



NLR TP 97389

Network Middleware for enterprise enhanced operation

W. Loeve, M.E.S. Vogels, H. van der Ven and E.H. Baalbergen

DOCUMENT CONTROL SHEET

	ORIGINATOR'S REF. NLR TP 97389 U		SECURITY CLASS. Unclassified												
ORIGINATOR National Aerospace Laboratory NLR, Amsterdam, The Netherlands															
TITLE Network Middleware for enterprise enhanced operation															
PRESENTED AT the 4th International Conference on Computer Integrated Manufacturing; 21-24 October 1997, Singapore.															
AUTHORS W. Loeve, M.E.S. Vogels, H. van der Ven and E.H. Baalbergen	DATE 970805	pp 15	ref 10												
DESCRIPTORS <table style="width: 100%; border: none;"><tr><td style="width: 50%;">Applications programs (computers)</td><td style="width: 50%;">Information flow</td></tr><tr><td>Client server systems</td><td>Man--computer interface</td></tr><tr><td>Communication middleware</td><td>Multidisciplinary design optimization</td></tr><tr><td>Computational fluid dynamics</td><td>Resource allocation</td></tr><tr><td>Computer networks</td><td>Software tools</td></tr><tr><td>Distributed processing</td><td></td></tr></table>				Applications programs (computers)	Information flow	Client server systems	Man--computer interface	Communication middleware	Multidisciplinary design optimization	Computational fluid dynamics	Resource allocation	Computer networks	Software tools	Distributed processing	
Applications programs (computers)	Information flow														
Client server systems	Man--computer interface														
Communication middleware	Multidisciplinary design optimization														
Computational fluid dynamics	Resource allocation														
Computer networks	Software tools														
Distributed processing															
ABSTRACT Enterprises are confronted with the challenge to lower cost of product engineering. Engineers enter and leave enterprises. Much engineering know-how can be conserved for use by successive generations of engineers in documented engineering procedures, in software for simulation and in the supporting documents and simulation related data. 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 multidiscipline design, analysis and optimization of industry products.															



Contents

1 Introduction	5
2 A UNIX Network for Optimal Engineering Support	6
2.1 Computer Network Required for Engineering	6
2.2 The need to Present the Network to Engineers as a Unity	7
2.3 Network Middleware for Enterprise Enhanced Operation	8
3 The use of the UNIX Network and Middleware for Computational Fluid Mechanics	9
3.1 CFD work Environment in one Organisation	9
3.2 National CFD Working Environment	10
4 The use of the UNIX Network and Middleware for Multi-Discipline Analysis and Optimization	11
4.1 Problem Analysis	11
4.2 Set-up of MDO Environment	13
5 Effort to Effectuate an IT Supported Cooperative Working Environment	15
6 Concluding Remarks	16
7 References	16



This page is intentionally left blank



Network Middleware for Enterprise Enhanced Operation

W. Loeve, M.E.S. Vogels, H. van der Ven, E.H. Baalbergen

National Aerospace Laboratory (NLR), P.O. Box 90502,
1006 BM Amsterdam, The Netherlands

Abstract. Enterprises are confronted with the challenge to lower cost of product engineering. Engineers enter and leave enterprises. Much engineering know-how can be conserved for use by successive generations of engineers in documented engineering procedures, in software for simulation and in the supporting documents and simulation related data. 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 optimization of industry products.

Keywords. Network middleware, enterprise automation, engineering.

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 information has to be conserved in enterprises in the form of software such as for simulation, electronic documents (engineering procedures and manuals for simulation software) 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.

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. 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. This makes that IT is at least a required core competence of the enterprise. In chapter 2 a UNIX network and network middleware is described by which enter-



prise enhanced operation is possible. In chapter 3 and chapter 4 the application of the network and the middleware is illustrated for application of computational fluid mechanics and for application of a combination of computational fluid mechanics and computational solid mechanics respectively.

2 A UNIX Network for Optimal Engineering Support

2.1 Computer Network Required for Engineering

The most advanced industrially applicable computational fluid dynamics (CFD) software for aerodynamic analysis in aircraft engineering at the moment is based on the Reynolds-Averaged Navier-Stokes equations (Rubbert, 1994). One of the interesting applications of CFD 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. For NLR aerodynamic optimization is the most demanding engineering application for which it was decided to procure a NEC SX-4/16. This computer is meant to be applied in support of engineering and scientific simulation in The Netherlands (Loeve 1996). With a NEC SX-4/16 (a vector computer with 16 processors and shared memory) it is easy to realize for CFD in engineering a computational performance on 14 processors of 8,500MFLOP/s (Ven van der, 1997). This is realized with relatively little effort (only DO-loop parallelization). One aerodynamic optimization of 40 analysis runs then takes about 10 hours.

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 to 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, are commonly equipped with UNIX operating systems. Complexity of the computer networks and of the collections of software systems used in engineering processes both increase continuously in industry.

The computer network that serves the two establishments of NLR is presented in figure 1. The external connections indicated in the figure concern primarily the use of the SX-4. This computer is by far the most powerful computer in the Netherlands as can be seen in (Dongarra 1996).

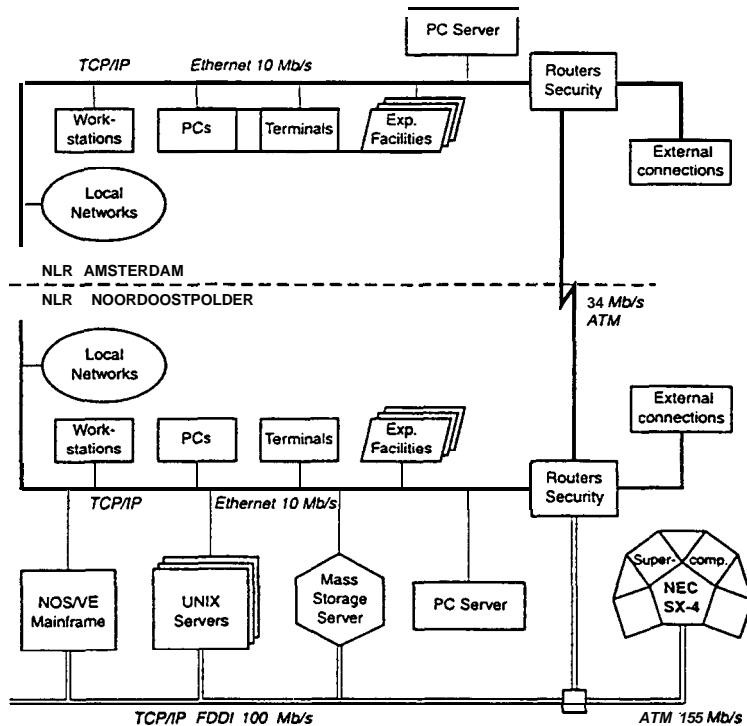


Fig.1 The NLR high performance computer network

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,
 - 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 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 engineers in the same way as if they all make use of one computer. Engineers have to be assisted in organizing, re-using and integrating a continuously growing amount of software, data and supporting help information in networks.

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 masons 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 always paid 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 facility called SPINE that supports the construction of general as well as application-oriented working environments (Loeve 1997). 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 customisation 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. 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 information management and middleware.

The desk top element of SPINE 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 tiles, 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 like 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.

SPINE is being further productized jointly by NEC and NLR. 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. At the moment SPINE is implemented in about 20 organisations for integration of local workstations in the organisations with the SX-4 at NLR. In some cases SPINE is just as in NLR used by other organisations for integration of not only user-workstations with the supercomputer but also with other servers.

3 The use of the UNIX Network and Middleware for Computational Fluid Mechanics

3.1 CFD work 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 ISNaS CFD working environment was aimed to support the use of flow simulation packages across the NLR network. The network consists of workstations and terminals that give access to mainframes acting as compute-, intranet-, internet- and data management servers. A supercomputer is integrated 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.

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 2. 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.

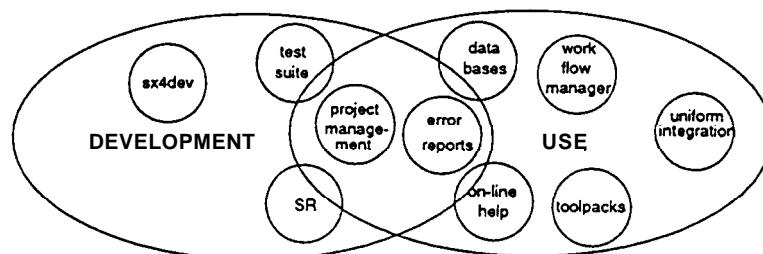


Fig. 2. Building blocks for both use and development of engineering software



In order to facilitate the use of remote resources NICE develops a nation-wide simulation centre 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' (see Section 3.1) 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 one single processor of 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 Consortium¹ is addressing integration of design and analy-

¹The MDO project (Multi-Disciplinary Design, Analysis and Optimization of Aerospace Vehicles) is a collaboration between British Aerospace, Acrospace, 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: BE952056).



sis tools creating a Multi-Disciplinary Optimization (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 monodisciplines 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.

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 $N \times N$ 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 3. The figure shows the involvement of the disciplines.

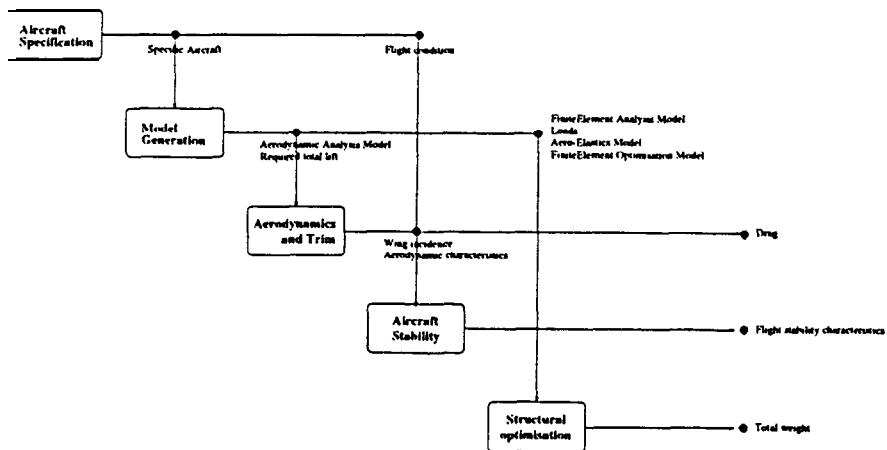


Fig. 3. Multi-discipline analysis process



The diagram is essential to avoid ambiguity in the analysis process. The diagram shows for example that the loads for the structural optimization in the present analysis process are the result of “model generation” and not of “aerodynamics and trim”. Such choices for the process should be explicit when interpreting the DOC value for an aircraft design. 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.

Changes in the mono-discipline software for the multidiscipline context

On the MDO level, it is assumed that all disciplines have their own (monodisciplinary) 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 “ismo” 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 NEC-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 MDO process model 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. This is the leading principle for the information engineering in the MDO environment. 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 4, three levels of the MDO environment are given.

From top to bottom we see in figure 4 first the highest level window which gives access to the environment including the tools. In the second window in figure 4 tool access is presented for version management as well as access to the specific MDO programs. The latter programs are shown in the bottom window with 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-optimization.

The diagonal elements in the MDO process (Figure 3) correspond to the "ismo" elements 2, 5, 4, 3, 7 respectively. The "ismo" elements 1 and 6 are additional

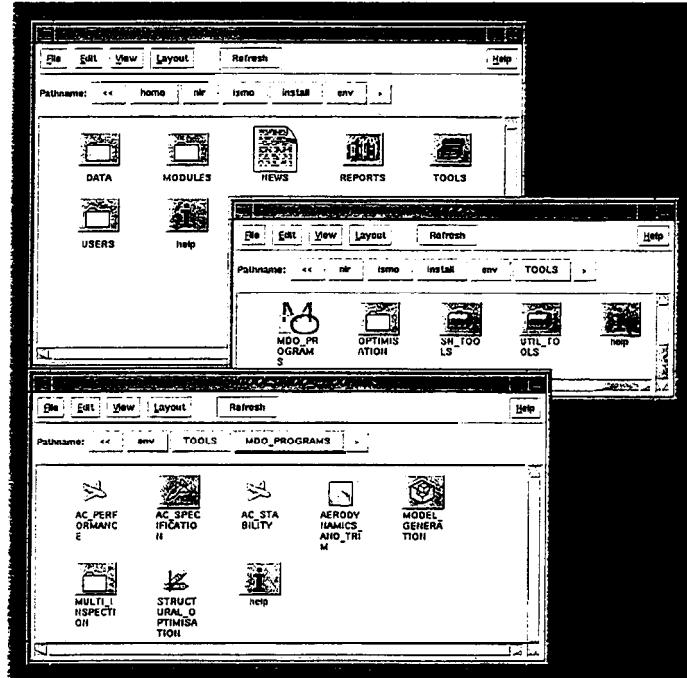


Fig. 4. Screendump of three upperlevels of MDO environment



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.

At each level engineers are supported by help information that is relevant for the concerning level. Manuals for instance can be accessed by dropping the icon representing a program on the help icon.

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 well as remote high performance computer as one single virtual computer for management, control, exchanging, accessing and using enterprise know-how.

7 References

1. C.J. Borland, J.R. Benton, P.D. Frank, T.J. Kao, R.A. Mastro and J.F.M. Barthelemy, (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 505519.
2. J.J. Dongarra, H.W. Meuer and E. Strohmaier, E (1996) TOP500 Supercomputer Sites November 1996; retrievable from internet by anonymous ftp to www.netlib.org/benchmark/top500.ps.
3. W. Loeve, (1976) A hierarchical network linking two research laboratories; Computer Networks 1 North-Holland Publishing Company.
4. W. Loeve, (1996) From parallel CFD to a tool for computer aided engineering; in *Parallel Computational Fluid Dynamics* Elsevier.
5. W. Loeve, E.H. Baalbergen, M.E.S Vogels and H. van der Ven, (1997) High performance computing in simulation that fulfils user-driven quality criteria; in *DASZA '97 Conference* ESA Publications Division.
6. MDO Consortium (1996) <http://www.telecall.co.uk/~srcbae/mdo>.
7. P.E Rubbett, (1994) CFD and the Changing World of Air-plane Design; AIAA Wright Brothers Lecture, Anaheim California; September 18-23.
8. H. van der Ven and J.J.W. van der Vegt, (1997) Experience with advanced CFD algorithms on NEC SX-4; to be published in *VECPAR '96 selected papers from the 2nd International Meeting on Vector and Parallel Processing (Systems and Applications)* (ed Laginha M. Palma et al), Springer.
9. M.E.S. Vogels and W. Loeve, (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.
10. M.E.S. Vogels, (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.