48

Network middleware for enterprise enhanced operation

W. Loeve, M.E.S. Vogels, H. van der Ven, E.H. Baalbergen
National Aerospace Laboratory (NLR), The Netherlands
Anthony Fokkerweg 2, 1059 CM Amsterdam, NL
tel. 31 (0)20 5113421, fax 31 (0)20 5113210,

Abstract

Enterprises are confronted with the challenge to lower cost of R&D and product engineering. More efficient transfer of know-how from R&D to engineering and between (generations of) engineers facilitates facing the challenge. Enterprise know-how has to be conserved in enterprises in the form of software such as for simulation, electronic documents and data. IT can enhance enterprise operation by giving engineers that work in or for the enterprise, easy access to the enterprise know-how. This requires a combination of distributed and centralized computing, a suitable network strategy and IT tools for management, control and use of the conserved enterprise know-how. In the paper the middleware SPINE for UNIX networks is described by which the network presents itself to the engineers as one single virtual computer. In the paper the considerations and the technical principles are illustrated by examples from two applications: a national virtual centre for Computational Fluid Dynamics for engineering and an IT framework for distributed multi-discipline design, analysis and optimisation of industry products. An estimation is given of the effort to be spent to realize and maintain such virtual single computers for a cooperating group.

Keywords

Enterprise network, middleware, IT, enterprise automation, engineering.

List of abbreviations

AQAP	NATO Quality Assurance Requirements for Design/Development and Production
CFD	Computational Fluid Dynamics
DOC	Direct Operating Cost
ESTEC	European Space Research and Technology Centre
HFS	Nation Wide Virtual Centre for Flow Simulation in The Netherlands
HPCN	High Performance Computing and Networking

Computer Applications in Production and Engineering. F. Plonka and G. Olling (Eds.)
© 1997 IFIP. Published by Chapman & Hall

IBM SP2	American Parallel Scalar Computer with Distributed Memory
ISNaS	Computer Based Working Environment for CFD
ismo	Computer Based Cooperative Working Environment for MDO
ISO	International Organization for Standardization
IT	Information Technology
MDO	Multi-disciplinary Design, Analysis and Optimization of Aerospace Vehicles, Project
MFLOP/s	Mega Floating Point Operations per second
NEC SX-4	Japanese Parallel Vector Computer with Shared Memory
NICE	Project aiming at a Nation Wide Virtual Centre for CFD in The Netherlands
NLR	National Aerospace Laboratory in The Netherlands
PSUE	Parallel Simulation User Environment, Software Utility
R&D	Research and Development
SPINE	Software Utility for Presenting a Computer Network as one Single Virtual Computer to Users
sx4dev	A Collection of Tools for Development of Software for the SX-4
TCP/IP	Computer Communication Standard
TNO-TPD	Technical Physical Institute in The Netherlands
UNIX	Computer Operating System, Trademark

1 INTRODUCTION

Enterprises are confronted with the challenge to lower cost of product engineering. In this context engineering staff is minimized. Increasing the efficiency of enterprises by minimizing staff requires increased attention for management and control of engineering know-how. In view of this engineering know-how has to be conserved in enterprises in the form of software such as for simulation, electronic documents and data on electronic carriers. For engineering know-how in these forms it is possible to make use of Information Technology (IT) for management, development control, access and application.

R&D has to lead to production of products that are appreciated better by customers, that can be delivered at lower cost and for which engineering and production is possible in shorter time. For R&D results to become useful for an industrial enterprise, R&D results have to become enterprise know-how. This means that R&D results have to be transferred from R&D organisations to industrial enterprises as ready to use software, documents and data.

Enhancing enterprise operations in engineering is possible by enhancing accessibility and applicability of enterprise know-how. Minimizing staff to lower enterprise cost nowadays very often means that engineers are only hired to perform specific jobs. This means that it is required to enhance accessibility and applicability of enterprise know-how for engineering with partly temporary staff members. Using IT makes it possible to transfer more efficiently know-how between (generations of) engineers at various geographical locations.

It is a task of the enterprise itself to control the use of IT for enhancing operations. As a consequence IT is at least a required core competence of the enterprise. In chapter 2 a UNIX network and network middleware is described by which enterprise enhanced operation is possible. In chapter 3 and chapter 4 the use of the network and the middleware is illustrated to generate IT based working environments for application of computational fluid mechanics and for application of a combination of computational fluid mechanics and computational solid mechanics respectively. Chapter 5 contains information concerning effort to be spent by any participating company for establishing the working environment and for maintaining it.

2 A UNIX NETWORK FOR OPTIMAL ENGINEERING SUPPORT

2.1 Computer network required for engineering

When we look at the most powerful compute servers installed in the world (Dongarra, 1996) we see, at the top of the list, massively parallel computers with many scalar processors and distributed memory and vector computers with several processors and shared memory. The vector computers are easier to use than the distributed memory computers, especially with computational mechanics software that is frequently used in engineering (Vogels 1995).

The most advanced industrially applicable computational fluid mechanics (CFD) software for aerodynamic analysis in aircraft engineering at the moment is based on the Reynolds-Averaged Navier-Stokes equations (Rubbert, 1994). One not too detailed computation for a full aircraft configuration requires about 5 hours in case of a sustained computer performance of about 500MFLOP/s. One of the interesting applications of these computations for the near future is support of aerodynamic optimization (Borland, 1994). Optimization requires more computer performance than analysis, because for optimization for instance about 40 times solution of the flow equations in combination with an optimization algorithm is required. With a NEC SX-4/16 (a vector computer with 16 processors and shared memory) it is possible to realize for CFD a computational performance on 14 processors of 8,500MFLOP/s (Ven van der, 1997). With knowledge of only a few techniques on vectorization and coarse grained parallelization, this is realized with relatively little effort (only DO-loop parallelization). One aerodynamic optimization of 40 analysis runs then takes about 10 hours. Based on this reasoning NLR decided to procure a NEC SX-4/16 for R&D and for support of engineering in The Netherlands (Loeve 1996, Hameetman 1997).

The process of engineering an aircraft is characterised by a large number of activities executed by engineers from various disciplines mostly located at different geographical locations. The engineers use a variety of simulation software tools. Engineers need workstations for graphical interaction with their tools. Sharing of know-how between specialists has to be safeguarded which means that information servers and intranet servers are required in addition to compute servers and workstations. The network has to give access, for all involved persons and groups, to the computers as well as the software, data and electronic documents that are stored on the computers. Computers, workstations and information servers such as data management servers with tape robots have UNIX operating systems. Complexity of the computer networks and of the collections of software systems used in engineering processes both increase continuously in industry.

2.2 The need to present the network to engineers as a unity

Many problems occur in practice when engineers try to use a network:

- Remote access to computers is difficult to organize.
- Programs lack uniformity in user interface and data interface. Incidental use is practically impossible because of time required to familiarize.
- Problems with programs on different computers can be:
  - finding out on which computer a specific program may run,
  - remote login and remote execution, facing the user with aspects such as authentication and accounting,
  - operation of different operating systems or variants of the "same" operating system cause incompatible utilities,
  - exchange of files between computers,
  - incompatible data formats due to different byte ordering and representation of real numbers in different computers.
- Users have to organize, manage, and find a way through more on-line information than that is accessible on a single computer.

Removal of the technical and organizational shortcomings mentioned above will enhance the enterprise operation for simulation in engineering. Information about know-how (data, documents and software) shall be made available on-line. Information shall be managed and controlled in such a way that reconstruction of old information is possible and information can be exchanged between specialists in the same way as if they all make use of one computer. Accessibility and applicability of the software has to be such that the user does not have to be skilled in areas such as, computer programming, computer operating systems and computer network tools. Finally the user has to be assisted in organizing, re-using and integrating a continuously growing amount of software, data and supporting help information.

Only by making high performance computing a natural part of desktop workstation applications, the use of high performance computing in engineering will spread according to the need for competitive enterprise operation.

2.3 Network middleware for enterprise enhanced operation

Since computers were introduced at NLR in the late fifties this institute has applied a centralized IT approach. The main reasons were economy of scale and the need to be able to procure the computing power that is required for the most demanding applications (Loeve 1976). In addition productizing of software developed by NLR for CFD in industry is performed by IT professionals. In this development much attention was paid always to quality management. This resulted in an ISO 9001 and AQAP-110 certificate for development of simulation software. It also resulted in development of architecture and construction principles for development of CFD software according to industry needs (Vogels 1997).

Finally, quality management resulted in the development of a software utility called SPINE that supports the construction of general as well as application-oriented working environments (Baalbergen 1994). A SPINE-based working environment gives access to local and remote computing resources and information in a network. The network presents itself with SPINE as one single computer (a "metacomputer") to its users. In addition to hiding the networking details, SPINE provides mechanisms for easy customization of the metacomputer (to the needs of a particular user, group of users, or application area), for easy operation of the metacomputer by the user, and for management and control of information. SPINE is being further productized jointly by NEC and NLR. Currently, SPINE is based on, and targeted towards UNIX computer networks, in which the individual computers run vendor-supplied UNIX and standard TCP/IP-based networking software and tools.

The working environments constructed using SPINE are highly portable and easy to customize. Moreover, the working environments are open and extendible, in that any UNIX and TCP/IP based computing system can easily be integrated, and that existing software and other forms of electronically available information can easily be integrated. To support the operational use of a working environment, SPINE comprises technical solutions for the key elements of a metacomputer: an easy-to-use desk top, facilities for the management of information, and middleware.

The desk top provides the user of a working environment with a graphical integrated user shell. The shell provides a uniform and consistent view with respect to presentation and manipulation of data files, documents, and source code, and with respect to execution of tools. The data management and processing operations are available in terms of point-and-click and drag-and-drop operations on windows and icons. The shell provides control over multiple windows in combination with multi-tasking. This makes it possible to have several tools running simultaneously. The shell also provides the user with contextual on-line help, both on the operational aspects of the working environment as well as on the applications.

In addition to the Windows look-and-feel of SPINE, the shell supports data-oriented, tool-oriented, as well as work-flow driven modes of working. The work-flow capability of the shell supports work-flow management, in which data and tools can be configured in a network that supports, or even may enforce, certain scenarios or activities. Work flows are considered to resemble an engineer's way of working, and to provide an intuitive means of specifying jobs and controlling the execution of the jobs. Work-flow management also enables transparent and automated information management, e.g., to enforce synchronization and reconstruction of activities and results.

SPINE supports the Computer Supported Cooperative Working aspects of information management by providing tools for managing storage, access to, and modifications of data, documents, software and any information related to this, and for the exchange of information among users. The task of SPINE's middleware is to manage the available resources in order to accomplish the single computer look-and-feel, and to exploit the potentials of the underlying computer network. The middleware also takes care of the application of data conversions required to deal with hardware and data-format heterogeneity.

SPINE, by being based on standard UNIX and TCP/IP software, has proved to be applicable for realization of environments that span both local-area and wide-area networks. For example, the NICE environment (see chapter 3) spans the local area networks of the NICE partners, but also gives access from remote workstations to NLR's SX-4 vector computer and CFD software as well as the compute power and software in other partners in the NICE project, over the Internet. In this way, a Netherlands-wide CFD working environment has been realized.

At the moment, several systems exist that support realization of, mostly only part of the aspects of, metacomputing. For example, the Parallel Simulation User Environment (PSUE), aims to provide a software platform that can be customized for particular simulations. However, PSUE provides a particular desk-top model, which leaves hardly any freedom with respect to customization, and which enforces a tool-oriented mode of working, whereas the SPINE desk-top model supports data-oriented, tool-oriented, as well as work-flow driven modes. Another example is SiFrame, an open working environment that aims at supporting design-process management in a multi-user environment. SiFrame, however, concentrates mainly on the work-flow driven way of system interaction.

Most developments in metacomputing are primarily driven by research in computer science, which has resulted in a gap between scientifically well-founded metacomputing solutions and industrial needs. This gap often leads to products that do not meet the demands from engineers. SPINE, although including results of scientific research, has its roots in a practical engineering environment, viz. ISNaS (Section 3.1) at NLR, has proved to be the solution that meets the demands in modern engineering. Examples of SPINE applications are given in the next chapters.

3 THE USE OF THE UNIX NETWORK AND MIDDLEWARE FOR COMPUTATIONAL FLUID MECHANICS

3.1 CFD working environment in one organisation

Since the earliest developments of SPINE it has been applied to ISNaS, a working environment for flow simulations (Vogels 1989). At first, the working environment was aimed to support the use of flow simulation packages across the NLR network that connects the two NLR establishments. The distance between the establishments is 100 km. The network consists of workstations and terminals that give access to mainframes acting as compute-, intranet-, internet- and data management servers. The centralized automation policy at NLR made it economically possible also to integrate a supercomputer in the network since 1987. Because of its success, more functionality was and is added continuously to the network and to the network middleware of SPINE.

ISNaS now supports both use and development of simulation software. The use of the simulation software is supported by integrating the entire pipeline of simulation analysis: geometry modellers, grid generators, flow solvers, and postprocessors. File transfer and remote logins are hidden from the user. Exchange of different file formats between the various tools is made transparent to the user using the SPINE facility for implicit file conversion. Exchange of documents and data is made possible through easy-interface databases. Feedback to the developers of both the working environment and the simulation software is made easy by the use of electronic forms.

The development of simulation software is supported by providing via SPINE a software version management tool, and a toolpack sx4dev which integrates several development tools on the NEC SX-4 (more about the toolpack in Section 3.2). A framework for the creation of a regression test suite is available within the working environments. For support of development control, electronic forms are used to manage error reports.

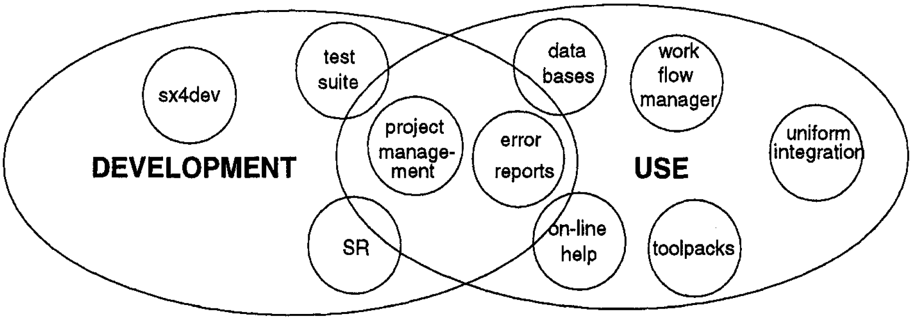

Figure 1 Building blocks for both use and development of engineering software.

For both users and developers project management systems can be set up. Such a project management system consists of an electronic project archive, a search engine on this archive, a drag-and-drop mechanism to submit documents to the archive, and a database for electronic forms. We will see an example of such a project management system in Section 3.2.

The building blocks of the working environment supporting both use and development of CFD software are shown schematically in Figure 1. As is clear from the figure development and use interact and overlap.
The development of a working environment is dominated by two principles:

- an integrated tool has added value to the user,
- the working environment shall adapt to the user and not vice versa.

This also implies that integrated tools do not stand alone in the working environment: they are surrounded by tools that either supply input or use the output of the specific tool. This integration of tools greatly facilitates their use.

The success of the present working environment is probably best exemplified by the following. A CFD trainee was asked to perform an analysis of air heater flow. The trainee had a thorough knowledge of flow physics, and numerical mathematics. But he was less familiar with supercomputers, networks, UNIX and postprocessing. Using the working environment he was able to perform and analyze a specific flow configuration within one week. The analysis consisted of the entire pipeline from preprocessing up to visualization.

3.2 National CFD working environment

To enhance the use of HPCN techniques the Dutch government partly finances the project NICE, Netherlands Initiative for CFD in Engineering. NICE aims to promote the use of CFD methods, developed at universities and large technological institutes, to industry and small and medium enterprises. On the one hand, new (parallel) algorithms are developed motivated by the needs and wishes from the industry, on the other hand, these algorithms are used to analyze flow problems from the same industry. Development and use are combined in a single initiative.

In order to facilitate the use of remote resources NICE develops a nationwide virtual centre for flow simulation HFS based on SPINE. HFS consists of two parts: a common part to all NICE partners, and a part specifically tuned to the needs and wishes of a NICE partner. The network on which HFS operates consists of the local network of the partner and the NEC SX-4 at NLR. HFS is a specialized 'ISNaS' at each partner, extended with a common set of tools for both project management and development on and use of the NEC SX-4.

Management of the project NICE is facilitated through HFS. A project management system is part of the common tools in HFS. The system consists of an electronic project archive, electronic error reports, and search engines to search both the project archive and the error report database. The project archive has a technical report repository. Development on and use of the NEC SX-4 is supported by the toolpack sx4dev that integrates tools for compiling, executing, analysing and debugging source. Automatic makefile generation allows for manipulation of the source code. Options of the tools are based on the experience of experts in supercomputing. Using this tool the computational kernel of a flow solver of TNO-TPD was compiled, executed and analyzed within a quarter of an hour. Before that, TNO-TPD had never worked on an NEC supercomputer.

By supporting both development and use of CFD software HFS is suited for a wide range of customers. From novice users inexperienced with CFD, through experienced users with a need of more powerful resources, up to developers of new algorithms and software.

An example of a customer needing more powerful resources is ESTEC. For a time-accurate simulation using a multi grid, multi block structured solver they required more memory than their IBM SP2 could supply. Moreover, throughput times became prohibitive. Even though the code was originally written for scalar machines, a porting effort of roughly three manweeks increased the megaflop rate from 80 MFLOP/s on the IBM SP2 to 500 MFLOP/s on the NEC SX-4.

4 THE USE OF THE UNIX NETWORK AND MIDDLEWARE FOR MULTI-DISCIPLINE ANALYSIS AND OPTIMIZATION

4.1 Problem analysis

To effectively resolve cross-discipline trade-offs both to improve aircraft performance and reduce development timescales and costs, concurrent engineering principles are under investigation for the preliminary design stage.

In Europe, the MDO Consortium [1] is addressing integration of design and analysis tools creating a Multi-Disciplinary Optimisation (MDO) capability (MDO Consortium,1996). The purpose is to develop and demonstrate the viability of MDO and validate this for simplified but realistic aircraft preliminary design tasks concerning the design of a large civil aircraft where a non-trivial interaction is expected between the mono-disciplines of aerodynamics and structures. NLR, one of the Consortium members, carries out the work in a multi-disciplinary team with specialists from aerodynamics, flight mechanics, structures, and information technology.

[1] The MDO project (Multi-Disciplinary Design, Analysis and Optimization of Aerospace Vehicles) is a collaboration between British Aerospace, Aerospatiale, DASA, Dassault, SAAB, CASA, Alenia, Aermacchi, HAI, NLR, DERA, ONERA, and the Universities of Delft and Cranfield. The project is managed by the British Aerospace and is funded by the CEC under the BRITE-EURAM initiative (Project Ref: BE95-2056).

In the MDO problem, different aircraft designs are compared by comparing the Direct Operating Cost (DOC) for the aircraft design. For the problem at hand, the DOC is simplified in a model containing drag and total weight. The problem of DOC evaluation for an aircraft design can be solved in more than one process, each leading to a different approximation of the DOC. Therefore a DOC value for an aircraft design can only be interpreted in the context of the process in which it was achieved.

Process model

The process is modelled in so-called N^2 diagrams where the *(N)* contributing disciplines form the diagonal of an $N*N$ matrix, the off-diagonal entries show the dependencies between the disciplines. An above-diagonal entry denotes a feedforward coupling, a below-diagonal entry denotes a feed backward coupling.

The top-level N_2 diagram for the NLR process for the early problem is given in figure 2. The figure shows the involvement of the disciplines. On lower levels, similar N_2 diagrams show the detailed processes. The level of detail of the N^2 diagrams is dictated by its usefulness for the MDO-process and usually stops at the level of a mono-discipline.

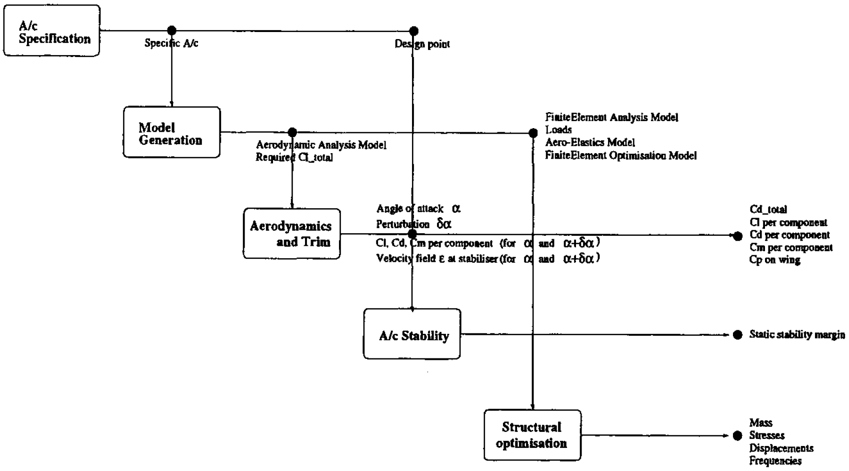

Figure 2 Multi-discipline analysis process

Changes in the mono-discipline software for the multidiscipline context

On the MDO level, it is assumed that all disciplines have their own (mono-disciplinary) application software. The application software runs on the NLR heterogeneous UNIX network, involving e.g. SGI work stations, CD4810 scalar compute server, and the NEC SX-4 parallel vector machine. Data, documents and software are stored via a data management server on a tape robot. Most software is home-brew, implemented in Fortran and C. The software frequently changes to satisfy additional requirements.

Three types of changes are required for the mono-discipline software to function in the multi-discipline context. First, the mono-discipline software has to accommodate for the communication with the other disciplines, on the level of detail and in the terminology specified in the N^2 diagrams. Whenever possible, this accommodation consists of interface programs which translate between the MDO-parameters/results and the mono-discipline's parameters/results.

In this way, the mono-discipline application software can remain unchanged, and can be used for solving mono- and multi-discipline problems.

Second, whereas in mono-disciplinary work it may be acceptable that the work halts when a specialist is unavailable, in multi-disciplinary work carried out in a team, it is not acceptable. The work may be taken over by a colleague less familiar with the MDO problem, or by a multi-discipline team member of a different discipline. A prerequisite for enabling someone else to perform the absent specialist's work, is accessibility to the mono-discipline software. This is organised in the MDO environment "ismo" (Section 4.2).

And, finally, again because of the team character of the multi-disciplinary work, it is required that the application software, used for the MDO work, is identified and reproducible.

4.2 Set-up of MDO environment

One of the purposes of the MDO environment is, as explained above to provide access to all mono- and multi-discipline application software of the MDO team. This includes the shielding of the computer network, and providing the virtual computer feel. For this purpose, the mono- and multi-discipline application software is integrated into an MDO environment based on SPINE.

Application software which is unchanged by the multi-discipline context, and which is already under version management, is considered to be identified and reproducible. Newly developed application software (such as the interface programs with the MDO-level), and changes in existing application software are version managed in the MDO environment.

The challenge for information technology is not only to capture the process as in N^2 diagrams, but also to supply an MDO environment which supports the work according to the MDO process view of the disciplines involved. The application software in the MDO environment is structured (in levels and groups) according to the N^2 diagrams. In this way, the MDO environment reflects the MDO team's view on the MDO process. In Figure 3, the highest level of MDO programs is given, showing the elements (in alphabetical order):
1 ac-performance; 2 ac-specification; 3 ac-stability; 4 aerodynamics-and-trim;
5 model-generation; 6 multi-inspection; 7 structural-optimisation.

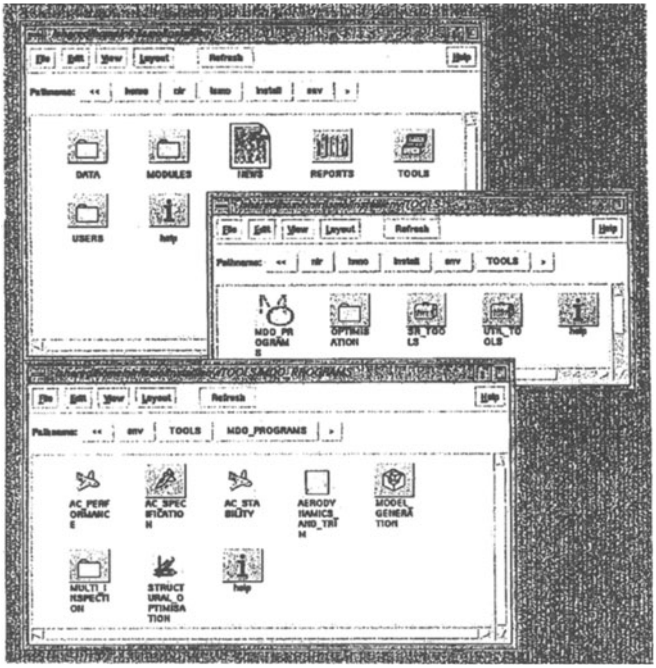

Figure 3 Screen-dump of"ismo" showing the top-level of MDO programs.

The diagonal elements in the MDO process (Figure 2) correspond to the "ismo" elements 2, 5, 4, 3, 7 respectively. The "ismo" elements 1 and 6 are additional compared to the diagonal elements in the MDO process; they are included in the environment "ismo" because:

- "ac-performance" has been identified as a contributing discipline, but has been left empty for the problem at hand,
- "multi-inspection" is a passive process, not producing any information to the DOC formula.

5 EFFORT TO EFFECTUATE AN IT SUPPORTED COOPERATIVE WORKING ENVIRONMENT

The common working environments for cooperating groups of engineers as described above are based on the principle that members of a cooperating group are willing and able to share information that is made available by members of the group via procurement or development. This concerns information in the form of software, data or document. Sharing information means that development control of information is applied by each originator. This concerns such aspects as version control as well as planning of activities and communication about planning issues. Also standards for the structure of information that support access of information, and maintenance procedures have been agreed upon by the group and are being applied. The information has been described preferably in such a way that the descriptions can be provided as help information for non-regular users or even be provided in the form of tools that facilitate or support execution of the procedures. Rules have been developed with regard to authority to modify, release, consult and apply information. Control mechanisms for access to the computers have been implemented to support application of the rules.

The effort to be spent to realize that specialists are willing to cooperate, to share information and to make this also possible strongly depends on the degree of application of process and product control in the concerning enterprise(s). It may require a costly change of mentality such as when for instance process and product control according to the ISO 9000 industry standard has to be introduced in an individualistic R&D environment. One of the reasons to spend the effort in such cases may be the need of transformation of the situation with operation mainly based on input financing to the situation with operation mainly based on output financing in a competitive environment. The effort may take several calendar years and may cost several manyears even for a department with less than 20 people. Realization of an IT supported cooperative working environment requires that all concerning individuals have a desktop computer that is connected to a common network. This may require an additional effort for enterprises with many isolated desktop computers.

Installation of the SPINE software and a working environment containing a small number of tools takes only one day effort by experts. Maintenance and extension of functionality of the working environment take roughly four manweeks a year, and depend on the number of users and the number of tools. These figures are based on several years' experience at NLR and at organisations that NLR cooperates with on the basis of SPINE working environments. For new working environment managers a day's course introduces them to the concepts and structures of SPINE. In our experience UNIX experts quickly grasp the potential of SPINE. SPINE is currently extended with functionality for support of working environment modifications. This will significantly reduce the effort required for maintenance and modification. It appears that using a SPINE based working environment even for a limited number of activities in an enterprise may facilitate further introduction of product and process control according to the ISO 9001 standard in the enterprise.

6 CONCLUDING REMARKS

Hiring engineering staff when needed requires that enterprise know-how is conserved in software for simulation, electronic documents and data.

For engineering in industry it is essential that cooperating engineers can use the computer network of the enterprise as the central nerve system for exchanging, accessing and using enterprise know-how.

Network middleware such as in SPINE can be used to realize that for cooperating engineers a complicated (national) high performance computer network can be accessed and used as one single virtual computer.

7 REFERENCES

Baalbergen, E.H. and Loeve, W. (1994) SPINE: software platform for computer supported cooperative work in heterogeneous computer and software environments, NLR TP 94359, National Aerospace Laboratory Amsterdam.

Borland, C.J., Benton, J.R., Frank, P.D., Kao, T.J., Mastro, R.A. and Barthelemy, J.F.M. (1994) Multidisciplinary design optimization of a Commercial Aircraft Wing - An Exploratory Study. 5th Symposium on Multidisciplinary Analysis and Optimization, Panama City Beach, Sept. 7-9, 1994. AIAA-94-4305-CP, pp 505-519.

Dongarra, J.J., Meuer, H.W. and Strohmaier, E (1996) TOP500 Supercomputer Sites November 1996; retrievable from internet by anonymous ftp to www.netlib.org/benchmark/top500.ps.

Hameetman, G.J., Loeve, W., Poppinga, G., Potma, K. (1997) Early bench mark results on the NEC SX-4 supercomputer; in *Parallel Computational Fluid Dynamics (Algorithms and results using advanced computers)* (ed P. Schiano et al), Elsevier ISBN 0 444 82327 1.

Loeve, W. (1976) A hierarchical network linking two research laboratories; Computer Networks 1 North-Holland Publishing Company.

Loeve, W. (1996) From parallel CFD to a tool for computer aided engineering; in *Parallel Computational Fluid Dynamics* (ed A. Ecer et al) Elsevier ISBN 0 444 82322 0.

MDO Consortium (1996), http://www.telecall.co.uk/~srcbae/mdo.

Rubbert, P.E. (1994) CFD and the Changing World of Airplane Design; AIAA Wright Brothers Lecture, Anaheim California; September 18-23.

Ven, van der H. and Vegt van der J.J.W. (1997) Experience with advanced CFD algorithms on NEC SX-4; in *VECPAR '96 selected papers from the 2nd International Meeting on Vector and Parallel Processing (Systems and Applications)* (ed Laginha M. Palma et al), Lecture Notes in Computer Science 1215, Springer 1997.

Vogels, M.E.S. and Loeve, W. (1989) Development of ISNaS: An Information System for Flow Simulation in Design, in *Computer Applications in Production and Engineering* (ed F. Kimural) North-Holland ISBN 0 444 88089 5.

Vogels, M.E.S., Ven van der H., Hameetman, G.J. (1995) The need for super computers in aerospace research and industry, in *Proceedings HPCN Europe 1995 Milano.*

Vogels, M.E.S. (1997) Architecture and construction principles for computational fluid dynamics software for engineering in industry; Thesis, ISBN 90-9010295-7; M.E.S Vogels, National Aerospace Laboratory Amsterdam.

BIOGRAPHY

W. Loeve's university degree is in aeronautical engineering. He is head of NLRs IT Division, IT consultant and responsible for quality management in the development by NLR of aerospace information systems for customers.

Dr M.E.S. Vogels's PhD thesis concerns architecture and construction principles of CFD software for engineering in industry. She is IT consultant and projectleader at NLR, responsible for the development of the IT framework in the MDO project.
Dr H. van der Ven's PhD thesis concerns harmonic analysis on Lie groups. He is HPCN consultant and projectleader at NLR, responsible for the ISNaS and NICE working environments.
Dr E.H Baalbergen's PhD thesis concerns distributed operating systems. He is projectleader at NLR for the SPINE development in interaction with applications such as mentioned above.