Lappeenrannan teknillinen yliopisto
Teknillinen tiedekunta. LUT Metalli
Tutkimusraportti 81

Lappeenranta University of Technology
Faculty of Technology. LUT Metal
Research report 81

Rantalainen Tuomas T.
Kortelainen Juha S.

Managing Off-Line and Real-Time Multibody System Models

ISBN 978-952-265-049-8
ISBN 978-952-265-050-4 (PDF)
ISSN 1798-4319

Lappeenranta University of Technology
Faculty of Technology
Department of Mechanical Engineering
P.O. Box 20, FIN-53851 Lappeenranta

Lappeenranta 2011

Managing Off-Line and Real-Time Multibody System Models

Tuomas Rantalainen[1] and Juha Kortelainen[2]

Contents

1 Introduction

Modelling and simulation has become a general tool in product development of mechanical products. The use of simulation enables the designers of the product to have feedback of the design and the interference of different sub-systems of the product. While the power and efficiency consumption of mechanical systems is increasing, the dynamics and thus also load, noise, and vibration problems are becoming more common due to lighter structures, higher loads, and faster operations. By using simulation from the early phase of the design process, many problems that arise from dynamical interaction of different sub-systems of the product can be avoided.

One major factor of the overall machine performance and function is the operator. Simulation is a good approach for taking into account the influence of the machine operator. To get full benefit by simulation-based approach, simulation should be taken into design process at the early phase. This simulation-based approach in product development enables separate design tasks to be performed concurrently, but may introduce challenges for modelling data management, when applied widely.

[1] Lappeenranta University of Technology

[2] VTT Technical Research Centre of Finland

In off-line simulation, operator influence is simulated with signals that are simple mathematical formulas or statistically collected data. To get realistic operator feedback, real-time feedback from simulation model is crucial. Real-time feedback can only be obtained by real-time simulation.

1.1 Objective of the Work

The objective of this work is to find and describe methods and approaches that can automatically or with minimal manual work transform off-line simulation model to real-time model.

The objectives of this work are:

1. to describe fundamentally the process and requirements to automatically compose a real-time model from an off-line model and
2. to describe the method of collecting all modelling data of a machine into one complete model, a product model that is easily manageable and that automatically updates one modification to all necessary sub-models.

The most eligible solution will contain low amount of redundant data and in most ideal case, there will be no overlapping data at all. The third aim is to seek and propose a method that combines the first and the second objective.

1.2 Scope of the Work

In this work, the focus is on mechanical systems and machines. The work contains a short literature review and new ideas are proposed. The scope of the work is on automatic composition of real-time model and modelling data management.

2 Challenges in Simulation-Based Design Process

2.1 Introduction to the Concepts of Off-Line and Real-Time Simulation

In a product design process, the objective of the simulation dictates the level of details and modelling principles in general. While the focus of the simulation is to predict loads and structural stresses, the accurate enough description of the geometry, structural flexibility, and behaviour of the actuators in the system are important. On the other hand, if the objective of simulation is to design and match the control system of the machine for specified operations, often more coarse mechanical model can be used. In general, if the objective of the simulation is to design a subsystem to fulfil the defined technical requirements, the computational speed is usually not in high priority, but only the accuracy of the model for the purpose matters. These kinds of models are called off-line models. On the other hand, a real-time model is a model that can be simulated in such a way that one simulated second is computed in a real second or faster. Real-time models are used for e.g. testing real control system hardware with a simulated plant mechanical system or studying human-technology interaction of a working machine. To be able to give the operator realistic feeling about controlling the machine, real-time feedback is needed. The mentioned cases require accurate timing of the system compared to real world.

To date, there is no general solution to combine the off-line and real-time simulation models of the same system. Even though, off-line and real-time models both represent the same product and have the same basic parameters. In most cases, the target system is such complex that the computational efficiency of the off-line model is far from real-time.

2.2 Simulation-Based Product Development Process

From the simulation point of view, product development can be seen as a series of sequential actions that lead to the complete product. In Figure 1 on the left is depicted a traditional straightforward design process and on the right a simulation-based design process in exaggerated manner. In some cases, simulation is used as an additional verification step at the end part of the design process, after which all fundamental design parameters have been permanently chosen. In this case, if simulation gives new piece of information to the process, it is laborious to transfer this information backwards in the design chain. Simulation-based machine design approach can be seen as an iterative process that goes iteratively through design steps. One design step is completed after simulation. This is due to the fact that by using simulation the designers can get feedback about the design decision quickly and cost efficiently. In addition, virtual prototype is developed and kept up-to-date all the time along the main design and is thus available for use.

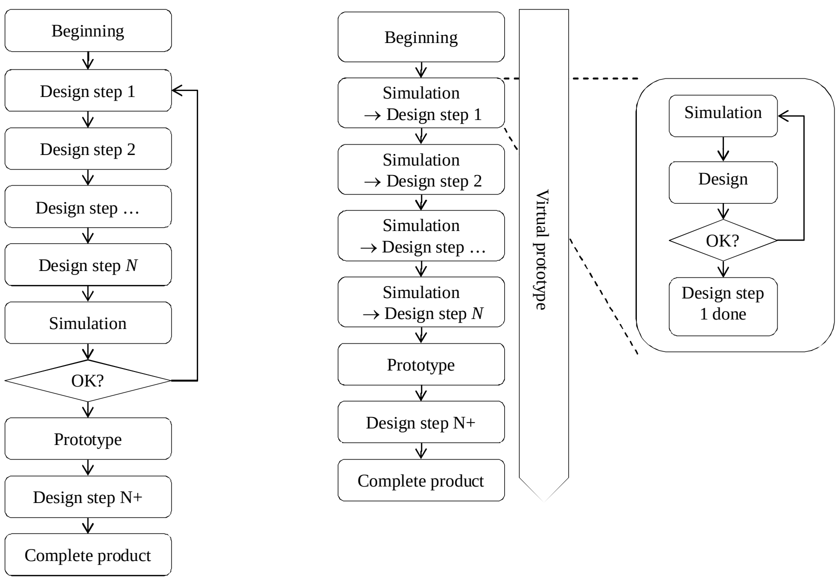

Figure 1: Simplified flowchart for traditional (on left) and simulation based design process.

In many cases, the machine operator has a significant influence on fatigue life of the machine. In the traditional design process approach, the influence of the operator is taken into account usually at the prototype phase, which is one of the last phases in the design process (Figure 1). In that phase, large amount of effort and resources has already been used and if there are some problems that need to be solved, it often means laborious modifications to those details that should already have been made on previous design phases. Going back to early design phase is time consuming and expensive, and can be avoided by using simulation through the design process. In a simulation-based design that takes operator influence into account, there is a demand to compose real-time model for every design phase. Manual composition of the real-time model will significantly increase the use of resources and decrease the attractiveness of the simulation-based design process. If the real-time model can be composed automatically or almost automatically out of the detailed off-line simulation model, it will decrease substantially the need of manual work; the saving in work time comes in every design phase.

2.3 Challenge of Managing Modelling Data

In Figure 2 is depicted an example of a machine under design process and the models and sub-models that are related to the machine via different modelling software. In a development process, the machine is typically described using several separate models. Models can contain completely or partially overlapping modelling data and in case of problems with version management, also contradicting data may exist. The models describe the same machine but differ from each other in two ways: in the purpose of usage and in the amount of accurate details. In addition, it is noteworthy to mention that in a design process there might be several different versions of the same model. Models are also typically updated separately. That kind of separate update will lead to unnecessary work and will increase the risk for a human error.

The amount of labour in managing large number of models will decrease the attractiveness of the usage of simulation in product development. This is understandable because of the additional work for keeping up models will reduce the total advantage.

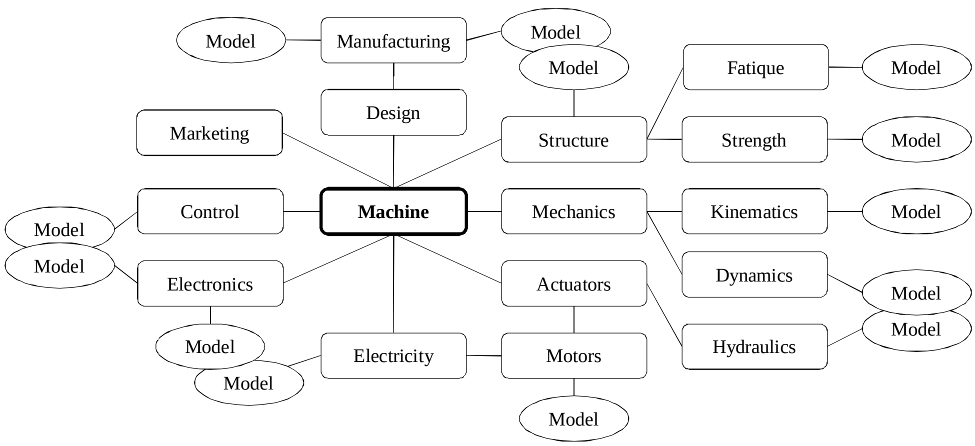

Figure 2: Models, sub-models and software that are attached to machine in a design process.

Typically commercial simulation software applications are concentrated on representing some specific and detailed problem in some specific simulation domain, and to create an overall product model, several separate simulation models need to be integrated together. This approach contains many challenges, e.g. sub-models do not necessarily have strict and accurate boundaries and same phenomena might be modelled in several sub-models from different point of views. In addition, one separate sub-model may describe several sub-problems either completely or partially.

One markable source for problems is the use of commercial software that do not interact with other software tools. Program independent way of describing modelling data will help in data transformation and it can also act as a neutral way of storing modelling data.

3 Modelling Data Management and Exchange in Multibody System Simulation

One possible solution for the problems in the context of multibody dynamics expressed in Section 2, i.e. automatic composition of a real-time model from an off-line model and scattered modelling data, is a general model. A general model would gather all modelling data into the same place and this could be the approach to reduce redundant data. A general model can describe the whole system, including multibody model and also some other models such as control system and fluid dynamic models.

To this point, general and software independent way of describing multibody dynamics model, not to mention multi domain simulation, does not exist. Already in 1997, Schiehlen [1] pointed out that the standard form of describing multibody system data does not exist. Standardizing could act as general and software independent approach for multibody dynamics. This problem is well known, and it has been recently studied [2], [3] and [4]. In this work, those previous studies and one new suggestion are introduced. Formerly proposed principles for general multibody models can be categorised as:

- document based general model, e.g. XML-based transfer format [4] and
- database based general model, [2] and [3].

A general model can contain, in addition to modelling data, metadata that for instance stores information about the modification history of the model. For the automatic composition of a real-time model, some information about the computational efficiency can be stored as metadata. Even thought in a general case there is no way to know a priori absolute need for the computational power for a separate component, the computational load can be estimated relatively among complementary components. Even thought this relative estimation is subjective, it is made a priori and there is no need to make the same estimation by the user every time when the real-time model is composed. The need for computational power can be estimated by classifying and counting equations that need to be solved for that component. In addition, a test simulation can be solved for obtaining estimation for performance of the component.

3.1 Document-Based Model

In a field of modelling and simulation, models are becoming more detailed and larger systems are being simulated. If modeller would like to use existing model components, e.g. dampers or actuators in a large model, there has to be a connection between the component and the model. In general, that is not a case. Another problem is that many of the modelling software are commercial and typically, if transformation for a component is made with older version of modelling software, this transforming will not necessarily work with a new version. Previously multibody modelling data is transformed with general model that is written in XML (eXtensible Markup Language) [5]. XML is designed to be a document format and it can be used as a transformation format between systems. The XML language is developed by the World Wide Web Consortium (W3C). By the definition, XML is not restricted to describe any certain type of data and therefore it can also be used for representing multibody system models. Document-based general model could interact directly with separate modelling tools (Figure 3).

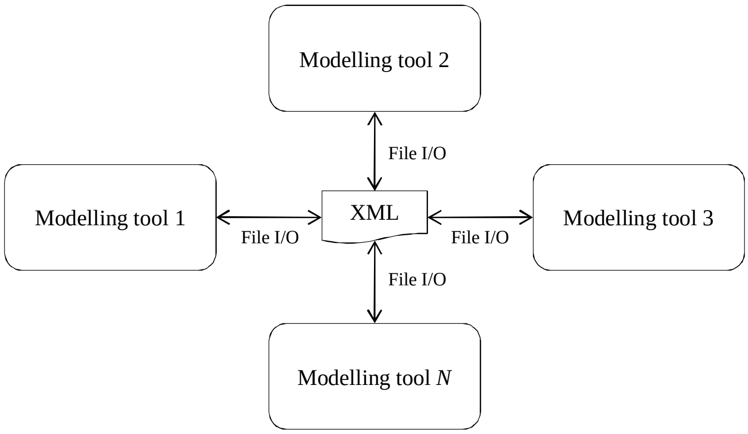

Figure 3: Document-based modelling data exchange.

A document-based system, where a document acts as a general transfer format, can be used as a general model. In a document-based management of modelling data, all data transformations between programs are made via transfer documents. Siemers et al. [6] proposed a meta-modelling environment based on a XML. In that environment, they connected several modelling programs together and the meta-modelling environment managed the simulation and collected simulation results. Gonzaléz et al. [4] introduced and designed a new XML-based modelling language for modelling multibody systems MbsML, which is based on XML.

General documents-based model is a straightforward approach and in this model, data is represented in a structural form. In small and nicely bounded models, well organized and defined document-based model is efficient. The limits of the document-based approach can be reached by modelling large systems and linking modelling data to other data sources. Modelling data that is cumbersome to present in a document-based general form is e.g. element meshes and geometry of a complicated structure.

3.2 Database-Based Model

Efficient methods for managing continually growing and complicating modelling data have recently been under development. Tisell and Orsborn [3] proposed a solution to the problem of managing modelling data. They noticed that even thought there are many software applications (for instance Dymola [7]) in the market that can handle multibody dynamic modelling data in a symbolic form; those applications are lacking features for storing and handling large amount of modelling data. They proposed that all modelling data for one multibody dynamic system should be collected into one database that can handle data in symbolic form. Their proposal said nothing about transferring data between programs during simulation. Even if all modelling data is collected into the same data storage (i.e. document- or database-based), it does not necessarily mean that they can exchange modelling data, not to mention data exchange during simulation. In Figure 4, two fundamental concepts for transferring data between two systems are depicted. The thick arrows represent modelling data flow and the thin arrows represent data flow during simulation.

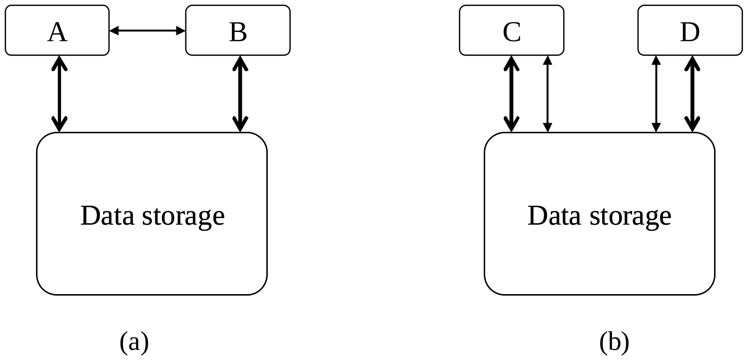

Figure 4: Data exchange during simulation, (a) straight connection and (b) through the data storage.

In Figure 4 left side, the programs A and B use the same data storage and they can even use the same pieces of modelling data in a simulation. Data exchange during a simulation is done by connecting the two programs together. This approach is not general and it will lead to very complicated system when the number of communicating programs increases. In the left side of Figure 4, the programs C and D get modelling data from the data storage and they also can exchange data during simulation trough the same data storage. The approach is general, and in this approach, one program can be replaced by another and it has minor affect on others.

It is possible to have multiple users simultaneously accessing the same database that is working as a general model. Ma et al. [2] shared design data with an active database system. Active database system automatically synchronizes model data in a database and informs connected users about the change of data.

Whilst all modelling data is in the same data storage, it is possible to get rid of redundant data. In addition, data management and modification becomes straightforward when all data is located in the same place without a need to store data to several separate programs. Database-based data management is also software independent when talking about connecting software. Only interface between database and software needs to be implemented. Every program with suitable interface can query, access, and modify data elements. Using a database-based system is a clear improvement to the situation when all modelling data is distributed into separate files with varying file formats.

4 Composing a Real-Time Simulation Model from an Off-Line Model

The main aim of the off-line model is to achieve good and accurate simulation result and the time a simulation takes to run is not important. Real-time simulation, on the other hand, has to go along with real time. Automatic composition of real-time simulation model can be divided into following sections:

- simplifying the model:
  - removing computationally demanding components or replacing them with more light ones and
  - simplifying functions of model components;
- representing modelling data in a more efficient computational form:
  - describing model topology in a recursive form and
  - symbolic optimization on level of equations.

4.1 Simplifying the Model

While composing a real-time model automatically, it is crucial to know or automatically estimate computational load during simulation for a single modelling feature. In a multibody system model, many components can be described with multiple levels of details. Some detailed features such as contacts, friction, and structural flexibility, can make the simulation model computationally heavy. For contact and friction models, there are several replaceable models with varying computational efficiency and therefore, for a real-time simulation the automatic selection of components is possible. The solution is to add several contact and friction models to the same general model and then choose a proper one depending on the required level of run-time or accuracy of the simulation. For describing structural flexibility, there are several methods but usually they are computationally heavy and that is why the most straightforward method to simulate multibody dynamics in real-time is to assume bodies to be rigid.

In commercial software applications, the most often used method for describing structural flexibility in multibody dynamics is to use the floating frame of reference formulation [8]. In that formulation, flexibility is described by assumed deformation modes. Accuracy of the method depends on how many deformation modes are being used and what deformation modes are included. In this method, the level of real-time can be selected by adding or removing modes for simulation. While using the floating frame of reference formulation, the main problem is to automatically select proper deformation modes. There is no general method or criteria to guarantee the selection of proper modes. For reducing computational load, the number of deformation modes should be minimized but still being able to have reasonable accuracy in deformation. One approach to automatically select proper modes is to use deformation energy method. In that method, one simulation with test load is analyzed and those modes that describe the largest amount of strain energy will be chosen [9]. The drawback in this method is the need for the test simulation. Loads for the test case should be selected properly, because the modes will be optimized based on that specific case.

4.2 Describing a Computationally Efficient Multibody System Model

When modelling data is represented in a general form, information about the topology can automatically be used to increase the computational efficiency of the simulation. An efficient method for reducing unknown coordinates to increase computational efficiency is to describe system topology in recursive form [10]. The principle idea behind recursive method of representing coordinates is to refer the body translation and orientation to the previous body in a kinematical chain. Adding one body to the system increases solvable coordinates only by few. The drawback of this method is that it has suffered from singularity problems. Recently recursive methods have been under research with promising results [11] and some of the singularity problems have been solved.

On an equation level, computational efficiency can be obtained by selecting coordinates in an optimal way, selecting proper form for equation of motion and selecting proper solver. This has also been under research and e.g. Cuadrado et al. [12] have been doing research on this topic. Methods for reforming equations of motion are not necessarily general and if they are applicable, they can improve computational efficiency of the model.

5 Managing Simulation Modelling data

With one modelling method and modelling software, it is common to model physical phenomena from a narrow point of view. For example, a multibody system model is used for calculating dynamic response of the system and a finite element model is used for calculating structural stiffness and stresses caused by various loads. In addition, one specific phenomenon can be described in several programs using varying accuracy. For instance, real-time and off-line multibody system model represents exactly the same system but with different level of details. For that, there is a reasonable and historical reason. Earlier one computer had very limited amount of computational power and for convenience, large systems were divided into smaller fractions and complicated phenomenon were simplified. However, whilst computational power of computers has been increased it has become possible to simulate larger systems with more detailed models. Nowadays the amount of modelling data is continuously increasing and managing the data is already a problem. In addition, modelling data is scattered into smaller portions that are not stored or even connected together and it makes modelling data management even more cumbersome. Geometry data can be stored in various models and formats, such as in CAD- and FE-model as well as in computational fluid dynamics and multibody system model. When geometry data is changed in a general case, the change is not automatically exported to all models.

The problem of managing overlapping modelling data can be tried to solve by automatically exporting changes from one program to another. This kind of approach needs interfaces to be implemented between all programs. After updating the program, in general, if interface is made by some other entity than the original provider of the program, interface will not work and it has to be remade. In addition, one of the basic problems, i.e. the scattered modelling data among various programs, is not solved in this approach.

5.1 Semantic Database-Based Approach

A possible solution for solving the presented problems, i.e. automatic composition of real-time simulation model and modelling data management is to generate one general model or several general sub-models that can interact with each other. In this approach, if one program is updated or replaced with another, only that program specific interface have to be created again while all other interfaces remain. In addition, in this approach every program has only one interface to database. This is beneficial compared to the concept in which every program has an interface with each other, in which the number of interfaces grows progressively while number of programs grows linearly. In Figure 5 is illustrated the principle of using semantic environment for integrating the off-line and real-time simulation model of the product.

A semantic database, such as Simantics [13], contains several domain ontologies that can be mapped to each other. Real-time model can be composed out of domain components by applying semantic restrictions and rules.

Philosophically semantic database concept and a relation database concept are more far away from each other than technically; the fundamental idea in a semantic database is to store knowledge, whilst a relational database is for storing data. In traditional databases, data elements are stored into tables and they are mostly designed for a data that can be represented in predefined form. A distinctive example of a traditional database is a member register for an association. The semantic form suits for data that is inconvenient to be represented in a table form. A descriptive example of that is a data element that can have vast amount of properties but has in practice only arbitrary few number of them.

Proposed approach solves automatically both the real-time model assembling and the data management problem in one solution but brings up some new challenges. Some of the remaining research challenges related to the solution are:

- semantic database technology, especially for modelling data management, is still new and has not been established,
- ontologies needed by a semantic database do not exist yet (but they are under development), and
- commercial software does not have interfaces available to semantic databases.

One of the fundamental problems in forming general multibody dynamic model is the fact that multibody data model does not have standardized form. Semantic data model could act as that standard.

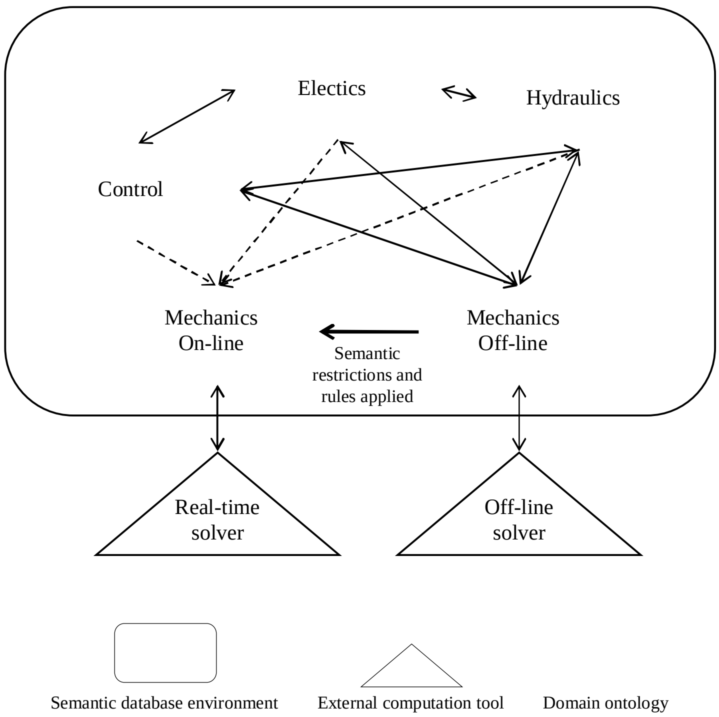

Figure 5: The principle of using semantic data model for integrating off-line and real-time simulation models.

6 Conclusions

In this work, automatic composition of a real-time multibody system model from an off-line model and the challenge of model data management have been studied and presented. In addition, previous studies and ideas are examined and presented. Managing scattered modelling data is already a problem and the problem will become more complex in the future, when modelling and simulation are applied more widely. Automating real-time simulation model composition from existing off-line simulation model is essential for making simulation-based design process more efficient and attractive.

Previously presented studies about document-based transfer formats for multibody system models are still valid and useful in several particular cases, but for large and complex simulation models, and while looking for more general solution, they are insufficient. In addition, document-based systems do not give full answer for automatic composition of real-time simulation model. The previously presented general models, which are based on relational databases, are not flexible and efficient enough for handling data with non-predefined structure.

It seemed that a general model that is based on a database could be the solution for modelling data management, but the database should be specially designed to handle modelling data. With semantic rule sets, it might be possible to automatically compose a real-time multibody system model from an off-line model. More precisely, semantic database may combine solutions for both problems, i.e. automatic composition of a real-time simulation model and managing complex modelling data, into one solution. Implementation of a semantic database and domain ontologies is worth of further research.

Acknowledgements

This report is produced in research project *Computational models in product life cycle – Codes*. The project is funded by Tekes (the Finnish Funding Agency for Technology and Innovation) and the participating companies.

References

[1] W. Schiehlen. Multibody System Dynamics: Roots and Perspectives. *Multibody System Dynamics*, 1(2):149–188, 1997.

[2] H. Ma, H. Johansson, and K. Orsborn. Distribution and Synchronisation of Engineering Information Using Active Database Technology. *Advances in Engineering Software*, 36(11-12):720–728, 2005.

[3] C. Tisell and K. Orsborn. A System for Multibody Analysis Based on Object-Relational Database Technology. *Advances in Engineering Software*, 31(12):971–984, 2000.

[4] M. González and J. Cuadrado. Mbslab: a New Collaborative Simulation Environment for Multibody System Analysis. In *Proceedings of ACMD 2004*, 2004.

[5] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, F. Yergeau, and J. Cowan. Extensible Markup Language (XML) 1.1 (Second Edition). W3C Recommendation, 2006.

[6] A. Siemers, D. Fritzson, and I. Nakhimovski. General Meta-Model Based Co-Simulations Applied to Mechanical Systems. *Simulation Modelling Practice and Theory*, 17(4):612–624, 2009.

[7] Website of Dymola: http://www.3ds.com/products/catia/portfolio/dymola (cited March 4, 2010).

[8] A. Shabana. *Dynamics of Multibody Systems*, volume ISBN 0-521-59446-4. Cambridge University Press, Cambridge, 1998.

[9] L. Meirovitch. *Elements of Vibration Analysis*, volume ISBN 0-07-041340-1. McGraw-Hill, New York, 1975.

[10] S. Kim and E. J. Haug. A Recursive Formulation for Flexible Multibody Dynamics, Part I: Open-Loop Systems. *Computer Methods in Applied Mechanics and Engineering*, 71(3):293–314, 1988.

[11] J. H. Critchley and K. S. Anderson. A Parallel Logarithmic Order Algorithm for General Multibody System Dynamics. *Multibody System Dynamics*, 12(1):75–93, 2004.

[12] J. Cuadrado, J. Cardenal, and E. Bayo. Modeling and Solution Methods for Efficient Real-Time Simulation of Multibody Dynamics. *Multibody System Dynamics*, 1(3):259–280, 1997.

[13] Website of Simantics: https://www.simantics.org/ (cited March 4, 2010).