INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED03 STOCKHOLM, AUGUST 19-21, 2003

LINKING ANALYTICAL TARGET CASCADING TO ENGINEERING INFORMATION SYSTEMS FOR SIMULATION-BASED OPTIMAL VEHICLE DESIGN

M. Nyström, T. Larsson, L. Karlsson, M. Kokkolaras, and P.Y. Papalambros

Abstract

An environment for simulation-based analysis, design, and product development is presented in this article. Specifically, we integrate the analytical target cascading (ATC) methodology for optimal design with the Active Mediator Object System (AMOS) II information management tool to represent and exchange engineering data in distributed and heterogeneous environments. The ATC process is applied to the design optimization problem of a vehicle model with many design variables. The vehicle model is decomposed into system models, and vehicle design targets are translated to system specifications. The ADAMS software is used to develop the necessary simulation models. AMOS II is used to enable the data exchange among the vehicle dynamics simulation software that is distributed on different computers connected by a network.

Keywords: Information system, database management, mediator technology, analytical target cascading, optimal vehicle design, simulation models

1 Introduction

Customer needs, marketing objectives, and legislation requirements drive the automotive industry to develop more comfortable, safe, and light-weight vehicles that consume less fuel, produce less emission gases, and have longer lifecycles with improved maintainability. At the same time, product quality has to be preserved, if not increased, to ensure competitive advantage. These goals are even more challenging to achieve considering the necessity of constantly reducing the lead-time as well as design, production, product, and maintenance costs. To fulfil all these demands and pave the way for further improvements, the product development methodology has to rely on an effective, dynamic, flexible, and robust design platform. Several different analysis disciplines need to be integrated within an engineering information environment for analysis and simulation, concept evaluation, optimisation, verification. Such an environment will enable the designer to aim at satisfying vehicle design targets by determining consistent designs, identifying limitations of different concepts, and selecting the best overall solution early in the product development process. So-called integrated virtual product development (IVPD) environments, or virtual proving grounds (VPG), make it possible to “do it right the first time” and avoid late and costly design iterations.

As the number of design variables and constraints (problem size) increases with increasing complexity of simulation models, large-scale optimizations are impossible to conduct due to convergence difficulties. Optimal vehicle design can therefore be accomplished only by decomposing the problem into smaller, more tractable subproblems. In simulation-based design optimization, the partitioning of the vehicle is dictated by the availability of models. Analytical target cascading (ATC) is a methodology for translating overall product design targets to specifications of systems and subsystems [1]. Optimization problems, associated with systems, subsystems, components, etc. in a multilevel, hierarchical, model-based decomposition, are formulated to minimize deviations from propagated targets subject to consistency constraints. The propagation of targets is achieved by establishing a functional dependency among elements in successive levels. That is, outputs of lower level elements are inputs to higher level elements. In this manner, once the optimal values of optimization variables are determined by solving a subproblem, they become response targets for the linked subproblems at the level below. The subproblems are solved iteratively until the process converges to a consistent design. ATC aims at avoiding design iterations in the later stages of the product development process, and facilitates concurrent engineering and outsourcing. Vehicle design case studies have demonstrated the usefulness of ATC [2], [3]. Global convergence of the ATC process has been proven under standard convexity and smoothness assumptions [4]. A rigorous framework for specifying and implementing the coordination process of solution of the subproblems by means of a computer science tool has been proposed in [5].

Design optimization of large and complex systems is also difficult due to the computational environment, which is characterized by distributed and heterogeneous applications that need to exchange data. The implementation of an engineering information system supporting this task enables the use of simulation-based optimization for the design of dynamical systems. The system is developed using the extensible and main-memory resident database management system AMOS II [6]. AMOS II is based on the mediator approach, which assumes a computer environment with a number of workstations connected through a high-speed network. The mediator system is a layer that mediates data between applications and data sources by providing methods to query, monitor, transform, combine, and locate data. The AMOS II data model is an object-oriented extension of the functional data model, and uses objects, types, and functions. Database technology [7] is intended to simplify the development of data intensive applications and traditionally this technology has been used mostly in business applications with simple data structures and many, yet simple, transactions.

The purpose of this paper is to demonstrate that optimal design methodologies, such as ATC, can be integrated with information systems necessary to represent and exchange engineering data in distributed and heterogeneous environments to formulate an integral approach to simulation-based optimal design and product development. ATC is applied to the design optimization problem of a vehicle model with many design variables. The vehicle model is decomposed into system models, and vehicle design targets are translated to system specifications. The ADAMS software is used to develop the simulation models. AMOS II is used to exchange data among vehicle dynamics simulation software that resides in a distributed computing network. AMOS II has been used earlier for the integration and management of engineering data in various research projects [8], [9], [10], [11]. Even though data are stored in it, AMOS II is not primary used for storage. Its integration capabilities, data modeling, and query language provide the ability to create a virtual database that consists of integrated data sources. Object-oriented modeling is used to model the data in the analysis process and the functionality offered by the vehicle dynamics simulation systems. The functionality in an external system is accessed through functions with specific arguments and results. When such a function is invoked, an external procedure will perform the operations and return the results to the database. Such a function could, for example, start a vehicle dynamics analysis at a remote computer. The ATC application uses high-level declarative queries to perform operations on the data. A declarative query specifies what should be retrieved without defining the execution order. The execution order of the query is determined by the database, which also manages the memory. The information system features a completely automated exchange of information.

2 Multibody System Analysis Domain

A multibody system consists of rigid bodies that are connected to each other by joints that restrict their relative motion. Analyzing these relative motions under the influence of forces is referred to as multibody dynamics. The Multibody System Dynamics (MBS) methodology for performing analysis in the integrated product development process is illustrated in Figure 1. MBS simulations (using different models) are carried out at different stages of the development process with different types of objectives. The MBS process, however, is similar independent of the level of detail in the model. Rapid modeling, low detail models and inexpensive simulations are desirable in the early stages to quickly evaluate different concepts. In these phases the quality of information is low and the time frame available for decisions limited.

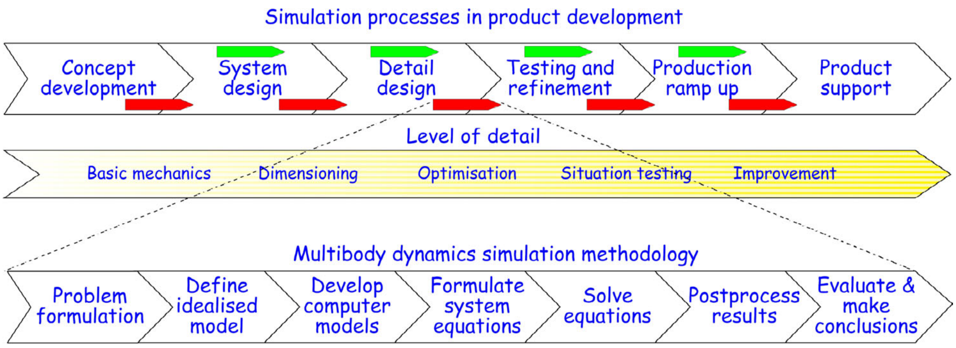

Figure 1. MBS analysis methodology in product development.

At later stages the product is modeled with higher level of detail, and more accurate analyses by means of higher-fidelity simulations are increasingly important. In these phases higher-quality information is often available in form of experimental data from prototype testing. This enables validation of the simulation models. After validation the simulation model can be used to predict the behavior of the system under different conditions.

The specific MBS analysis of a vehicle system is characterized by Haug as the following sequence of steps [12]:

- Modeling the multibody system
- Deriving the equations of motion
- Simulating the trajectories of the generalized coordinates
- Animating the vehicle system by moving pictures
- Evaluating the dynamical performance by adequate criteria

A vehicle model of the Volvo V70 is considered in this work. The simulation models, depicted in Figure 2, were built in ADAMS/Car [13].

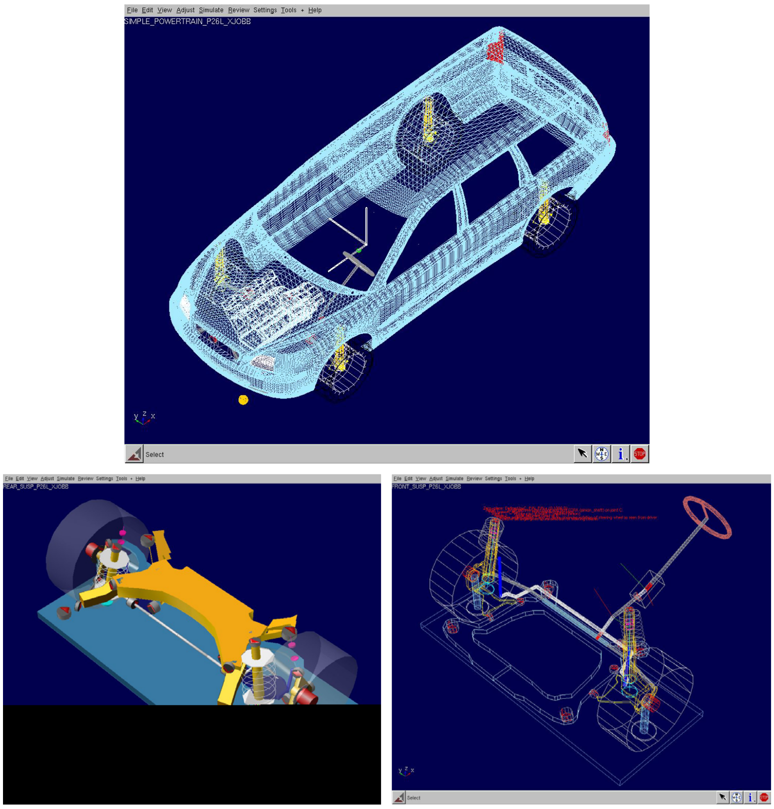

Figure 2. Full and partial simulation models of the Volvo V70.

The motivation behind this type of vehicle dynamics analysis is the prediction of ride comfort, handling, and stability [14]. The aim is to find an optimal parameter set for prescribed excitations, such as asymmetric or symmetric pothole, rough road, passing over a cleat, and J-turn maneuvers.

3 Mediator System Overview

Active Mediator Object System II (AMOS II) [6] has been used as a framework to create an engineering information system that wraps the dynamic simulation software ADAMS [13]. AMOS II is an object-oriented, main-memory resident, extensible mediator system that uses the mediator wrapper approach to information integration [15]. In addition, AMOS II is a multidatabase system where different mediators can communicate with each other remotely. The purpose of data mediation is to provide applications with a virtual view, or a collection of views, of the data in different integrated data sources. Thus, data mediation creates a virtual database where data are available for computations. A view is a representation of data that does not exist physically but can be queried as if it did exist physically. For example, a view can be declared as a query. When such a view is used, the query that declares the view will be executed, and relevant data will be made available to the view. When dealing with a mediator, a view that integrates data is declared as stored in a data source. When queries are posed over this view, the mediator will fetch the relevant data in the data source in order to provide the results of the query. Since different types of data sources may be present, a wrapper is necessary to translate queries between the mediator and the data source. This mediator wrapper approach divides the information system into three distinct layers: the data source layer, the mediator layer, and the application layer, illustrated in Figure 3.

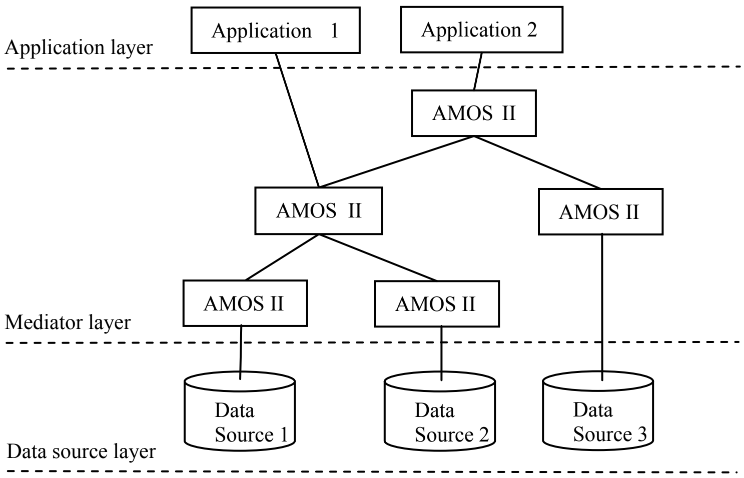

Figure 3. Mediator approach to information integration.

AMOS II uses the object-oriented data definition and query language AmosQL. Objects can be literals or surrogates. Literals consist of basic types such as integers, reals, and strings. Surrogates have object identifiers (OID) that are maintained by the system. An object is an instance of a specific type and each type has an extent which consists of all object instances that belong to the type. AMOS II supports multiple inheritance; the types are organised in a hierarchy of subtypes/supertypes and are stored locally in the mediator. Proxy types represent types that are stored in another mediator database, integrated data source, or system. Derived types and integration union types are used for data integration purposes to resolve different heterogeneities [6].

Functions are used to model properties of objects, relationships between objects, and operations on objects. Functions can have multiple arguments and results. AMOS II functions are stored functions, derived functions, foreign functions, database procedures, and proxy functions. Stored functions are used to model attributes. Derived functions are defined in terms of object-oriented queries. Foreign functions are functions defined in an external programming language such as C or Java, and are typically used to extend the database with functionality that can not be expressed in AmosQL. Database procedures are functions that have a procedure body defined in AmosQL; they are normally used for updates to the database.

4 Linking ATC to the Engineering Information System

The AMOS II mediator system has been used to link ATC to an engineering information system that manages the simulation software ADAMS and its data. Following the concepts of the mediator wrapper approach to data integration, ADAMS is considered as a data source present in the data source layer. ATC has been implemented as an application to the engineering information system and uses queries to find the proper information. The mediator provides the ATC application with a specialised view of the information the application needs. The intention is to provide a general view of simulation data that is independent of what simulation system that has been wrapped by the mediator. Thus, the ATC application can be developed using this general view and does not need to be updated if changes occur, e.g., in the interface between the mediator system and the simulation system. It also allows the ATC application to access data from various wrapped simulation systems using the same view. The ADAMS simulation software has been wrapped, making its data and computations available through the mediator system. The ATC application has also developed to use the information made available by means of simulations in the mediator system. The three different layers of the engineering information system are presented in the following sections.

4.1 The ADAMS Layer

The ADAMS simulation software consists of the pre-processor, the solver, and the post-processor used to build the simulation models, solve the equations, and analyse the responses. Typically, engineers run the software interactively by means of a graphical user interface (GUI). The simulation process required in this work is automated: a number of simulations are performed automatically with different sets of design variables. Therefore, an interface to ADAMS has been developed using J2EE Technology [16]. The interface provides functionality to update data, remote execution of solver, and extract results from the simulations. An Object Request Broker (ORB) process is set up on a number of workstations [17]. The ORB is able to answer to incoming Internet Inter-ORB Protocol (IIOP) messages. When a message is received the ORB activates a Java program that implements the function on that specific host. Hence, different functions in the Java program can be executed and used to update data, execute the ADAMS solver, and extract data [18]. ADAMS data can be modified through the ORB in a relatively simple manner since it consists mainly of different design variable values of the type real. Data extracted from the simulation results consists of several large vectors that contain accelerations, velocities, etc. in different time steps.

4.2 The Mediator Layer

The mediator layer is used to integrate the information and to provide high-level interfaces to the application layer. Earlier work has dealt with methods for integrating data sources available through a general IIOP wrapper [8]. This IIOP wrapper has been utilised to enable communication between the AMOS II mediator system and the ADAMS simulation software. A view to ADAMS has been generated in the mediator system, over which it is possible to use queries. The queries to the view are translated into IIOP messages that are sent remotely to the ORB that wraps ADAMS. The technique enables remote updates, remote execution of ADAMS, and remote access to simulation data. Simulation responses are manipulated using the representations and operations described in [19]. Intermediate results of the analyses are stored locally in the mediator layer for processing by the ATC application.

4.3 The ATC Layer

ATC has been implemented as an application that uses a specialised external view of the simulation data provided by the mediator layer to access the relevant data. In this section we give an insight into this view, illustrated in Figure 4, and explain how the data is queried from the ATC application. For each type of simulation we define design variables and their bounds, and generate a set of different combination values of these variables by means of a design of experiments. We then execute a simulation with each combination of design variable values as input to generate a response surface, which we finally use during the optimization process to predict simulation responses. Design variables are stored in the type *DV* in the mediator database; each collection of design variables is associated with a simulation through the relationship *dvs*. The type *DV* has the attributes *name* and *value*, used to store the name and the value of the design variable, respectively. The type *Simulation* has the attribute *name*, where a name describing the analysis is stored. The results from a simulation are stored in the type *Response* which has the attributes *name* and *value*, used to store the name and the value of the response, respectively.

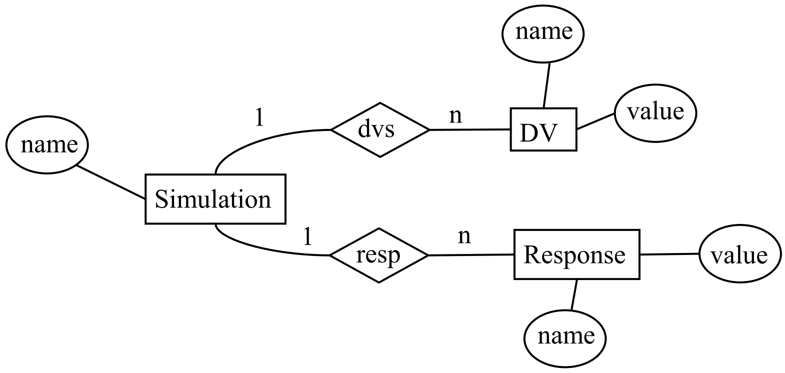

Figure 4. ATC view of simulation data.

The ATC application uses this view for evaluation of the simulation results. The simulations performed on the Volvo V70 generate results at both the vehicle level and the system level. The ATC application performs a search among the generated results different conditions needs to be fulfilled. The example query (1) in AmosQL shows how the ATC application gets the minimum value of the response from a number of simulations. In order to get the value a number of conditions need to be fulfilled. The first condition matches the value of the design variable at the complete vehicle level with the value of the response at the system level. The next condition selects the proper simulations at the vehicle level and the system level. Finally, the simulation with the lowest value of the response that fulfils all the conditions is selected among the complete vehicle simulations *cv* using the function min.

```
SELECT min(value(resp(cv)))
FROM simulation cv, simulation sr, dv cv_dv, response sr_r
WHERE  value(cv_dv) =  value(sr_r)
AND cv_dv = dvs(cv)
AND name(cv_dv)=”rear toe fy”
AND name(cv)=”complete vehicle”
AND sr_r = resp(sr)
AND name(sr_r) = “rear toe fy”
AND name(sr)= “rear  suspension”;                                  (1)
```

Note that this is an example query for demonstration purposes; it has therefore been simplified. The real query would not search for an exact match; inequality conditions would be used instead. In addition, queries that are specific to the ADAMS models need to be executed to materialise these simulations necessary for evaluation as described in Figure 4. In the normal case, a graphical user interface that presents data and provides a number of predefined queries to execute would also be made available to the user.

5 Performance Simulations of the Volvo V70

The objective of the simulations is to study the handling performance of the Volvo V70, and find the geometrical settings and bushing characteristics (present at the system level) that satisfy a desired target function (present at the vehicle level). The vehicle has been partitioned according to a bi-level hierarchy consisting of a vehicle level and a system level. The complete vehicle is modeled at the vehicle level for simulating handling performance, while the front and rear suspensions are modeled at the system level. Figure 2 illustrates the complete vehicle model (top), the detailed front suspension (bottom right), and the rear suspension (bottom left).

We have chosen to study the yaw velocity overshoot, which is a measure of the handling characteristics during rapid steering maneuvers. The target function at the vehicle level will consist of these characteristics. The characteristics are known to be conflicting and we need to find the optimal compromise. The yaw stability simulation is performed in two steps. In the first step the steering wheel angle is determined to yield a lateral acceleration of 4 *m/s* when the speed of the vehicle is 80 *km/h*. In the next step the steering wheel angle should be increased in steps to a magnitude equal to four times the initial steering wheel angle or until instability occurs. The responses from the simulation are the side slip angle, yaw velocity, the lateral acceleration, and the vehicle velocity. These responses are used to calculate the yaw velocity overshoot, which is included as the first term in the objective function at the vehicle level. The side slip angle constitutes the second term of the objective function at the vehicle level. The design target is to minimize both terms of the objective.

Design variables at the vehicle level consist of the suspension curves for the toe angle as a function of jounce travel and the toe angle as a function of the lateral force. The first curve is controlled by one coefficient that determines the gradient of the curve for the rear suspension and three coefficients describing the curve gradients for three different steering angles for the front suspension. The second curve is controlled by one coefficient determining the gradient of the curve for each suspension. Therefore, the total number of variables at the vehicle level is six. At the system level simulations are performed to determine the characteristics of the front and rear subsystem suspension when different forces and moments are applied in different directions. The responses of the simulations are a number of different curves describing the characteristics of the suspension. Two of these curves are used as design variables at the vehicle level as described above. The design variables consist of geometrical points and bushing stiffness that need to satisfy a number of design inequality constraints.

The results of the simulations indicate that the bushing stiffness at the system level do not affect the yaw velocity overshoot as much as the geometrical design variables of the suspensions. The results show that the torsion rod inner and the torsion rod outer of the rear suspension should be moved upwards in order to minimise the yaw velocity overshoot. The main scope of this article is to demonstrate the capabilities and usefulness of the developed information system on a “real-world” application using the very same simulation models used by a car manufacturing company such as Volvo Car Corporation when designing a vehicle. Future work will focus on the application and the presentation of more elaborate results.

6 Conclusions

The developed information system allows the linking of optimal design methodologies such as ATC to dynamical simulations such as ADAMS by means of AMOS II. This environment is developed by using a declarative high-level language that provides flexible means to search for an optimal design. Hence, challenging issues of memory management, execution order, and computing distribution are addressed by the mediator system. Moreover, this integration approach features object-oriented modeling capabilities and a query language used for the management of the intermediate data in the mediator layer. The technique used for wrapping the ADAMS system has proven to be sound. The information management system has been tested on models provided by Volvo Car Corporation, and standard handling simulations were performed. The simulation results indicate that the developed system have a good potential to improve the vehicle dynamic simulation process, by providing ATC, efficient management of simulation data, and remote access to a simulation system.

References

[1] Kim H.M., "Target Cascading in Optimal System Design'', PhD thesis, Department of Mechanical Engineering, University of Michigan, Ann Arbor, Michigan, 2001.

[2] Kim H.M., Kokkolaras M., Louca L.S., Delagrammatikas G., Michelena N.F., Filipi Z.S., Papalambros P.Y., Stein J.L. and Assanis D.N., "Target Cascading in Vehicle Redesign: A Class VI Truck Study'', International Journal of Vehicle Design, Vol. 29(3), 2002, pp.1-27.

[3] Louca L.S., Kokkolaras M., Delagrammatikas G., Michelena N.F., Filipi Z.S., Papalambros P.Y. and Assanis, D.N., "Analytical Target Cascading for the Design of an Advanced Technology Heavy Truck", Proceedings of the ASME International Mechanical Engineering Congress and Exhibition, November 17-22, 2002, New Orleans, Louisiana, paper No. IMECE-2002-32860.

[4] Michelena N.F., Park A. and Papalambros P.Y., "Convergence Properties of Analytical Target Cascading'', Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, September 4-6, 2002, Atlanta, Georgia, paper no. AIAA-2002-5506, to appear in the AIAA Journal.

[5] Etman L.F.P., Kokkolaras M., Papalambros P.Y., Hofkampf A.T. and Rooda J.E., "Coordination Specification of the Analytical Target Cascading Process Using the Chi Language", Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, September 4-6, 2002, Atlanta, Georgia, paper No. AIAA-2002-5637.

[6] Risch T. and Josifovski V., ``Distributed Data Integration by Object-Oriented Mediator Servers'', Concurrency and Computation: Practice and Experience, Vol. 13, 2001, pp.933-953.

[7] Garcia-Molina H., Ullman,J.D. and Widom J., "Database Systems: The Complete Book", Prentice Hall, 2002, ISBN 0130319953.

[8] Nyström M., "Engineering Information Integration and Application Development using Object-Oriented Mediator Databases", Department of Applied Physics and Mechanical Engineering, PhD thesis, Luleå University of Technology, 2003:04.

[9] Orsborn K., "On Extensible and Object-Relational Database Technology for Finite Element Applications", Department of Computer and Information Science, Linköping Institute of Technology, 1996, ISBN 91-7871-827-9.

[10] Koparanova M. and Risch T., “Completing CAD Data Queries for Visualization”, International Database Engineering and Applications Symposium (IDEAS), Edmonton, Alberta, Canada, July 17-19, 2002.

[11] Johansson H., “Conceptual Information Models to Integrate Data Management in Engineering Simulation”, Department of Applied Physics and Mechanical Engineering, PhD thesis, Luleå University of Technology, 2002:33.

[12] Haug E.J. (ed.), “Concurrent engineering: Tools and technologies for mechanical system design”, NATO ASI Series, Series F: Computer and System Sciences, Vol. 108, Springer-Verlag, Germany, 1993.

[13] MSC.ADAMS, http://www.adams.com/.

[14] Schalk Els P. and Van Niekerk J. L., “Dynamic Modelling of an Off-Road Vehicle for the Design of a Semi-Active, Hydropneumatic Spring-Damper System”, Vehicle System Dynamics Supplement, Vol. 33, 1999, pp.566-577.

[15] Wiederhold G., “Mediators in the Architecture of Future Information Systems”, IEEE Computer, Vol. 25(3), 1992, pp.38-49.

[16] JavaTM 2 Platform, Enterprise Edition (J2EE), http://java.sun.com/j2ee/.

[17] Siegel J., “CORBA: Fundamentals and Programming”, Wiley, 1996, ISBN 0471121487.

[18] Nergård H., “Conceptual Suspension Modelling in Automated Vehicle Dynamics Simulations”, Master thesis, Luleå University of Technology, 2003:36, 2003.

[19] Nyström M. and Orsborn, K., “Computational Database Technology for Component Mode Synthesis”, Proceedings of The Sixth International Conference on Computational Structures Technology, Prague, Czech Republic 4-6 September, 2002.

Acknowledgements
The authors would like to thank Jan Hellberg and Lars Lidner of Volvo Car Corporation for providing models and knowledge on the vehicle dynamics simulation process, and Henrik Nergård of Luleå University of Technology for his help with the ADAMS software.

Corresponding author:
Mattias Nyström
Luleå University of Technology, Sweden
Address: Luleå University of Technology, SE-97187 Luleå, Sweden
Tel: +46 (0)733 107568
E-mail: mattias.nystrom@cad.luth.se
http://www.cad.luth.se