

DETC01/CIE-21282

**DISTRIBUTED MULTIBODY DYNAMIC ANALYSIS WITHIN
PRODUCT DEVELOPMENT**

Tobias Larsson

Division of Computer Aided Design
Department of Mechanical Engineering
Luleå University of Technology
SE-971 87 Luleå, Sweden
E-mail: Tobias.Larsson@cad.luth.se

Andreas Larsson

Division of Computer Aided Design
Department of Mechanical Engineering
Luleå University of Technology
SE-971 87 Luleå, Sweden
E-mail: Andreas.Larsson@cad.luth.se

Lennart Karlsson

Professor
Division of Computer Aided Design
Department of Mechanical Engineering
Luleå University of Technology
SE-971 87 Luleå, Sweden
E-mail: Lennart.Karlsson@cad.luth.se

Keywords: Distributed engineering, multibody dynamic analysis, product development

ABSTRACT

A method that supports distribution of multibody dynamic analysis is proposed and developed. Ethnographic methods are used as a means for gaining a deeper understanding of the engineering analysis work practice, and the findings form the base for a cooperative design of the system. The concept of design rationale is applied in order to deal with current problems of engineering analysis, such as irrelevant input and output, as well as satisfying the need for useful, real-time feedback. The web based simulation environment, applied to vehicle system dynamics, contains the simulation packages ADAMS and MATLAB and incorporates database technology. The possibility to distribute simulation models and results, from simulation experts to design experts and engineers, as well as subcontractors, is created. The proposed method requires changes in the existing multibody dynamic simulation methodology regarding aspects of incorporation in product development theories.

INTRODUCTION

The challenge for industry today is to arrive at a product design that has been systematically optimised according to customer needs. Products must hit the market faster, more often, at lower development costs, while still meeting demands on quality and performance [1,2]. Consequently, specialised disciplines have to be integrated in a simultaneous, or concurrent, engineering process where all the relevant activities, such as design, analysis, testing, and manufacturing are involved and coupled at an early stage of the development process [3,4].

Simulation techniques of mechanical dynamic systems have a large potential in product development but are only partly used today [5]. This is due to problems like modelling complexity, software immaturity and user inexperience. Even though computer-based tools for modelling and simulation have rapidly changed the best practice of product development, simulation of multibody dynamics is far from a product development support used throughout the whole company today. In order to make it possible for a design engineer, and not only for a specialist in dynamics, to use simulation methods

for development of dynamic systems, the complexity of the methodology for using them must be reduced.

The last decade's rapid development within Internet and network technology, where almost any computer can be linked to the Internet, has created new dimensions in terms of product development. Successful companies will get the right information and tools to the right person at the right time, regardless of where the person is located [6]. The idea of sharing computer-aided design services in a computer-network oriented design environment has been addressed by several researchers during the past couple of years [6-11].

With its foundation in empirical findings, forming the problem definition, a method, or tool, that supports web based engineering analysis is proposed and developed in this work. The aim is to improve product development by using multibody dynamic simulation (MBS) earlier and in more phases of the product development process (PDP). An infrastructure is developed and a prototype system within vehicle system dynamics is implemented to illustrate a framework that provides dynamic analysis services over the Internet.

DESIGNING FOR USABILITY

In order to build a tool that truly supports engineering analysis, i.e. designing for usability, design decisions need to be anchored in the real work activities of the users – not in rigid, written specifications based on preconceived notions about the actual work activity. System developers, in general, try to understand the work of users by applying their own concepts of efficiency, reliability, usability and so forth, to an imagined work practice, which they have little or no knowledge about [7]. Since users and developers do not share a common practice, or a common language, a communication breakdown is likely to occur somewhere along the line, usually ending up with the users having to adapt to the system instead of the other way around.

Empirical work

Qualitative research methods [8], such as participatory observation, written field notes and informal conversations have been used as a means for gaining a deeper understanding of the engineering analysis work practice that the tool is supposed to support. The subjects in our study are mechanical engineers with differing skills and experience within the field of dynamic analysis. The overall aim of the study was to find out how engineers *really* work with tools for dynamic analysis, and how the computer tools that they have chosen to use support their work activities. Engineers have a variety of software tools available for performing dynamic analysis. Since the ADAMS [9] software package is a widely known and used tool for dynamic analysis, we concentrated on studying ADAMS users. We defined these users as *beginners*, *intermediate users* or *experts*, with respect to both their professional experience of using the tool, and to the different levels of software functionality that they have used. Three users from each category were involved in the study.

The empirical study clearly shows that performing dynamic analysis is not solely about humans and computers exchanging unambiguous information through input and output. The interplay between human and computer is of great importance to the end result. Dynamic analysis involves creating a meaningful interaction where the computer is used to guide the engineer through a complex translation phase - going from internal and external models to a virtual representation of the product being developed. The level of meaningfulness depends largely upon the ability of both humans and computers to act and react in ways that are easily understandable by the other part.

Work-of-the-work vs. Work-of-the-tool

The activity of dynamic analysis also involves activities, which easily could be defined as bottlenecks in product development. The engineers are exposed to constant strain, which more often is related to tool-oriented problem solving, i.e. the software, than to work-oriented problem solving, i.e. engineering analysis. One of the experts on ADAMS said that he and his colleagues often talk about *pushing through* or *sweating out* an analysis, which clearly shows that performing advanced dynamic analysis is generally considered a very difficult and time-consuming process. In fact, all the engineers in the study agreed that there is no such thing as just *working-the-work* [10] when doing dynamic analysis. To put it another way; *tool-related problems are so common that it is hard to tell the difference between working-the-work and working-the-tool.*

Ethnographic findings

There are a couple of main findings to consider in the design of a simulation supporting system. These findings are divided into three categories – *input*, *output* and *feedback* - in order to highlight the fact that engineers have to deal with bottleneck issues at several stages of the activity.

Input

- *Information overload* occurs when the user has to consider multiple options for input, some of which are completely irrelevant for the task at hand, and some of which the user has no knowledge about.
- There is a lack of *natural mapping* between the way the user wants to do things, and the options and sequence of steps that the system suggests.

Output

- *Information overload* occurs when the user has to take multiple answers into account, some of which are completely irrelevant for the task at hand, and some of which the user has no knowledge about.
- There is lack of *natural mapping* between the way the user wants to have the results, and the way that the system provides the result.

Feedback

- The way that the studied system provides feedback is not efficient enough. Trial and error is still the best way to get useful feedback. There is a lack of both real-time feedback and feedback which provide not only warnings or errors, but also suggest ways to deal with these problems.

The common denominator is that the system forces the users to let go of the main thread - stepping out of their *engineering mindset* - only to handle the tool, which actually was supposed to function as a support in their engineering work. They are continuously using parts of their intellectual capacity to think about and act according to events, which do not really add quality to the product they are developing.

It seems reasonable to assume that the activity of performing dynamic analysis will work better with a system that helps the users focusing on the relevant issues of their work, without having to put too much effort into activities which, honestly, should be automatic processes that occurs without conscious *thinking*. Thought and reflection seem to be more powerful when applied to problems that are relevant to the main activity, which, in this case, is *performing dynamic analysis*, not *using tools* for performing dynamic analysis.

WEB BASED SIMULATION TOOL

A distributed MBS analysis environment, eMBS, according to the basic client-server idea in Figure 1 has been developed, with its application within the area of vehicle system dynamics.

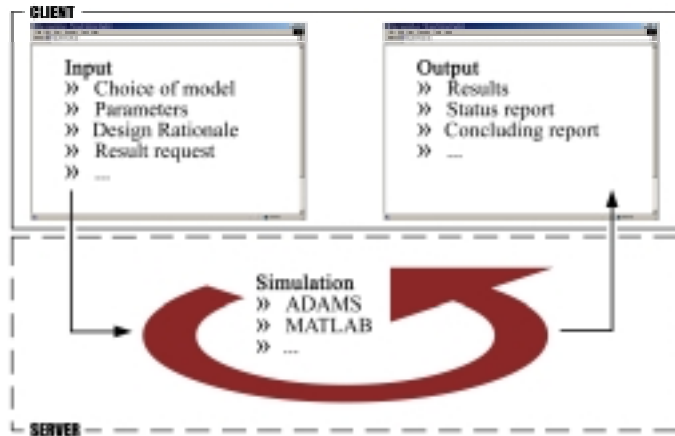


Figure 1. The idea of distributed MBS analysis

The approach is based on the idea that the end user/client chooses a model and corresponding values for input parameters and what results are needed as output and then submits the analysis. The underlying server side simulation software performs all the numerical integration and generation of desired results. By using server side software and a web client, the need for client hardware and software resources are reduced, creating an easier way to access simulation capabilities.

The MBS application

A multibody system is a system that consists of solid bodies, or *links*, that are connected to each other by *joints* that restrict their relative motion. The study of multibody dynamics is the analysis of how mechanism systems move under the influence of forces. The MBS analysis methodology for performing analysis in the PDP is seen in Figure 2. Simulation processes are carried out at different stages of the development process and with different types of objectives and simulation models. The methodology for MBS simulation however is similar independent of the level of detail of the model. In the early stages fast modelling and comparative analyses are crucial in order to evaluate different concepts. The accessible information is often insufficient and the time frame for decisions limited.

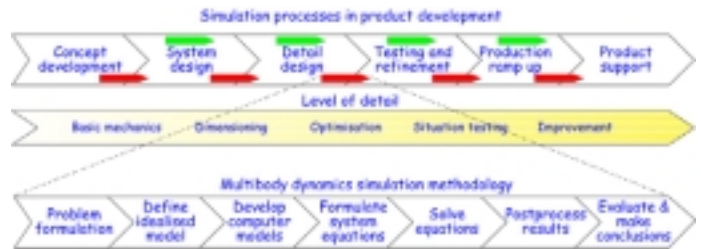


Figure 2. MBS analysis methodology in PD.

At later stages the product is defined with higher level of detail, and accurate analyses are increasingly important. In these phases, better information is often available in form of experimental data from prototype testing. This makes it possible to verify the developed simulation models. After verification, the simulation model can be used to predict, if dealt with in a correct way, the behaviour in other situations than the experimentally tested ones. If simulation constraints are violated, according to design rationale, the results are no longer valid.

The specific MBS analysis of a vehicle system is characterised by Haug [3]:

- Modelling of a multibody system.
- Generating the equations of motion.
- Simulating the trajectories of the generalised coordinates.
- Animating the vehicle system by moving pictures.
- Evaluating the dynamical performance by adequate criteria.

The vehicle system application in this work is a front suspension to a car, a quarter model. The simulation model, built in ADAMS, is shown in Figure 3.

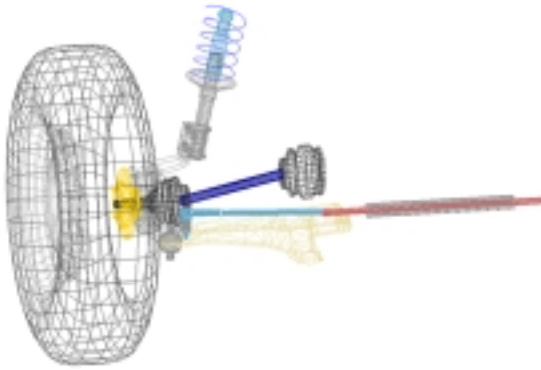


Figure 3. Quarter model of an automotive front suspension.

The motive for analysis of this type of mechanical system is the prediction of ride comfort, handling, and stability [11]. 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 manoeuvre.

Simulation procedure in eMBS

The simulation software expert is performing the whole MBS procedure, see Figure 2, hereby creating and validating the simulation models intended for use by others than simulation experts. The user, i.e. anyone granted access to the web page seen in Figure 4, choose a model to simulate or load a parameter set for an earlier performed simulation.

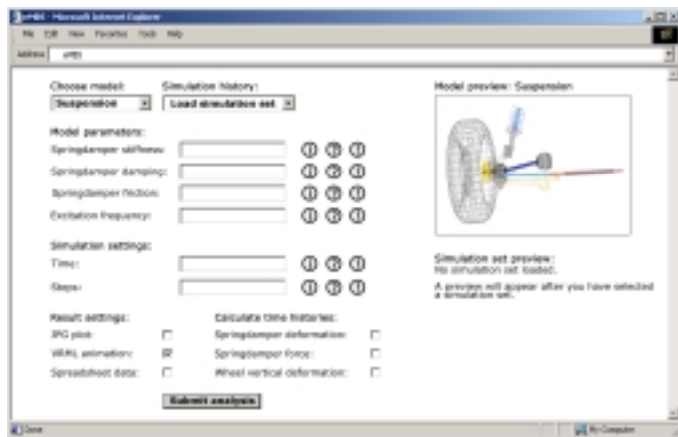


Figure 4. Input interface, simulation model of front suspension.

The user then enters parameter data for the simulation to run, decides what results are needed and starts the analysis. When the simulation is complete, the results appear in a web page according to Figure 5. The domain experts can hereby carry out the simulations rather than having all analysis work performed by the experts with knowledge of the simulation tools. Ultimately, this is leading to that decisions can be made

by those with actual domain knowledge and without the need for continuous discussions with the analysis department.

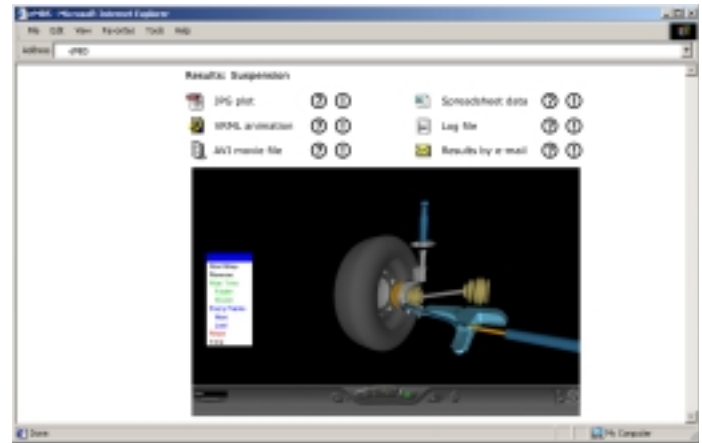


Figure 5. Output interface, with VRML result displayed.

Design Rationale

The activity of distributed analysis has to be facilitated by helping the user understand the underlying reasons for design, i.e. assisting the user in understanding the functionality of the tool as well as how to use the simulation model. In this respect, we are also designing to facilitate learning. Since the user is not an expert in performing analysis, there are obvious advantages with an application that can mediate some of the knowledge and expertise that the simulation expert possesses. A distributed system for dynamic analysis is not just about distribution in time and space; it is also about distribution of knowledge.

Design rationale [12] is a concept that on many accounts represents this view on *knowledge distribution*. It is an excellent way to give the users access to a richer view of the tool, and thereby also a richer view of how to use it.

Even though design rationale traditionally is used as design documentation, to be used by other designers, a similar technique can be used when creating a meaningful interaction between the user and the analysis tool. Incorporating design rationale into the eMBS system will help the users to an increased understanding of the tool and its functionality, since the design intent of the simulation expert/model builder is communicated, including the underlying reasons for choosing certain parameters or ranges, and any kind of information that could assist the domain expert in performing the analysis.

Graphical user interface

Web interfaces can be rendered by a wide variety of web browsers, available on almost all types of available computer systems. The familiarity of presenting information in this way increases the spectra of the audience for using the tools behind the web interface. Web interfaces require much less bandwidth than other technologies for presenting graphical user interfaces (GUI) remotely, i.e. X window systems. As a consequence this enables the distributed use to include clients with low

bandwidth, such as dial-up connections. It also provides the base for using the tool in the field on a laptop with cellular phone dial-up connection.

Web browsers are relatively mature client applications and together with the use of technologies, such as Java, Active Server Pages (ASP), or Dynamic HTML (DHTML), interactive pages can be developed granting access to whatever application or resource needed for the logged on end user.

The web interface for the eMBS system is developed in ASP and DHTML, giving interactivity to the interface, yet requiring not more than a version 4 browser.

Input interface

The input interface consists of drop-down lists for choosing simulation model and/or simulation set, parameters for the chosen model, settings for the simulation, such as time and simulation steps and result settings. A simulation model preview is also available to provide the user with a visual overview of the system. To each parameter in the model there are additional information available, implemented by the model creator; parameter information, design rationale and comments, represented by **i**, **?** and **!** in Figure 4. The number of parameters and results settings is depending on loaded model and hereby controlled by the model creator.

Result format settings contain plots (JPG images), VRML models (interactive 3D animations), movies (AVI) and spreadsheet data for access to raw data. These result settings are model dependent and may contain more information if specified by the model creator. The results to be produced in the desired formatting are checked in the “Time histories” option. Here, all the results the model creator intend to share will appear when the model is loaded. After parameters and settings are chosen the model will be executed at the click of the submit button.

Output interface

The output interface consists of results and analysis related information such as log files together with design rationale and comments, represented by **i**, **?** and **!** in Figure 5. Result plots and VRML/AVI animations are shown at the bottom of the page when requested. All results are stored at the server side until requested by the client, either by hyperlink or by request for e-mailed results.

Software tool integration

One important difference between a system supplied by an Application Service Provider and the eMBS system is the design rationale, i.e. the tailoring of an MBS tool to facilitate simulation in product development. An Application Service Provider system is basically a windowing system, and hereby it does not add or remove any functionality of the software in relation to if the software would have been installed and used locally. The eMBS system with the design rationale add value since it is adapted to the user needs and competence, see comparison in Figure 6 where the bottom right is the Figure 4 input interface.

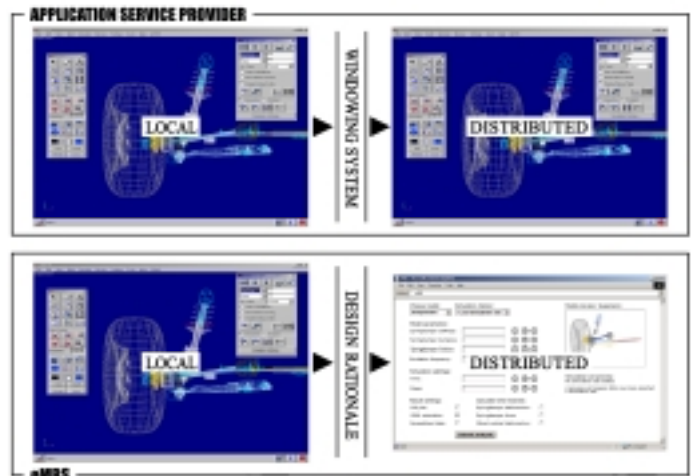


Figure 6. Application Service Provider distribution (top) and design rationale eMBS system (bottom).

The eMBS system is built of Active Server Pages (ASP), a database, the mechanical simulation package ADAMS, the scientific simulation package MATLAB [13] and a web server, according to Figure 7.

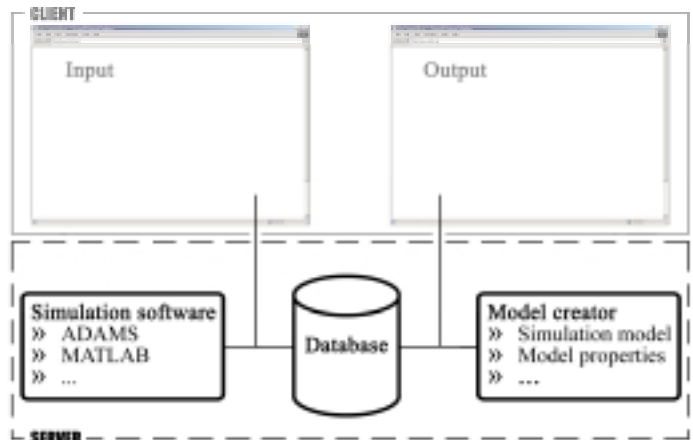


Figure 7. Tool integration.

Some additional Windows/UNIX scripts have also been developed to allow execution on both Windows and UNIX servers. The ADAMS software is mainly for mechanical simulation while the MATLAB software is for general computation efforts and for solutions of domains not implemented in ADAMS, such as control systems. The two packages are coupled to each other allowing co-simulation capabilities and hereby allowing multi domain simulation.

The model creator populates the database with simulation models and corresponding model specifications. The simulation model is a text file containing the ADAMS simulation model, including any linking to MATLAB models. The model specification file contains properties of interest for the ASP web pages, such as parameters, design rationale, parameter standard

value and range, parameter information, result components etc. The DB keeps track of the simulation model and the corresponding properties file as well as any comments generated from the end user concerning the simulation model.

The ASP web pages request information concerning available models from the DB. When a model is chosen, the DB responds with the corresponding information from the model properties file. When the analysis is executed the ADAMS/MATLAB environment is addressed from the ASP pages with simulation parameters and the DB releases the simulation model. Simulation results are tracked and organised by the database. Generation of plots, VRML animations, AVI animations, and spreadsheet data is performed by subroutines, native software commands and macros, in the ADAMS/MATLAB environment.

When the analysis is complete, the output ASP page addresses the DB for information regarding the performed analysis. Links to the results are published on the web page and when the user requires result information it is passed on.

The key to the successful system is in the mapping of web input parameters to the simulation model parameters. This is performed in the parameter mapping part of the model specification file. Another crucial task is the ASP web page addressing the simulation software and the execution of the software in a total batch mode.

DISCUSSION

By designing tools that are rooted in the work practice of the users, and by actively involving these users in the cooperative design of such a system, a work situation where engineers can concentrate on their primary work activities is facilitated. Clearly, focus shifts from the system, work-of-the-tool, to the engineering analysis, work-of-the-work. The fieldwork findings form a base for the web interface, in a what-you-see-is-what-you-need fashion, controlling the underlying simulation software. The effort is not to enhance ASP coding or ADAMS/MATLAB programming individually, but rather to see the integrated tool as a product development support tool.

Engineers that traditionally have had to rely on simulation experts are given the possibility of performing some parts of analysis themselves. Basically, they will be able to make relevant dynamic analysis earlier and more often in the process, which ultimately will leave them in the advantageous situation of increasing product quality at the same time as the time-to-market is reduced.

The system is hardware independent for the end user – only a web browser is needed - and the user can work anywhere in the world, at any hour in a thin client manner. The most evident advantage is the possibility of distributing models from simulation experts to design experts and engineers. Another advantage with the distributed system is that it allows companies to protect company specific knowledge by distributing parts of models or complete models in a black box fashion, where some parameters are available for modification, while the company sensitive parameters and structures are

protected. The approach of the modelling is similar independent of studied problem, hereby allowing for simulation model reuse.

Additionally, since companies are becoming actively involved in a larger part of their product development processes, their understanding of both the product and the process will increase.

CONCLUSIONS AND FUTURE WORK

A method that supports distribution of multibody dynamic analysis is proposed and developed. Ethnographic methods are used as a means for gaining a deeper understanding of the engineering analysis work practice, and the findings form the base for the cooperative design of the system. In order to deal with the problematic issues of engineering analysis that emerged during the empirical study - such as irrelevant input and output - as well as satisfying the need for useful, real-time feedback, the concept of design rationale is suggested and applied. The web based simulation environment, with its application within vehicle system dynamics, contains the simulation packages ADAMS and MATLAB together with a database and ASP web pages. The possibility to distribute simulation models, and results, from simulation experts to design experts and engineers, as well as within the extended enterprise, is created. The proposed method requires changes in the existing product development process since simulation support now can be added in more phases of the development process.

Future work includes more empirical work and testing, in order to establish a knowledge base for further integration into company development processes. Development of the prototype MBS system to facilitate the usage of simulation modules is also to prefer. This would enable the possibility of parallel simulation model development since a full simulation model could be combined by a number of smaller ones. A full simulation model can also be constructed by connecting modules that are distributed by subcontractors, without the subcontractors having to give away their specific product expertise.

ACKNOWLEDGEMENTS

The financial support from the Polhem Laboratory, the Swedish Foundation for Strategic Research and ENDREA is gratefully acknowledged.

REFERENCES

- [1] Wheelwright & Clark, Revolutionizing product development: Quantum leaps in speed, efficiency, and quality, The Free Press, New York, 1992.
- [2] Womack, Jones & Roos, The machine that changed the world, Rawson Associates, New York, 1990.
- [3] Haug, E. J. (ed.), Concurrent engineering: Tools and technologies for mechanical system design, NATO ASI Series,

Series F: Computer and System Sciences, **108**, Springer-Verlag, Germany, 1993.

[4] Andreasen, M. M., Hein, L., Integrated product development, IFS (Publications), Springer-Verlag, Berlin/Heidelberg, 1987.

[5] Schiehlen, W., Multibody system dynamics: Roots and perspectives, *Multibody System Dynamics*, **1**, pp. 149-188, 1997.

[6] Regli, W. C., Internet-enabled computer-aided design, *IEEE Internet Computing*, 39-50, 1997.

[7] Olsen, G. R., Cutkosky, M., Tenenbaum, J. M., Gruber, T. R., Collaborative Engineering Based on Knowledge Sharing Agreements, *Engineering Data Management: Integrating the Engineering Enterprise*, ASME 1994.

[8] Ely, M., *Doing Qualitative Research: Circles within Circles*, Falmer Press Ltd, London, England, 1991.

[9] ADAMS: <http://www.adams.com>

[10] Holtzblatt, K. and Jones, S. 1993. Contextual Inquiry: A Participatory Technique for System Design, in *Participatory Design: Principles and Practice*, ed. A. Namioka and D. Schuler. Hillsdale, NJ: Lawrence Earlbaum.

[11] Schalk Els, P., 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 33* (1999), pp.566-577.

[12] Burge, J., Design Rationale, Draft Technical Report, Worcester Polytechnic Institute, 1998.

[13] MATLAB: <http://www.mathworks.com>