

Dependencies Between Design Product Models and Simulation Models

Reiner Anderl, Sven Kleiner

Darmstadt University of Technology, Department of Computer Integrated Design, Germany
E-mail: kleiner@dik.tu-darmstadt.de

Keywords Concurrent Engineering, design, simulation, product data model

Abstract Multidisciplinary engineering is essential to develop mechatronic products. The key factor is to integrate methodology, systems and models. This paper will discuss the various design representations and the dependencies between product models. An approach for capturing dependencies is presented.

1 INTRODUCTION

According to a survey there is still a big deficiency concerning the application of methods like Simultaneous Engineering, Concurrent Engineering and Systems Engineering in European product development environment [3]. Especially the missing integration of different engineering domains is a handicap of today's product development process. Specialised methods are established in different fields of engineering, but these are unknown or obscure for participants of other engineering disciplines. Hence, a common understanding is of crucial importance for process success.

Besides methodology, information technology affects productivity of engineering work and product quality. Numerous software tools for product design, analysis and simulation support engineering. Tools for design (CAD-systems) respectively computation and simulation (CAE-systems) result in specialised tasks, where skilled experts are required. Engineering tools are divers and mixed, according to requirements of users who work with such tools. This leads to an isolated work concerning design and engineering.

From the point of view of engineering, the mechanical engineering scene has increasingly incorporated electrics and electronics for the control of the mechanical elements. Mechatronics is an integrating discipline that combines the fundamentals of computer science, mechanical and electronic

engineering to meet the needs of industries involved in automation/robotics and related areas where there is a need to integrate these technologies. Tools for the design and simulation are needed and play an important role to evaluate designs and to support design decisions. Problems made organisations realise, that they have to co-ordinate the way product developers collaborate and to interconnect engineering tools. Reuse of product models and the mapping of data can reduce the gap between design and engineering.

2 ENGINEERING OF MECHATRONIC PRODUCTS

The methodology of Systems Engineering is part of the systems approach. This means that the system (product) consists of sub-systems, components and parts with borders and relations to the environment [4]. It is important to make a mechatronic design from systems approach. Advanced knowledge of experts from different disciplines is necessary in order to get the best performance. Providing the multidisciplinary design team with valuable methods, tools and information will enhance the productivity.

2.1 Methods and Tools for Design

The feature based and parametric design approach used as modelling techniques in CAD-systems have led to a massive progress in efficient design and engineering of products and represent the state of the art.

Design by features extended solid modelling, by dividing complex geometry in simple elements (features). This features can be classified in body features, form features, operation features and enumerative features, e.g. holes, threads, slots, pads etc. Every feature has its own semantic and could be customised by parameters. Most CAD-systems offer libraries with features, which can be combined and reused in different ways. The totality of pre-defined and user-defined features represents the geometric (static) design of a product. The main aspects of modelling with constraints are structuring a solid model as a history of features, using topology objects and their geometric parameters and applying constraints to these objects. The definition of geometric shapes is strongly supported and can be performed easily. The variation of designed geometry is easy to define [6]. These modelling techniques allow an easy way to modify models, re-use products, create product variants and families besides analysing the model consistency. In future, knowledge-based CAD-Systems will capture, retrieve, modify and create rules to establish designs based on best practices.

2.2 Methods and Tools for Modelling and Simulation

Within the scope of virtual product development, computer simulation allows to evaluate and to compare alternative designs without having an expensive hardware prototype. Modelling of systems behaviour and simulation is important for designing mechatronic systems. In controller design, many different software tools for system analysis and simulation are available [1]. In core these systems are based on very efficient mathematical-numeric libraries for the most important tasks and include functions for modelling, simulation and prototyping as well as for analysis and visualisation. The product is described by an underlying mathematical model, where the coefficients represent the essential product parameters.

The modelling of mechatronic systems is mostly supported by multi-functional, mathematically oriented calculation and simulation packages (e.g. Matlab/Simulink or MatrixX/Systembuild), where the simulation model is presented by a block diagram. A precondition in a block diagram is that different blocks do not constrain each other's properties or parameters. This implies, that parameters of various physical components need to be defined in different (sub)blocks. In most cases, block parameters are defined separately and automatically related to the properties of the block diagram, in order to investigate the effects of parameter variation. As a consequence therefore, system components represented by blocks cannot easily be substituted by others [2].

Object-oriented modelling techniques using iconic diagrams (e.g. electrical network diagrams), energy based modelling approaches (e.g. bond-graph) or more general modelling languages including ready to use model libraries (e.g. Modelica) do not have this problem. Different software packages have been developed, that support modelling and simulation with modelling languages (e.g. Dymola) or bond graphs (e.g. 20-sim, CAMP-G) besides using equations and block diagrams. In future, viewing the mechatronic system in various representations (multiple views) can help to get a common understanding and a good insight of its properties [1].

2.3 Design Product Models and Simulation Models

The design of a mechatronic system can be split into a number of distinct tasks. Different tools to solve these tasks are available, but an integrated design environment is not existing. An integration of simulation causes handling of product data, which have been produced in earlier design

phases, e.g. system properties like mass, pressure, load. In other words, the mathematical model for the simulation consists of parameters and constraints, which are predetermined. All these data have to be considered within simulation otherwise there will be a gap between data used in the simulation and data generated during design. Furthermore, new features of CAD-models could represent other product properties as well, which are used for modelling and simulation. In consequence of a physical description of the design that contains all the data that are relevant for a certain class of simulation experiments, designers could interact with this representation to formulate simulation experiments in physical terms [8].

As a result, the need for an interface between design product models and simulation models is one of the key issues for successful integration.

3 CONCEPT OF INTEGRATION

A holistic approach, which is based on integrating design, modelling and simulation, comprises a three-level-architecture. The integration of methods and processes (methodology), systems and models. The integration of methods and processes means supporting methods and operations within departments, companies and co-operations as well as workflow management. The main objective of system integration is a spanning communication in the CAX environment providing (network-)service. Model integration signifies that different tools using proprietary product data models are sharing data using a common data base. The result of model integration is a discipline overlapping product model. To limit the scope of a general product model, a number of special purpose models modularise the common product data model and are tailored to certain design aspects.

3.1 Integration of Systems

Regarding communication between engineering tools and sharing of engineering data the following architectures are popular.

First, tools are connected directly. This architecture implies an one-to-one interface between tools using data access methods of provided application programming interfaces (API). APIs depend on software and software releases. Operation and service of one-to-one interface is very expendable. Second, the interconnection of tools is indirect. The communication between engineering tools is supported by Engineering Data Management Systems (EDM) using data files or a data base, which are defined by neutral data formats.

Besides direct/indirect interconnections, the continuing revolution of standards like CORBA (Common Object Request Broker Architecture) or DCOM (Distributed Common Object Model) will drive architectures to flexible integration concepts based on object-oriented principles, where tools and techniques used can communicate actively their data.

3.2 Integration of Product Models

With the assistance of a common product data model, a data mapping between CAX-tools can be provided. For the integration of different model views a defined data model structure is crucial and entails efficient methods for data mapping between all interesting views. The idea of STEP (Standard for the Exchange of Product Model Data, ISO 10303) is based on a single, standardized product model, that holds all relevant data. However, the missing of parametric, constraint-based and feature-based model characteristics is a handicap of the STEP specification and the described additional relations between design and simulation are not yet part of any STEP product data model (Application Protocol).

The question comes up, how to cover dependencies between different product models and how to establish a common product data model from bottom up. The following approaches require knowledge in both the engineering domains and information modelling to be successful.

3.2.1 Model Definition from the Process View

From the point of view of engineering processes, data exchange between product development tasks need to be analysed first. The model definition task from process view can be participative or expert driven to different degrees. In a participative project the users have a strong role through the entire definition phase. Their knowledge will be utilised to secure a properly working framework. Usually, the engineering designers in the company are the people that have the best knowledge of how the product development process actually works. While most companies have a formally defined product development process it is not always followed in the actual day-to-day work. Data modelling experts lack this knowledge and can thereby only adapt the model framework after the prescribed process (ideal case). However, the expert can provide input on how the product model interacts with the entire CAX environment and supports data modelling.

According to data and information requirements, a first rough application independent framework should be defined after having a model of the product development process. A customised structure of the information content needs to be formed additionally, which should be

independent from implementation for further development and modifications. The users should improve the data exchange processes associated with product development before the model definition. It is possible to define a product model from an already existing product development process and data exchange issues, although doing this will not fully utilise the power of new information technology.

3.2.2 Model Definition from the Product View

Data model definition from the product view is based on native data models representing different product views. The main issue is to map the various product descriptions in a common data model. Therefore, native data formats need to be analysed first. The analysis should result in a common structure of the information content, which is the most difficult part. A very important issue is to check model structure and dependencies regarding properties (parameters), structuring data classes and adding important class properties subsequently [7]. Figure 1 illustrates data models from a Simulation System (Matlab/Simulink) and a common reduced data model for design and simulation.

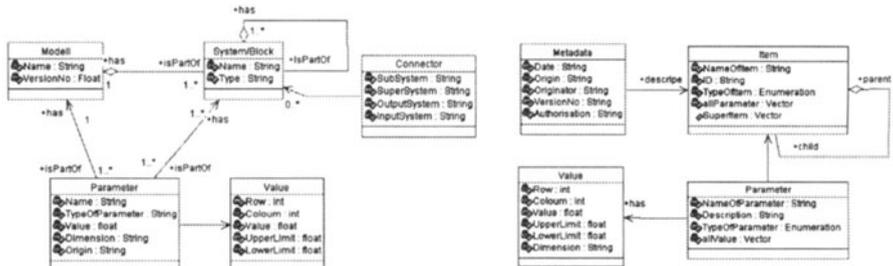


Figure 1: Data model for a block diagram and a merged data model (simplified)

Design product models are used by CAD-systems, whereby different systems are working with different native data models. CAD-models are representing mainly product structure (e.g. assembly hierarchy, bill of material) and product geometry (shape). A model in CAD-systems may be described as a composition of static product properties, which are safeguarded by analysis methods and tasks (e.g. structure analysis, modal analysis, multi-body-analysis). The main purpose of CAD-model is the same as of drawings, they are blueprints for product manufacturing and assembly.

With modelling and simulation programs, the behaviour of dynamic systems, such as electrical, hydraulic systems and any combination of these systems can be simulated. Mostly, systems are modelled using block diagrams. Block diagrams may be described as a composition of lower level

submodels. Submodels consist of blocks, which are (pre-) defined icons connecting inputs/outputs and representing equation descriptions internally.

The mapping between different product descriptions must be transparent. The structure of block diagrams and design models are divergent due to different approaches in modelling. Therefore, rules of interpretation (mapping rules) must be obvious for engineers when relating blocks of simulation models to parts of design models (assemblies).

The object-oriented physical-systems modelling approach using model description languages (e.g. Modelica, Sidops+, NMF) supports data mapping from design product models to (object-oriented) simulation models straightforward, especially when using elements from model libraries such as clutches, pumps, motors etc. Object-oriented physical system models do not have an influence because of encapsulation and a non-causal (declarative) model description. Consequently, modelling physical systems in an object-oriented way could easily adapt the structure and properties of CAD-models and hence, share a common product data model (Table 1).

Table 1 Data Mapping between a general Design Model and Simulation Models (excerpt)

	CAD (Design Model)	SIMULINK (Block Diagram)	MODELICA (OOModel)
Product	Assembly	System	Model
Item	Subassembly	Subsystem	Submodel
	Part	-	Component
Element	Feature	Block	Class
Relation	Mating condition	Line, Branch	Connection
	Constraint	-	Equation
Parameter	Parameter	Parameter	Parameter, Type

4 IMPLEMENTATION

In accordance with the proposed architectures for the integration of systems and models, neutral or standardised data formats are the key enabler for successful implementation. Within STEP, the translation to and from file formats is supported by off-the-shelf tools (e.g. Caselib from ProSTEP GmbH). In this case, data translation requires the EXPRESS language where both data formats are to be defined by EXPRESS schemes.

In a first prototype implementation of an environment for data management of aircraft actuators, a general conversion tool using indirect connections has been developed. This tool supports data transfer between different applications and has an interface to a PDM-system, which ensures simultaneous work with the same or related information. The neutral data

scheme was developed following the process-oriented approach and implemented using Java [6].

5 CONCLUSIONS

Based on the situation described in the previous sections, development of data models drives the integration of basic services for multidisciplinary engineering. Hereby a structure of the information content has been presented according to mentioned product views. The differences between design product models and block diagrams as one description for simulation models has enforced a resulting data model representing only parameters.

The object-oriented simulation methods and models are structured and robust enough to get beyond, integrate multidisciplinary engineering concerning methodology as well as models by enabling automated data mapping with simple algorithms. At this point, the collective approach of design, modelling and simulation of mixed-domain systems as well as the integration of object-oriented modelling techniques require further research.

6 REFERENCES

- [1] van Amerongen, Job: *The Role of Control in Mechatronics*. In: Engineering Science and Educational Journal, Vol. 9, Nr. 3, pp. 105-112, ISSN 0963-7346, 2000
- [2] Broenink, Jan; Kleijn, Christian: *Computer-Aided Design of mechatronic systems using 20-SIM 3.0*. In: Proceedings of WESIC'99, 2nd Workshop on European Scientific and Industrial Collaboration, Newport, UK, Sept 1st-3rd, 1999
- [3] Grabowski, Hans; Geiger, Kerstin: *Neue Wege zur Produktentwicklung*. Stuttgart: Raabe, 1997
- [4] Haberfelner, Reinhard: *Systems Engineering*, 9. Auflage. Zürich: Verlag industrielle Organisation, 1997
- [5] Kleiner, Sven et al: *Data Management for Mechatronic Systems*. In: Proceedings of the 6th International Conference on Concurrent Enterprising. Toulouse, France, 28-30 June 2000
- [6] Mendgen, Ralf: *Methodische Vorgehensweise zur Modellierung in parametrischen und featurebasierten 3D-CAD-Systemen*. Aachen: Shaker Verlag, 1999.
- [7] Rumbaugh, James et al: *Object-Oriented Modelling and Design*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc. 1991
- [8] Sahlin, Per: *A tool for data mapping from design product models to object-oriented simulation models*. In: Proceedings of the 11th European Simulation Symposium: Simulation in Industry, October 26-28, 1999, Erlangen, Germany