design of a prototype necessary. This approach enables early updates of concepts and structures as well as re-combination of virtual products including the planning process.

### EXAMPLE

The simplified development of an automotive system, denoted as CAR, is presented as an exemplary process following the product lifecycle, see Figure 1. Following on the basic goal definition a shared concept is developed, where constraints are defined and the conceptual design is more or less specified. Hereupon, in the design & construction phase engineers develop iteratively simulation models and virtual prototypes which are analyzed and, if the result fits the requirements, forwarded to the process and production planning as well as market and economic analyses. The white arrows represent desired steps of enhancing the model up to production, while the black arrows illustrate possible re-engineering decisions.

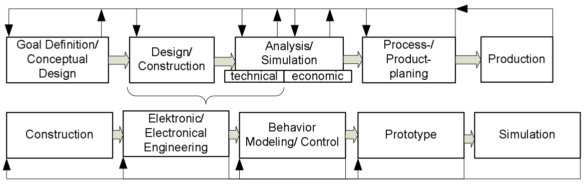


**Figure 1: Life Cycle**

The design itself concentrates on the geometry and connections as well as on mechanic systems, electrical drives, and controlling components. A designer of the elementary geometry could assume the structure: one body, two axis constructions, which have themselves an axis, two wheels, and two connection elements, see Figure 2a. The connection between different elements is also defined. Many mechanical definitions like weight, volume, or material properties can be used as well as predefined connection types between elements, see Figure 2b.

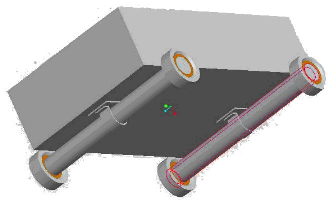

**Figure 2a: Example CAR (3D CAD)**

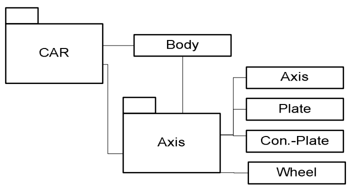


**Figure 2b: Common concept CAR**

According to our example, the following domains cooperate in this process:

- Electric design: the engineer designs the electric drives of the system and the electric power management. Information from conceptual design, geometry, and product libraries, e.g., engine databases, are combined.
- Simulation & Analysis: the effects are simulated with models. The results of simulation are used to identify weak spots or deficiencies and give information for further improvements. The backbones of product analysis are typically the following domains:
- Mechanical analysis, e.g., SimMechanics: mechanical simulation models are used for analyzing the mechanical behavior of prototypes, e.g., kinematic behavior, testing of possible collisions, and allowed movements.
- Mechatronic analysis, e.g., Modelica described in [4]: the mechatronic model adds electric drive and control components to the mechanism model.
- Finite element (FE) analysis (ANSYS & NASTRAN): FE models are used to analyze the elastic behavior of certain system parts. Typical tasks of FE analysis are the computation of resonant frequency or bending stress [6].

After these technical analyses several economic management tests arise, i.e., marketing, production planning, cost, and inventory. The product can be chosen to build, re-designed with new or adapted constraints, again including economic and technical decisions. Depending on the suggested changes the process of engineering is restarted.

## 2. RELATED WORK

The combination of approaches in Virtual Engineering (VE) and Enterprise Resource Planning (ERP) systems and an integrative architecture, combining the positive effects of new database approaches for VE and ERP systems require a deeper view on the systems.

### Virtual Engineering

The product development process involves several specialized domain experts with own vocabularies, knowledge, and tasks. Thus, every expert uses domain tools and enclosed information management systems, but all work on the same conceptual design and exchange information with other groups. The interaction and heterogeneity of used data models and information systems lead to data interoperability or integration aspects and management problems within product development processes. VE targets the computer-supported parallelization of design & construction as well as simulation & analysis in order to reduce product development time and cost. A virtual prototype (VP) represents a computer based prototype of a real world artifact. It can be tested from different points of view. The VP as well as tests can be visualized in Virtual Reality (VR). Therefore, VPs comprise and combine all information, e.g., CAD geometry designs, product data, behavior models based on FE analysis, or mechanism models, and special geometry models for tests in VR. Synthesis steps, e.g., parameterization, and analysis steps are alternating during product development. In order to shorten and improve the development process, existing and verified designs, simulation models, and VR scenarios have to be re-used, modified and re-combined for new developments. Several solutions including the integration into one common data structure, e.g., presented in [1, 2], transfer of information, e.g., in [3], integration into one common tool and a storing system [5] are in the focus of research. All approaches lack in either exchanging information or specialized view or re-design.

### ERP

ERP is a company-wide computer software system used to manage and coordinate all resources, information, and functions of a business from shared data stores. ERP software is a recent addition to manufacturing and information systems that have been designed to capture and organize the flow of data for the whole product life cycle. ERP software attempts to link all internal company processes into a common set of applications that share a common database. It is the common database that allows an ERP system to serve as a source for a robust data warehouse (DW) that can support sophisticated decision support and analysis.

ERP software usually has a central database as its hub, allowing applications to share and re-use data more efficiently than previously permitted by separate applications. The database of an ERP system is functional- or process-oriented organized [14]. In an ERP system the data cannot be efficiently analyzed directly, because ERP uses OLTP (Online Transaction Processing) to handle the data. OLTP is a class of program that facilitates and manages transaction-oriented applications, typically for data entry and retrieval tractions.

Data can be exchanged among disparate systems especially disparate CAD systems by Step or related solutions like MechaStep or IGES. For storing the design data it should be first archived within the organization that produced the part. Today many big corporations have archived data using a CAD/CAM format that is no longer supported by any vendor. Existing database approaches such as constraint databases are being investigated to archive design data in neutral exchange formats. To avoid geometry predominance, integrated design could be organized around an integrated product model managing design information. Current approaches usually assume that integrated design needs a unique integrated model. Typical solutions are described in [7, 8].

Product Data Management (PDM) systems are used to be integrated into a common environment within the product lifecycle. PDM is a tool that helps engineers to manage engineering data and product development processes. As PDM systems are widely used to reduce the product development time, they need to exchange product data with CAD systems. It is necessary to integrate CAD and PDM systems, because CAD systems generate the product data and PDM systems manage these data. The management of product structure data is the main function of a PDM system. PDMs should enable engineers and other users to search design models and re-use knowledge and company best practices by combining artificial intelligence techniques such as neural networks and expert systems with CAD and object-oriented databases.

Comprehensive solutions, integrating different disciplines are just researched and implemented for individual companies. These solutions are not useable for engineering clusters with many participating companies. Although the list of solutions in the disciplines themselves is long, i.e., product development, economic solutions, data exchange, an integrated approach is missing. An open architecture integrating heterogeneous systems in the same database or a common database schema as a solution for concurrent control and exchange of information is necessary and will be presented in the next section.

## 3. VE & ERP INTEGRATION

In this section the integration of both views of the engineering process is described. This solves on the one hand the addressed challenges and on the other hand enhances further the development process, due to use of all available information in the complete product life cycle.

**Challenges**

Although there are several data based solution systems in the engineering and management disciplines, where in each discipline itself the integration of different domains is in the focus of current research, a data management of both is not yet one of these solutions. The result is that both disciplines work on the same product with high concurrency and data redundancy, but the interaction between both design lines is limited to an input and output. The other discipline is seen as a black box.

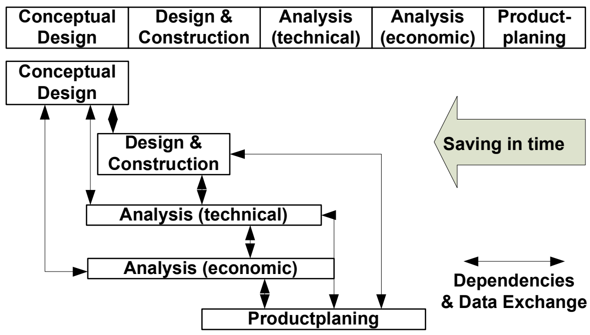


**Figure 3: Concurrent Engineering & Management**

By integrating both approaches into a common data schema, the main tasks of each domain should not be changed. Both need comfortable complete data storage with metadata management, data integrity, version management as well as the combined multi user interaction, consistency control, and specialized views. Those requirements for a new system are described below to enhance engineering and management at the same time, see Figure 3. The integrated data schema enables concurrent work processes, i.e. technical and economic analysis, where engineering steps can use "in-work" models to estimate their results.

One of the main challenges for databases is the (re-) definition of views. Since every cooperative partner has its own tools and data definition language as well as a concept of necessary information, the data integrated for one product has to be separated in many ways. These can be combined ones, for example the parts list, where products for the mechanical and electrical engineer are analyzed and enriched with additional information stored in databases as well as cost lists from PDM systems. The

user view on the data is the same as if he would work alone.
Comfortable complete data storage with metadata management is essential. This means, that different tools are used and it is not possible to have integration with manual re-definition of data and connections. Most work should be done automatically and every data should be storable and their metadata managed.
Information stored in the new data model should influence design decisions. Therefore dependencies and conductions have to be defined and used. Furthermore, those dependencies between information have to be held in a consistent state. An electrical engineer designing a connection between two elements which are not connected in mechanical model are inconsistent as well as constraints given by the design concept cannot be ignored by the engineers.
One of the most complicated tasks is the integration of different users, working on the same model with their own view, storing and updating the model at the same time. These changes have to be integrated in a persistent way.
Especially in heterogeneous production networks, consisting of many companies, the possibility to protect someone's information from the access is necessary. However, data security issues necessary in many forms even for each designer, who wants to work without the risk of changes in his area. This means, that he needs a warning for incompatible/inconsistent data instead of an automatic change routine is an essential requirement.

**One adaptive Schema in VE**

To build one adaptive schema both parts and their integration schema have to be considered:
In our scenario, three general data models are used: a CAD data model, a data model for mechatronic systems, and a FE data model. Typically, CAD systems use a feature-based, parameterizable, hierarchical data model. The construction is organized into assemblies, sub-assemblies, and parts. The properties of assemblies and parts are denoted as features and parameters. A part feature is its volume, i.e., its geometry. Additional features are for instance material or surface specifications. Assemblies group parts or other assemblies and assign positions like parameter values to them. The mechatronic data model consists of the mechanic model and the electric model. All components have parameters, e.g., inertias, center of gravity, mass. Components are classified into bodies and engines, while they are connected by ports. A port has a defined position on a component. The FE model is based on a mesh model of the possible simplified original geometry. This geometry can be derived from the CAD system, but can also be an abstract simplified geometry, e.g., 1D or 2D. The mesh model consists of mesh entities that are distinguished into elements, faces, edges, and nodes. Parameters are assigned to elements to describe materials, movement constraints, and masses.

Connections and hierarchies in the models are differently expressed and denote different real world concepts: for instance, in the CAD model, a hierarchy means a construction hierarchy and in the mechanic model connection corresponds to kinematic dependencies. Therefore, simple 1:1 correspondences between data model elements are only partly possible. Often complex conversions are necessary that also take actual models and instances into account.
Most ERP models are based either on the customer or product, but are just divided collections of reachable data without connections or interdisciplinary character. Thus the character of data models is in the structure, information area and data model are hard to characterize. To overcome this, current research is focused on the integration of different ERP solutions into one DW and also about the question why past research in this area is not used in current ERP systems. The integration of ERP systems in this paper assumes a DW, where different ERP solution databases are integrated into a common data structure which is developed and enlarged with the tasks of the integration database.

**Integration-Architecture & Schema**

The integration architecture can be divided into three basic components: the ERP data warehouse, the VE integrative DW and the multi database.
The ERP data warehouse consists of a basic structure, which holds product based information on products or product parts. The UID (unit id) represents the link to different material or part list databases. Further information, e.g., on cost, can be stored as well as customer information or delivery information. The information schema in the data warehouse is integrative, which means, that different tools are tested of consistency and can change data which is already defined. An adjustable feedback function translates changes on relevant data back to each related system, or just adds a new version with a warning to all inconsistent data variants. Furthermore an adaptable view function is included. The VE database is based on the solutions of the component based virtual family definition in [9, 10]. Here, several closely related information on technical descriptions are managed and held consistent. Feedback, views, and consistency are integrated, too. The Multi database integrates both approaches into one schema, as depicted in Figure 4.

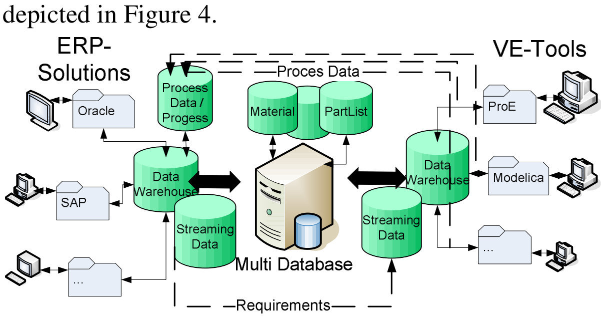


**Figure 4: Integration Database (Basic Idea)**

The Multi database is based on a component-oriented model that describes VPs in a multidisciplinary way. Figure 5 illustrates the main concepts of the model using UML notation.

A domain model describes the behavior, form, or function of a VP in one engineering domain, e.g., mechanic behavioral model or the CAD model. A domain model is composed by sub-models. Since ERP data models are combined and often not especially related to one basic model, they can be allocated to the component directly. Each model provides a set of parameters and of ports. Parameters are quantities that describe the characteristics of the model and the characteristics of a VP in a certain domain. A port is a connection point where models can be combined. Through connected ports signals, material, and forces are transferred. The libraries are integrated in the ERP data model as well as common ERP data with their constraints and specified processes. In summary, a model is represented by a 3-tuple M = (id, parameters, ports) while ERP models are represented by ERPM= (id, constraints, processes, parameter).

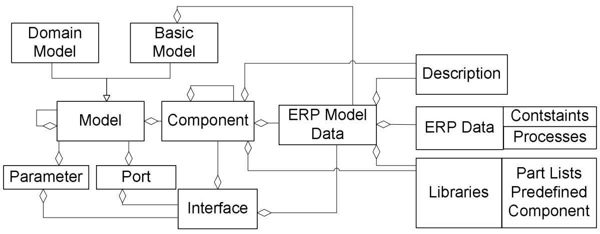


**Figure 5: UML Schema**

Each domain model and ERP model represents one view to a VP and does not support inter-domain correspondences. Therefore, components are introduced which are hierarchically organized. A component represents a conceptual part of a VP and encapsulates all models as well as their dependencies for this artifact. A component ensures the dependencies between different domains, while domain models combine sub-models within one domain. Furthermore, components can contain sub-components. Components provide interfaces for communication and parameterization. An interface consists of parameters as well as ports. Constraints and mappings ensure the consistency control within one component. Combining the concepts, a component is defined as a tuple C = (id, M, C, Pdept, PMdept, I) with an identifier id, M and C are sets of encapsulated models and components, respectively. Pdept represents a set of port mappings between different domain models and PMdept is a set of parameter dependencies. Constraints and processes can be translated into dependencies or parameter. Finally, the interface I, consisting of a set of external ports and parameters, describes the behavior of the component to the environment. The interface is mapped to internal models and components.

Figure 6 illustrates our exemplary component CAR. The component contains of two domain models, a CAD model as well as a combined ERP model. A set of dependencies describes the internal relations between the domain models. The external interface offers ports and parameter that are mapped internally to the domain models. The component can be distributed and instantiated in the coupled way allowing the usage directly in CAD as well as ERP models.

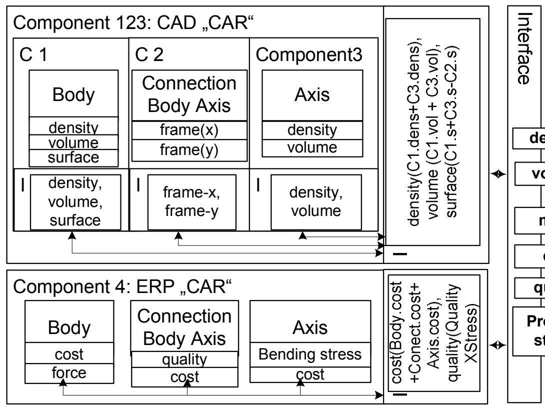


**Figure 6: Component Construction of CAR**

**Views**: An information overload results if too many details are used at the same time in the development process. Views filter the relevant information for each costumer, the engineer as well as the manager or initiator. The views can be integrated in the schema and processes as illustrated in Figure 7. The arrows illustrate the combination and integration of data representations as well as the influence of changed properties or parameters. Based on the (meta-) data which are stored in files and folders the integrated data are stored in one component based structure. Inconsistent information is eliminated and libraries as well as files are linked in the (meta-)data repository.

The global view contains all parameters and ports in each file and can be addressed directly. The economic interface contains all relevant DW information on ERP data and hides the technical information. The technical interface contains technical data from virtual engineering. Business and economic information are not viewable. Another defined view is the combined interface, which contains information on both or information which is necessary in both disciplines as well as new views on the data. Views are defined and planned to be supported by an adaptive view selection where dependencies are defined. Thus any combination, new sights and approaches are possible.

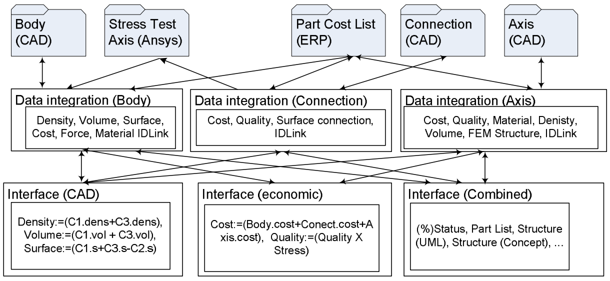


**Figure 7: Views**

## 4. Conclusion and Outlook

In this paper we present data driven and integrated engineering of prototypes in the product development domain, which is illustrated by an automotive example. Based on the description of a virtual engineering integration solution the ERP data warehouse integration into a common structure is focus in this paper. The new integrative architecture is based on the challenges in the field of domain expert integration. Especially, the view concept enables different approaches to use and specify the integrative data as well as dependencies and further control options. The next steps in this field are the integration of different ERP systems in one data DW and the implementation of the system. Further steps and integration areas, such as logistics or training, are planned to be done in the future.

**Acknowledgements**

The work of Stephan Vornholt is supported by the European Commission: European Regional Development Fund, COMO C1-3201201 and C3-3201201. The Work of Veit Köppen is funded by the German Ministry of Education and Science (BMBF), project 01IM08003C.

**References**

[1] Fenves SJ, Foufou S, Bock C, Sudarsan R, Bouillon N, Sriram RD, **“CPM 2: A Revised Core Product Model for Representing Design Information”** Technical Report, NISTIR 7185, 2004.

[2] Xue D., Yang H, **“A concurrent engineering-oriented design database representation model.”** Computer-Aided Design, 36:947-965, 2004.

[3] Juhasz T, Schmucker U, **“Automatic Model Conversion to Modelica for Dymola-based Mechatronic Simulation.”** In: Bachmann B (ed.) Proceedings of Modelica’ 2008, The Modelica Association, Bielefeld, p. 719-726, 2008.

[4] Modelica Association, **“Modelica - A Unified Object-Oriented Language for Physical Systems Modeling -- Language Specification -- Version 3.0”**, 2007.

[5] Bettaieb S, Noel F, **“A generic architecture to synchronise design models issued from heterogeneous business tools: towards more interoperability between design expertises.”** In: Engineering with Computers, 2007.

[6] Armstrong CG, **“Modeling Requirements for Finite-Element Analysis”**, Computer-Aided Design 26(7):573 – 578, 1994.

[7] ISO, **“ISO 10303:1994 - Industrial Automation Systems and Integration - Product Data Representation and Exchange”**, 1994.

[8] Pavez L., **“STEP Datenmodelle zur Simulation mechatronischer Systeme.“** Abschlussbericht des Verbundprojekts MechaSTEP, Forschungszentrum Karlsruhe GmbH, 2001.

[9] Vornholt S, Geist I**, “Flexible Integration Model for Virtual Prototype Families.“** In: Proceedings of PLM08 5th International Conference on Product Lifecycle Management, Seoul, Korea, 2008.

[10] Vornholt S, Geist I, **“Interface for Multidisciplinary Virtual Prototype Components.”** In: 19th International Conference on Database and Expert Systems Application (DEXA), 1st International Workshop on Data Management in Virtual Engineering DMVE '08, 2008.

[11] Esteves J., Pastor J., **“Enterprise resource planning systems research: an annotated bibliography.”** In: Communication of AIS 7(8), 2001.

[12] Silwood Technology Ltd, **“Saphir White Paper: Metadata management for Enterprise Applications.”** Manual, 2008.

[13] CA, Transforming IT Management, **“Technology Brief: CA ERwin Saphir Option.”** Manual, 2009.

[14] Simon, Elke, **“Implementation of an Enterprise Resource Planning System with Focus on End-User Training“**, Hamburg: Diplomica Verlag, 1999.