Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-2000-257

Reducing conception times and promoting the use of distributed resources in space system design

J. Kos and A.A. ten Dam

Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-2000-257

Reducing conception times and promoting the use of distributed resources in space system design

J. Kos and A.A. ten Dam

This report is based on a presentation held at the 50th International Astronautical Congress, Amsterdam, 4-8 October 1999. The paper has appeared as IAF-99-U.1.02.

Division: Information and Communication Technology
Issued: May 2000
Classification of title:

Summary

Know-how and know-why of an enterprise is embodied in application programs, data, documents, and personal skills. Time and budget constraints, and an increased multidisciplinary approach result in the necessity to accommodate simulation programs and simulation tools. To be competitive, these programs and tools must be made available with minimal overhead. For this, NLR has made a number of working environments (depending on the application) to combine know-how, know-why and tools to support teams in realising project goals on time and on budget. Each working environment hides the distributed NLR computer infrastructure, which contains several servers that are located at two sites with a distance of 100 km, for the users. The present paper highlights the use of the ISMuS (Information System for Multi-body Systems Simulation) working environment for space system design and development. Special attention is paid to ISMuS' features for knowledge transfer using distributed resources through which conception times are reduced.

Contents

1 Introduction 4

2 Solutions to critical design issues in robotics 4

3 ISMuS and its support to the use and development of TRaCE 6

4 Sharing distributed resources 9

5 Knowledge transfer to other space applications 10

6 Model repositories 12

7 Concluding remarks 15

8 References 16

(17 pages in total)

1 Introduction

Timely knowledge transfer is crucial on all aspects of system design to realise reduction of conception times. The computer aided control engineering (CACE) working environment ISMuS (Information System for Multi-body Systems Simulation) improves knowledge transfer such that shorter conception times are realised, which in turn leads to shorter development time-scales and reduced cost. To support design and development, ISMuS contains various models for several applications domains, a number of model development tools, accompanying documentation, and all kind of data. The present paper illustrates the use of ISMuS on the development of one particular simulation environment, namely the TRaCE (Trajectory and Constraint Evaluation) simulation environment for the simulation of constrained robotic manipulators. Through ISMuS, the knowledge gathered from TRaCE has been distributed to various other applications. Moreover, the transfer of knowledge on simulation of interactive systems to training simulators involving vehicles is presented. In addition, the development of model repositories as carriers of enterprise knowledge is discussed.

The remainder of the paper is as follows. Section 2 deals with some critical design issues for space-borne robotic manipulators and describes the role of TRaCE and other models to solve these issues. In Section 3 a short overview of ISMuS is given and its use is illustrated on the development of TRaCE. Special attention is paid to the knowledge transfer during the development of TRaCE. Section 4 is an intermezzo about the SPINEware middleware package through which the ISMuS working environment has been created. The knowledge transfer from TRaCE to other applications through ISMuS is illustrated in Sections 5 and 6. Section 5 deals with training simulation involving (space) vehicles that are interacting with their environment; Section 6 concerns the development of model repositories for space applications. Section 7 contains the conclusions of the paper.

2 Solutions to critical design issues in robotics

Space-borne manipulators are typically used for moving payloads. Whenever a space-borne manipulator comes in contact with its environment, e.g. a satellite or the space station, it may not damage itself, the space station, or the payload. This makes the design of for example the control system for the robotic manipulator critical.

For over a decade NLR has worked on modelling, control, and simulation of a robotic manipulator that interacts with its environment[8]. The robot simulation environment TRaCE has been developed to investigate and solve the numerical problems in the simulation of a robotic

manipulator that is in or comes into contact with its environment and to evaluate control laws for automatic control of the manipulator under such conditions. NLR employees use TRaCE to execute feasibility studies on CACE topics that are identified both at NLR and in industry, whereas for instance master degree students from universities examine novel academic control paradigms on their merits via simulation studies.

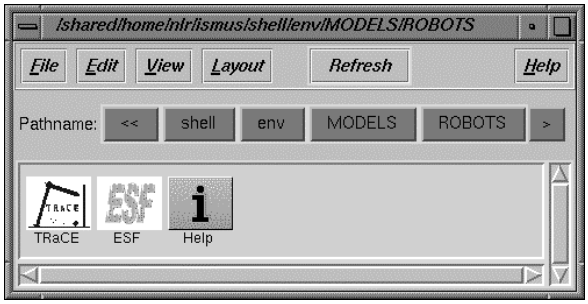

Figure 1 Robot model window in ISMuS

Knowledge that has been gathered through TRaCE has been explored in the contributions of NLR to the development of the ERA Simulation Facility (ESF) for real-time simulation of the European Robotic Arm (ERA)[3,10] on the International Space Station Alpha.

The 2D robotic manipulator in TRaCE is simulated throughout the various motion phases and events:

- Pre-impact phase, also called the free motion phase,
- Impact phase, the phase in which the effect of making contact are notable in the manipulator and/or surface,
- Post-impact phase, the phase in which the robot remains in contact with the surface,

Release phase, the phase in which a controlled release is made.

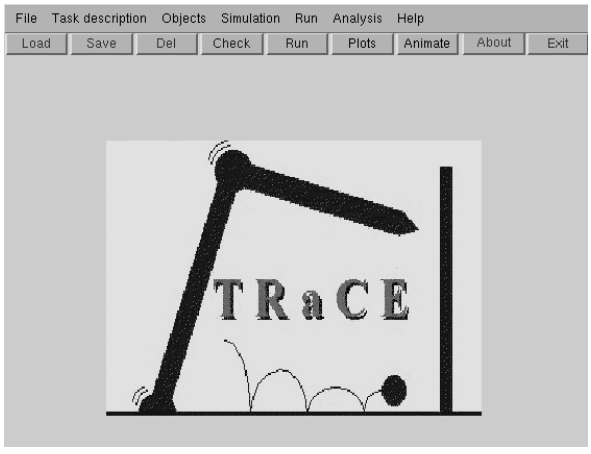

Figure 2 Opening window of TRaCE

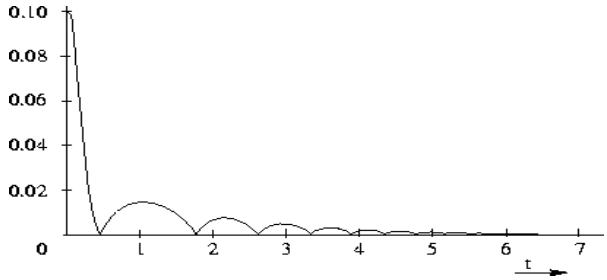

Figure 3 Simulation result using TRaCE: a potentially damaging contact between the end-effector of a rigid 2-D robotic manipulator and a rigid surface

TRaCE is a Matlab/Simulink based simulation facility. It contains various models of the robotic manipulator (including sensor models) and the environment (in particular with respect to friction between the manipulator and its environment in case of contact). Throughout the years the available set of controller paradigms that can be evaluated has been extending more and more, including various impedance control laws and adaptive control laws. The control laws focus on various controller tasks such as control to make smooth and fast contact with a surface and combined position/force control of the robotic manipulator whenever it is in contact with a surface. Control laws are evaluated extensively on robutstness with respect to various uncertainties: sensor model uncertainties, parametric uncertainties, manipulator model uncertainties, and friction model uncertainties. An example of a TRaCE simulation result is depicted in Figure 3.

3 ISMuS and its support to the use and development of TRaCE

Simulation and control engineers use the ISMuS working environment in the following NLR expertise domains:

- Robotics,
- Vehicles, and
- FDIR (Fault Detection, Isolation, and Recovery).

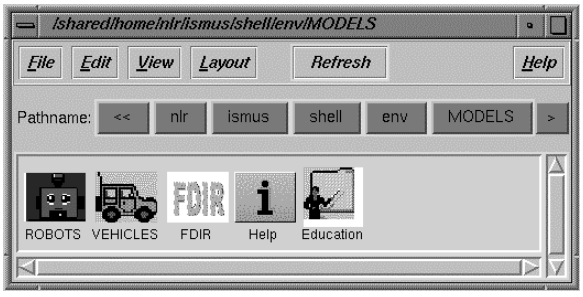

Figure 4 Application domain window in the Models part of ISMuS

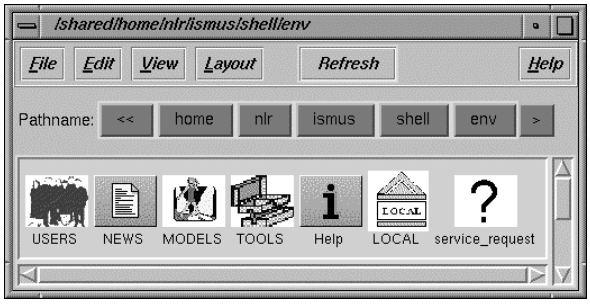

Figure 5 Top level window of ISMuS

ISMuS provides the engineers (Figure 5):

- Shared data, available to all ISMuS users;
- A private working directory (Local icon in Figure 5);
- The accessible working directories of other ISMuS users for co-operation (Users icon in Figure 5);
- The ISMuS service request system (Service request icon in Figure 5);
- Other icons.

Here data should be understood in a broad sense. Shared data in ISMuS includes at present:

- Validated models for simulation and control engineering in the ISMuS application domains (Models icon, see Figure 5);
- Tools for simulation and control engineering, which are used in the ISMuS application domains (Tools icon, see Figure 5);
- Educational models to learn about simulation engineering, control engineering, and the tools that are used (Education icon, see Figure 4);
- Documents about the models and tools, such as software user manuals and programmers guides;
- General utility tools such as text editors;
- Project management tools;
- Any other information that may help the engineer to use the tools and models available. This information is directly accessible from the working environment by opening the Help icon, upon which the appropriate html-pages that reside in NLR's intranet are opened.

The ISMuS service request system is a Web-based system that allows for complete traceability of the requests throughout the complete request handling procedure.

In the remainder of this section the use of ISMuS during the development of the TRaCE simulation facility is illustrated. The TRaCE simulation facility is used both for simulation and for further development of the facility.

Simulation executors can find TRaCE in the Models/Robots part of ISMuS (cf. Figure 1). When the TRaCE icon is double-clicked the window in Figure 6 appears. By further double clicking on the TRaCE_program icon the window in Figure 2 appears. Through the graphical user interface of TRaCE, the simulation executor sets the parameters as he likes, analyses the results, and saves both the input and the output data. The saved data can be accessed from the private working directory in ISMuS ('Local' in Figure 5) for further analysis. Guidance on the use of TRaCE (as far as it is not self-explaining) is found by clicking on the TRaCE_DOCS icon in Figure 6, which gives access to an on-screen-readable version of the Software User Manual. Besides using his own input data, the simulation executor can also adapt input data from ISMuS.

Developers can access the TRaCE window (Figure 6) from the Matlab window in the Tools part of TRaCE. By clicking on the Help icon in the window in Figure 6 the developer gets the information on the software configuration procedures, which are based on NLR's sr software configuration tool. Typically, the developer stores a work version of TRaCE in his private working directory so that he can access it through ISMuS.
SMuS provides PC-like drag-and-drop methods for editing, copying, viewing etc. of files. The developer can use his favourite tools for editing, viewing, etc.. To test the newly developed version of TRaCE, the work version is started by simple drag-and-drop of its main directory on the Matlab icon in the Matlab-tools working directory (cf. the CACE_tools window in Figure 9). Further information about the TRaCE program files and structure can be found in the TRaCE_DOCS directory, which contains the Programmers Guide of TRaCE. The latter directory also contains the source files of the Software User Manual and Programmers Guide for updating when necessary.

Figure 6 TRaCE window in ISMuS

Short introduction periods to TRaCE are crucial, since the academic trainees - who introduce novel approaches to control and numerical simulation - are only a short time at NLR. To shorten

the introduction period for them (and for new employees) some special features of ISMuS have been developed:

- Information on how to get acquainted with ISMuS can be found on NLR's intranet;
- Information on first time preparations before using TRaCE for the first time are found by clicking on the Help icon in the TRaCE window (see Figure 6);
- A short introduction to the use of TRaCE is found on the TRaCE Help page;
- Some special introductory simulation programs have been developed, which can be found in the Educational directory in ISMuS, see Figure 7.

The Educational directory contains an introductory simulation program for numerical issues in interacting mechanical systems and an introductory program for issues in control of mechanical systems. The introductory programs now also serve as a low complexity platform for testing purposes.

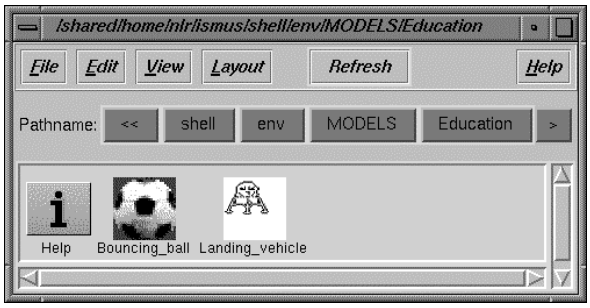

Figure 7 Eductional window in ISMuS

4 Sharing distributed resources

At NLR the SPINEware middleware package[13] is used for construction and operational use of functionally integrated working environments. SPINEware is developed by NLR for NEC[4]. SPINEware relies on the industrial standard on application level, the client-server approach. SPINEware supports the following functions[4,6]:

- Access to computer facilities and know-how as if everything is located on a PC (the virtual computer approach); this includes a graphical windows based user interface and tools to execute standard or routine chains of activities;
- Environment management, enabling customisation of personal environments;
- Know-how management, such as software version management on distributed computers;
- Graphical user interface based operation of tools to execute standard or routine chains of activities;

Moreover, SPINEware provides specific tool sets, such as tools for installing applications on the NEC/SX parallel vector computer at NLR, which acts as the central computing facility for large, demanding (both with respect to through-put time and memory use) application jobs. To enable programs to function transparently within distributed environments, this approach will be implemented in the near future using the emerging CORBA standard.

Various working environments have been created at NLR with SPINEware for specific aerospace domains and projects. The present paper concerns the ISMuS working environment for the development of and use in CACE. Some other working environments are the ISNaS working environment for Computational Fluid Dynamics[5] and the ISMO working environment for a project on Multidisciplinary Design Optimisation[4].

5 Knowledge transfer to other space applications

Various other space applications at NLR profit from the knowledge that has been gathered through the TRaCE simulation facility and that is distributed through the ISMuS working environment. Knowledge is transferred between model developers, simulator developers, and simulator users in the application domains. For instance, the graphical user interface set-up is generically applicable for various other (Matlab/Simulink) applications and it reduces the conception time for the simulation user through the similar look. In the present section the emphasis is on the transfer of the knowledge about the simulation and control of interacting (mechanical) systems from TRaCE to other space application domains. First a discussion of the critical issues in simulation of interacting (mechanical) systems is given. Next, the focus is on training simulators involving vehicles on rugged terrain. The transfer of knowledge from TRaCE to this field is highlighted. Through TRaCE the conception times of the employees in the SIMULTAAN[12] project with respect to the vehicle-terrain interaction have been greatly reduced. It is self-evident that the results that have been obtained through the SIMULTAAN project are stored in ISMuS for later use in future projects.

From a mathematical point of view, interactions of mechanical systems with their environment fall into the class of hybrid systems, or non-smooth dynamical systems. At NLR there are many applications that deal with non-smooth dynamical systems. Examples are:

- Robotic manipulators: simulation and control of constrained motions of the European Robotic Arm ERA, and of the manipulator in NLR's robotics laboratory;
- Wind tunnel experiments: contact between an aerospace model with the walls of the test section in a wind tunnel must be avoided at all times to prevent damage;

- Wheeled vehicles: simulation and control of a wheeled planetary rover under several terrain conditions.

Simulation of contact dynamics is by no means a trivial task, and the list below - taken from reference 7 - gives an idea of the difficulties that have been encountered and for which solutions have been derived at NLR.

- Study of wellposedness of impact dynamical systems, i.e. properties of solutions. This is important to obtain a unified treatment of contact problems across boundaries of application domains, and for enabling real-time simulation of constrained multi-body mechanical systems. Part of NLR's expertise is presented in reference 9.
- Study on impact between two rigid bodies via macroscopic laws that relate the motion after and before the shocks. A particular solution is described in reference 8.
- Develop new models of contact-impact laws for specific applications using engineering competence available in an organisation, see also Section 3.
- Investigate the complex dynamics of vibro-impact systems that may model systems with clearances. An example of such a system is ERA on the International Space Station[3,10].
- Study of numerical algorithms to integrate systems subject to unilateral constraints. Simulation of non-smooth mechanical systems is known to be liable to numerical instabilities. A procedure for real-time simulation has been developed at NLR[11].
- Design control strategies to improve the behaviour of mechanical systems subject to repeated impacts with the environment. Control of constrained robotic manipulators is part of ongoing research at NLR. Knowledge is accumulated in TRaCE, see Sections 2 and 3.

The transfer of knowledge from TRaCE through ISMuS is illustrated in the (space) vehicle domain (see also Figure 4). Within the SIMULTAAN (Dutch for simultaneous) project, in which the Dutch simulation industry jointly worked on distributed training simulation[12], NLR has, amongst others, been responsible for the vehicle Behaviour Model Component of a training simulator. The Behaviour Model Component simulates the behaviour of a large class of wheeled vehicles (with at least four wheels). The Behaviour Model Component consists of three interacting models (see also Figure 8):

- a Behaviour Model,
- a Control Model, and
- a Terrain Model.

Special attention is given to the interaction between the vehicle and the terrain to ensure proper motion cues to a driver during training sessions.

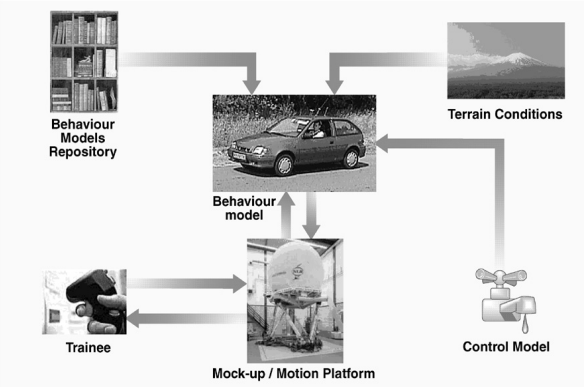

Figure 8 Overview of the Behaviour Model Component in the SIMULTAAN project

The know-how and know-why that is stored in TRaCE is re-used in the development of the Behaviour Model and the Control Model in SIMULTAAN. The four motion phases mentioned in Section 2 also apply to the vehicle behaviour with respect to vehicle-terrain interaction. Also numerical methods for the simulation of contact are re-used. Moreover, the know-how in TRaCE with respect to the development and the evaluation of control laws is used within the development of the Control Model in SIMULTAAN.

With the generic vehicle behaviour models a specific space vehicle can be simulated by appropriate choice of the parameters, such as the gravitation constant, mass properties and wheel characteristics. With appropriate modifications, these generic models can be made suitable to simulate other vehicles. For example, a planetary rover simulation can be obtained by relatively simple modifications to the SIMULTAAN multi-body dynamics model by adding, amongst others, more wheels, and by using a more complex, hinged-frame multi-body chassis model.

More details about the use of ISMuS in the development of the generic vehicle behaviour federate is presented in reference 1.

6 Model repositories

Model repositories are useful to provide access to departmental, workgroup or enterprise models. Know-how and know-why is stored in software, data and documents, which are placed in computer-based engineering infrastructures (i.e. repository). Tools must be available to access know-how and know-why in a coherent manner and to support a way of working that is directed towards accumulation and re-use of knowledge.

The present ISMuS model repository is of limited complexity and mainly provides systems and control engineers with available models. The SPINEware tool sr is used for configuration management in ISMuS (cf. Section 4). In ISMuS model repositories are seamless integrated with tool sets so that the complete software development process is supported over various projects. This line of working is detailed below and future developments are indicated.

To develop application software, e.g., to develop a space vehicle simulation model in the real-time simulation environment EuroSim, typically the following object-oriented development approach is taken. Firstly, an object-oriented design of the application software is made. Next it is identified which submodels of the application software need to be made and which submodels are available in ISMuS. The submodels that are not present in ISMuS are developed using CACE tools that are available in ISMuS. Finally, the submodels are connected through dedicated interfaces. This software development approach is supported by ISMuS. It contains both tools for the development of new models and a model repository.

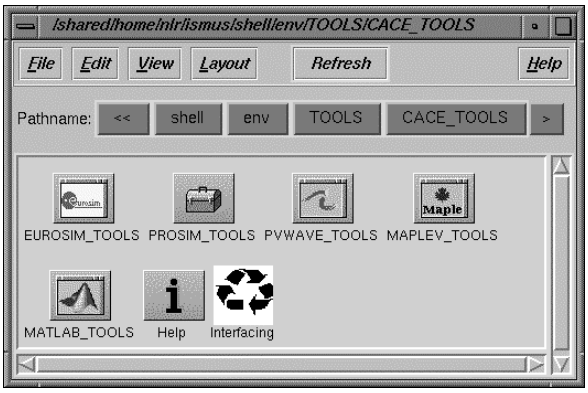

Figure 9 CACE-tools window in ISMuS

ISMuS supports the development of new models by providing appropriate model development tools, like MATLAB/SIMULINK, see Figure 9. The model is exported from the model development tool. Ideally, a model can be imported directly into a simulation tool.
ISMuS contains several simulation tools, like EuroSim, see Figure 9. EuroSim is a real-time simulation environment for real-time simulation with (if desired) a person and/or hardware-in-the loop (cf. Figure 10). The use of EuroSim within working environments is presented in reference 2.

The approach above is not always feasible: in many cases the exported model is not compatible with the simulation tool. In these cases the Interfacing icon in ISMuS (cf., Figure 9) gives the engineer access to information, procedures, and/or tools that will help him to extend and/or to modify the exported model such that it becomes compatible with the simulation tool.

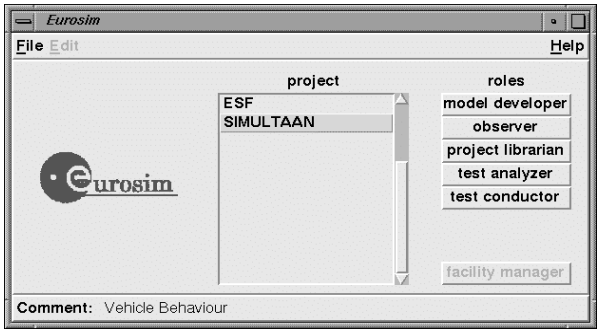

Figure 10 EuroSim simulation environment

At present, expertise is available with respect to the use of DYMOLA in the development of a multi-body model. The model has been extended so that is compatible with EuroSim. In the near future other model development tools like SIMPACK will be considered for inclusion in ISMuS.

Present developments are directed to further incorporation of model repositories. The envisaged model repository will consist of two parts:

- A shell, i.e. the IT part of the model repository;
- Aerospace models, i.e. the application dependent part of the model repository.

The model repository shell will provide the means to store and retrieve in a coherent manner all information that is relevant with respect to models. For each model this information will include at least:

- Model code;
- Design Documentation;
- User manuals, and related information;
- Relevant session results;
- Any other information (for example proprietary information).

The model repository will also enable to find related models, e.g., one model being a submodel of another model. Advanced model version management will be included in the shell.

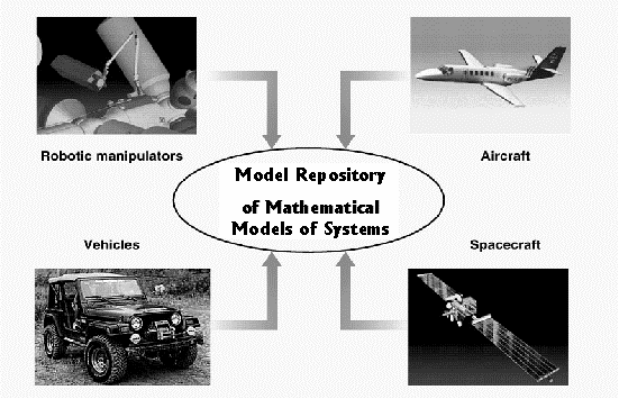

Figure 11 Model repository in ISMuS

In addition, information on the coupling of components to a simulation tool, for instance the coupling of a hardware component to EuroSim may be stored in a similar fashion (with the obvious replacement of the actual hardware by the interface software, such as DIS, HLA or any other interface description).

The shell will also provide users with standardised interfaces with commercial tools such as DYMOLA and DCAP for multi-body modelling, and MatrixX and Matlab/Simulink for control engineering. Furthermore, the model repository will provide interfaces to a 'simulator composition' tool that will provide a user with real-time dynamics and control simulation software ready for use in EuroSim. The model repository tools shall support a distributed architecture, i.e. the underlying databases may be installed on distributed computer systems yet offering the end-user a single comprehensive structure of storing and retrieving the desired model information. The aerospace application models in ISMuS will be a further extension of the models that are already present.

The availability of a model repository within ISMuS also facilitates fast and early prototyping of systems by providing simulation models as objects to an engineer, thus stimulating the re-use of existing models. In this way the complete system life cycle is supported.

In the framework of GARTEUR, NLR has recently started to work on the development of a European-wide model repository as sketched in the preceding paragraphs.

7 Concluding remarks

NLR's working environment ISMuS for Computer Aided Control Engineering (CACE) provides all means that the developers of simulation software and simulation executors need for

their job. ISMuS includes model development tools, models, and accompanying documentation. Through a coherent presentation of the means and dedicated interfaces (many of them are standard SPINEware) the developers and simulation executors can work faster, more accurate, and with more job satisfaction. Data from distributed resources are integrated in the engineer's working environment as if the data were not distributed at all. In this way conception times are reduced considerably and the use of distributed resources is self-evident. Through continuous extension of the contents of ISMuS with newly acquainted knowledge NLR will be able to deliver its advanced products faster and faster. Through ISMuS NLR demonstrates its experience with the creation of working environments for various external and internal customers, and part of its competence in space system design.

8 References

1. J. Kos, A.A. ten Dam, G.W. Pruis, and W.J. Vankan, Efficient Harmonisation of Simulation Competence in a CACE Working Environment: Its Use in the Developoment of a Generic Vehicle Behaviour Federate, *Proceedings of SESP '98, Workshop on Simulation for European Space Programmes,* ESTEC, Noordwijk, 1999.
2. A.A. ten Dam and R.J.P. Groothuizen, 1998, Competence Management for EuroSim Centred CACE Working, *Proceedings of DASIA '98 DAta Systems In Aerospace*, Athens, Greece, 393-398.
3. C. Maegaard and P. Beerthuizen, 1998, Development of a Safety Critical Hard Real-Time System in a World of Changes, *Proceedings of DASIA '98 DAta Systems In Aerospace*, Athens, Greece.
4. W. Loeve and M.E.S. Vogels, 1998, Competence management in engineering environments with HPCN, *Proceedings of the Conference HPCN Europe*, Amsterdam, High Performance Computing and Networking, Springer.
5. R.J.P. Groothuizen and H. van der Ven, 1998, A NICE HPCN Centre for Flow Simulation, *Proceedings of the Conference HPCN Europe*, Amsterdam, High Performance Computing and Networking, Springer.
6. E.H. Baalbergen, 1998, SPINEware: a Practical and Holistic Approach to Metacomputing, HPCN Europe '98.
7. B. Brogliato, 1996, *Nonsmooth Impact Mechanics, Models, Dynamics, and Control*, Springer Lecture Notes in Control and Information Sciences 220.
8. A.A. ten Dam and J.C. Willems, 1997, A System Theoretical Framework to Study Unilaterally Constrained Dynamical Systems, *Proceedings of European Control Conference ECC'97,* Brussels.

9. A.A. ten Dam, K.F. Dwarshuis, and J.C. Willems, 1997, The Contact Problem for Linear Continuous-Time Dynamical Systems: a Geometric Approach, *IEEE Transactions on Automatic Control*, 42, 4, 458-472.
10. R. Boumans, C. Heemskerk, E. Holweg, S. Kampen, M. van Lent, A. Pouw, 1996, ERA Baseline Capabilities and Future Perspectives, *ASTRA Conference,* ESTEC, Noordwijk, The Netherlands
11. A.A. ten Dam, 1992, Stable Numerical Integration of Dynamical Systems Subject to Equality State-Space Constraints, *Journal of Engineering Mathematics*, 26, 315-337.
12. http://www.nlr.nl/public/hosted-sites/simultaan/ model.html
13. http://www.spineware.nl/