Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-98463

SPINEware: A framework for user-oriented and tailorable metacomputers

E.H. Baalbergen and H. van der Ven

Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-98463

SPINEware: A framework for user-oriented and tailorable metacomputers

E.H. Baalbergen and H. van der Ven

This report is based on an article to be published in Special Issue on Metacomputing of the International Journal on Future Generation Computer Systems, Elsevier Publ.

This report has been prepared under the Software Development Contract between NEC and NLR, signed on 19 April 1995.

The contents of this report may be cited on condition that full credit is given to NLR and the authors.

Division:	Informatics
Issued:	October 1998
Classification of title:	unclassified

Summary

Computing networks in enterprises are rapidly growing, not only in capacity, but also in complexity. The users usually get faced with networking details and with a variety of heterogeneous computing systems. In the present paper we present SPINEware as a facility for reducing the complexity of computer network usage.

SPINEware is a facility that supports the development of working environments on top of existing computer networks. Such a working environment reveals itself to the user as a powerful and easy-to-use single application environment - a metacomputer - providing uniform and network-transparent access to the resources and applications available from the computer network. To further reduce the complexity, SPINEware-based metacomputers may be tailored for particular end users and application areas.

Contents

5 Figures

(22 pages in total)

1 Introduction

In order to be more efficient and to lower the cost of product engineering and know-how management, today's enterprises constantly invest in their computer infrastructure [1]. To increase the computing power, new computers are added to the network, existing computers get replaced by more powerful systems, and network resources get replaced by faster and more advanced systems. As a result, however, the complexity also increases, not only at hardware and operating-system level, but also at end-usage level. Engineers using an enterprise's computing infrastructure are typically faced with low-level details emerging from the individual computers and the network involved. Even worse, they frequently need to familiarize with the ever-growing and changing computing infrastructure [2]. Also, the enterprise as a whole, using the computing power for managing their processes and competence, may suffer from the complexity and the inefficiency arising from it.

SPINEware is aimed to remedy the situation. It is based on the concept of providing the end user with a single and coherent application environment on top of a heterogeneous computer network, along with a powerful user-oriented and tailorable desk-top system. The application environment - henceforth called working environment - gives access to potentially all resources from the computer network. SPINEware is a collection of tools and software modules that supports the implementation of such working environments. It does not provide a ready-to-use metacomputer for the end user. SPINEware allows the creation of such metacomputer and its tailoring to the specific wishes of the end users. For this both knowledge on how to use the SPINEware tools and awareness on the end-user requirements is required by the working environment administrator. The emphasis of SPINEware lies on detailed elaboration on the end-user viewpoint on the one hand, and on the provision of a single through-the-desktop environment of the heterogeneous enterprise computer network resources on the other hand. The development of SPINEware started early 1992 at the National Aerospace Laboratory NLR (as a product called "SPINE"). SPINEware is being further developed since 1996 jointly by NLR and the Japanese computer manufacturer NEC.

In this paper, we present SPINEware as seen from an end user's technical point of view. The application of SPINEware to support process control and competence management is presented in [3, 4].

A rationale for SPINEware, which we consider is essential background information for a good understanding of the product, is presented in Section 2. The SPINEware product is described in Section 3. The current status of the product is described in Section 4. Section 5 presents the existing SPINEware-based metacomputer ISNaS as a case example. Section 6 contains concluding remarks.

2 Rationale for SPINEware

The development of SPINEware is largely motivated by the complexity arising with the use of today's computer infrastructures by end users who are not computer experts. To provide a solid basis for understanding the capabilities of SPINEware, we discuss this complexity in detail in this section. We conclude with a brief description of the SPINEware approach towards reduction of the complexity.

The complexity emerging from the computing infrastructures found nowadays in most industry and research enterprises is mainly due to the three following characteristics:

- *Distributed* - The computing infrastructure usually consists of several computers, located at one or several sites (depending on the enterprises geographical distribution), and usually connected through local-area and/or wide-area high-speed and high-bandwidth networks. For this reason, the computing infrastructure of an enterprise is usually referred to as the enterprise's *computer network*. For reasons of security, an enterprise may restrict and carefully control access to (subsets of) their computer network by using fire-wall techniques.
- *Heterogeneous* - A computer network typically comprises a variety of heterogeneous computers, including desk-top systems as well as centralized, generally accessible systems. The desk-top systems are usually considered personal systems, with a single system located near each end user. The desk-top systems include PCs (usually running a PC operating system such as Windows 95 or Windows NT), graphics workstations (usually running a UNIX-based operating system such as IRIX, HP-UX, and SunOS), and graphics ("X") terminals connected to some UNIX host system. The centralized systems usually include general-purpose as well as dedicated computing, file storage, and input/output servers. The computing servers typically include mainframes, supercomputers, and parallel computers. In addition to the hardware and operating-system variety, users of a computer network usually have to deal with a heterogeneous set of software applications and data formats.
- *Integrated at operating-system level* - The most striking characteristic an end user of a computer network is confronted with, is that the computers are integrated into the network up to the operating-system level, rather than at end-user level. The operating systems running on the computers usually provide TCP/IP-based network utilities such as TELNET (remote login), FTP (file transfer), RSH (remote command execution), and NFS (networked file system) to facilitate access to other computers. However, except for products such as NFS which enable network-transparent access to remote files by the end user, most of these utilities still require explicit control by the end user. Consequently, the user - although inter-

acting with the computer network from a single desk top - still sees the individual computers from the desk top; see Figure 1.

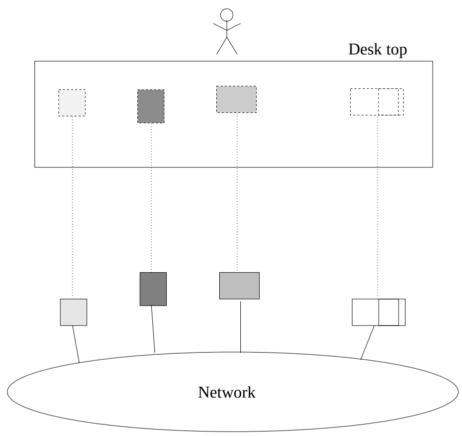

Fig. 1 The user of a computer network usually sees the individual computers from the desk top.

To summarize, today's computer networks are system oriented rather than end-user oriented. The visibility of the network and operating-system details to the end user certainly increases the complexity of computer networks. Users may have the impression that they need to adapt to the system rather than vice versa: that the system is adapted for their needs, which would be ideal. From the end user's point of view, the following aspects resulting from the system orientedness, contribute to the complexity:

- The end user accessing a remote computer (e.g., via TELNET or some other remote login facility) usually gets faced with the native operating system and command interpreter of the remote computer. Consequently, the user may be confronted with a variety of native systems. For example, a PC user familiar with Windows accessing a supercomputer running some UNIX flavor may find the UNIX command-line interpreter (usually the Bourne shell or some derivative) cumbersome.
- The end user usually has to deal explicitly with networking details when, for example, starting a program on a computer other than the desk-top system. The details include selection of the proper computer to run the program, transferring input files to and output files from the remote computer, and remote login or execution. Also, in wide-area computer networks, the end user may have to deal with fire walls, which may involve the use of different networking utilities, such as SSH (Secure Shell) instead of RSH (Remote Shell).
- The user usually gets confronted with details resulting from hardware heterogeneity (i.e., different byte ordering, word sizes, and floating-point representations), and software heterogeneity (different ways to start programs with specification of input files and options,

different styles of user interaction, different data representations).

- The user is typically confronted with a potentially huge amount of information - including software, data, documents - that is much larger than available from a single computer. This has impact on the way to organize, access, and manage the information.

The goal of SPINEware is to provide a solution that provides the end user with access to potentially all resources available from the computer network, thereby reducing the complexity by hiding the network details and providing uniform access. An important requirement that motivated the development of SPINEware is that the solution is *total*, in that it covers all of the aforementioned aspects that contribute to complexity, and that the solution is *end-usage oriented*, that is, easy to use by, and even tailored for, end users from a particular application area who are not computer experts [5]. As such, SPINEware aims to support the ideal of adapting the system to the end user.

The basis for the SPINEware solution is the *working environment*, which provides the end user with a *single environment* (a “metacomputer”) on top of the computer network, together with a powerful end-user oriented and tailorable desk-top system. SPINEware is a facility that supports development of such working environments. One of the guidelines applicable to the development of SPINEware is the reuse of existing software - including TCP/IP-based networking tools - whenever possible and applicable. Resulting from this, SPINEware may be considered “glueware”, providing the glue to turn a computer network, and the software and any other information available from the network, into a single and coherent system.

3 Description of SPINEware

In this section we will present the product SPINEware and its solution - in the form of working environments - towards reduction of the complexity of computer network usage by computer non-experts. We first describe the notion of a (SPINEware-based) working environment, which is the basis for presenting a computer network to the end user. Next, we describe the SPINEware product itself. Finally, we describe how SPINEware is actually used in development and usage of working environments.

3.1 SPINEware-based working environment

A (SPINEware-based) working environment is a system that presents a computer network to its users as a single, user-oriented application environment. We will have a closer look at the working environment by considering its key features: the single computer notion (i.e., transparency of networking details), being tailorable for particular end usage, and its graphical user interface (GUI).

The first key feature of a working environment is that it provides its end users with access to the resources from the network as if the resources were available from a single computer (i.e., metacomputing). The resources include computing power, data storage, and input/output facilities, as well as information present in the form of software, data, and documentation. As opposed to the situation depicted in Figure 1, in which the end user still is confronted with the individual computers being operated from a single desk-top system, the end user of a working environment sees a single computer; see Figure 2.

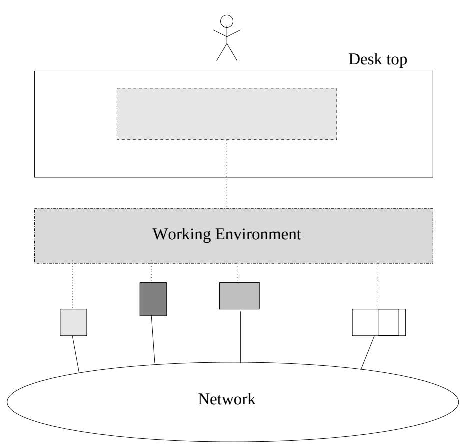

Fig. 2 The user of a working environment sees all resources from the computer network as if these were located on a single computer.

For the user of a working environment, the information from the network is organized into a single virtual directory tree, containing files, (sub) directories, and tools. (A *tool* actually is a regular file which has been designated as tool; see Section 3.3 for details.) The directory tree may span (parts of) the file systems of the interconnected computers, thus providing easy access to native files and directories. It allows files, directories, and tools to be arranged in a way that is most appropriate for specific end usage. Although not visible for the end user, the directory tree is implemented using UNIX facilities such as symbolic links and NFS wherever possible. FTP may be used in cases NFS does not apply. The user can browse the virtual directory tree, and manipulate its contents, using the file-browser facility from the GUI, which hides the machine boundaries and other implementation details.

The second key feature is that a working environment can be easily tailored, accommodated, for particular end usage, such as specific application areas or groups of users. The tailoring consists of defining (and modifying) an end-user oriented “view” on the network, to organize, filter, control, and present the available resources in a way that is most convenient for the end user. A view is defined in terms of the virtual directory tree and its contents, together with a description of how to present the contents to the user (in terms of icons) and how to react on icon operations by the user. In the directory tree, the applicable tools, data, and documentation may be arranged in a suitable way for the end users. In addition, the SPINEware information management tools (see Section 3.2) may be linked into the directory tree, and hence may be made available in a convenient way.

The third key feature is the working environment’s GUI, henceforth called the *User Shell*. The basic facility of the User Shell is the file browser, which enables the user to browse the virtual directory tree defined for the working environment. The contents of a directory are presented to the user as a set of icons in a window. Using the file browser, the user is able to apply the usual file and directory manipulation operations (e.g., open, copy, move), and to activate tools, through point-and-click and drag-and-drop operations on icons and windows. This mode of user interaction - originating from the Apple Macintosh - is a widely accepted solution for user-friendly computer interaction, as can been seen from the popularity of PC Windows systems.

A tool, which may run on a different computer, and which may involve file transfer and data conversions, can simply be activated by the end user by double-clicking its icon, or by dropping a selected set of input files on its icon. Any options can be specified by filling out values in a “tool option form” that is popped up in response. Help information about specific tools or files can be obtained by dropping the corresponding icons on a help button, which is an icon representing the SPINEware *help* tool.

The User Shell also provides the *Workflow Editor* for data-flow driven computing: tools and placeholders for data files (“data containers”) may be organized into a so-called “tool chain”. The Workflow Editor supports easy construction of tool chains by providing a canvas for arranging the tools and data containers as icons, and for specifying flows as arrows between icons. It also controls the execution of tools activated from the chain. As such, the Workflow Editor provides the basic building blocks for definition of work flows.

3.2 SPINEware

SPINEware supports the development of the working environments as described in the previous section. In more concrete terms, SPINEware is a tool kit, a collection of tools and software modules that support the implementation of working environments. In this section, we will give a global overview of SPINEware in terms of its tools and modules. In Section 3.3, we will concisely describe how SPINEware is used.

The SPINEware tools and modules are subdivided into four components: User Shell, Working-Environment Management System, Middleware, and a collection of Information Management Tools. These components are described below.

The *User Shell* is a collection of software modules that together constitute the GUI of a working environment. It provides a working-environment *control panel* for messages, for customizing global GUI settings, and for starting other SPINEware functions. The User Shell provides *file-browser windows* for browsing directories from the virtual directory tree. It also provides *tool-option forms* for interactive specification of options at tool launching. The User Shell finally provides the *Workflow Editor* for specification and manipulation of tool chains (see Figure 3 for an example of a tool chain for a CFD application).

The *Working-Environment Management System* supports the construction of working environments. It has tools for creating an initial working environment (with minimum contents and functionality), and for extending and modifying working environments. This functionality is described in more detail in Section 3.3. The basic purpose is to glue together the resources from the computer network and the other SPINEware components (e.g., User Shell) into a single and coherent metacomputer. In addition, the Working-Environment Management System provides the *Tool Integrator* that supports the integration of tools in working environments, which is one of the basic tailoring activities as we point out in Section 3.3. It also provides support for the construction of *tool packs*, which are collections of tools serving a particular purpose, and which can easily be imported in a working environment. The notion of tool pack is discussed also in Section 3.3.

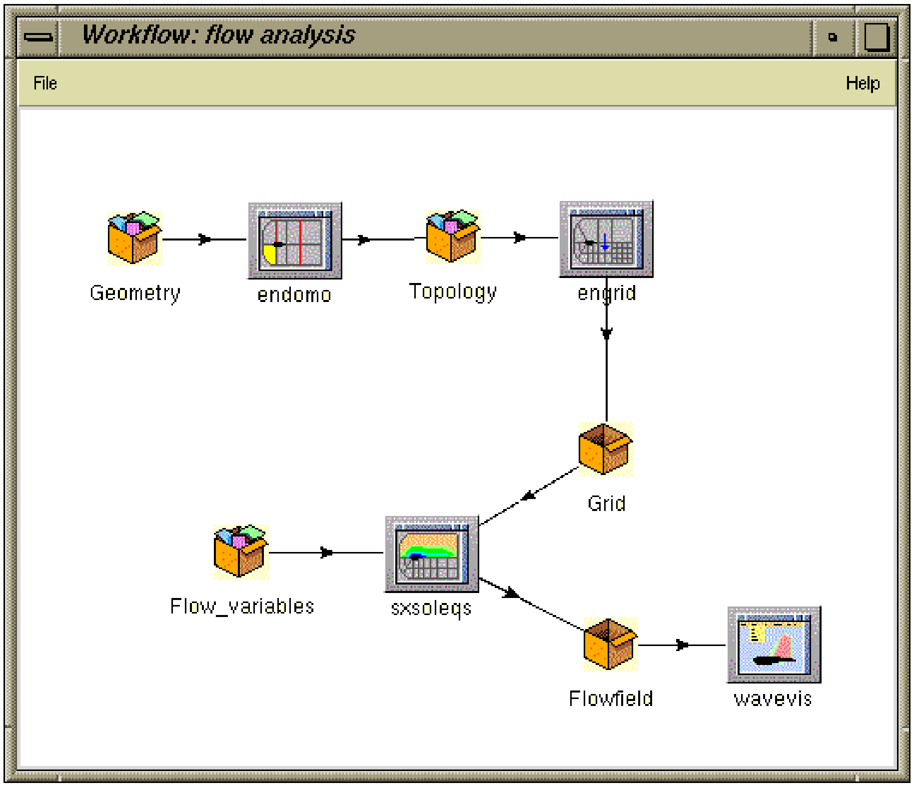

Fig. 3 An example of a tool chain for CFD: tools for pre- and postprocessing are connected with the flow solver.

The SPINEware *Middleware* supports the network transparency during usage of working environments. Its primary target is to support transparent remote execution of programs on different computers, thereby hiding details on transfer of input and output files - if any and if necessary - and the actual remote execution. This functionality lies at the basis of tool integration. To deal with (local-area as well as wide-area) security measures, the middleware is accommodated for use of Secure Shell (SSH), which is a widely accepted tool for secure access to remote resources.

The SPINEware *Information Management Tools* are organized into a collection of tool packs which are readily available for inclusion in working environments. The collection includes: SR for software source-code version control (a wrapper for RCS - the Revision Control System); DFMS for management of data files; E-FORMS for definition and manipulation of WWW-based electronic forms; E-SEARCH for definition of WWW-based search engines; COMMON for general file and directory manipulation (beyond simple move and copy); and SX4DEV for developing and targeting software for an NEC SX-4 supercomputer. In addition, a tool CONVERT is provided for automatic selection - from an available set of data-conversion tools - and execution of a series of conversion tools to convert a specified data set into the specified format.

3.3 How to use SPINEware in developing working environments

In the previous section, we have introduced the Working-Environment Management System as the SPINEware facility that supports the development of working environments. In this section, we will concisely describe the steps towards construction of a working environment.

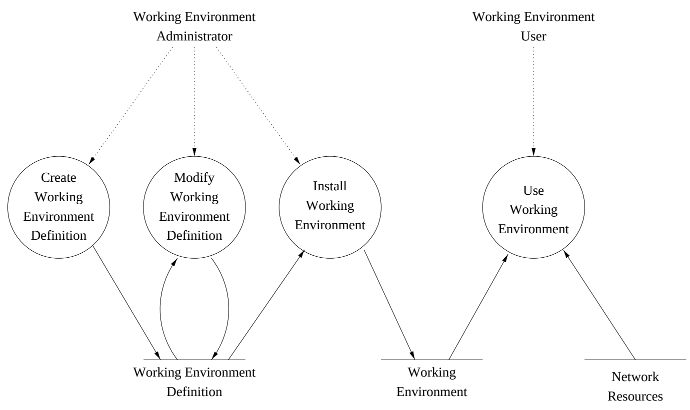

Fig. 4 The process of developing a working environment using SPINEware.

The process of developing a working environment is globally depicted in Figure 4. The figure first shows that two different kinds of users are involved. The *Working Environment Administrator* is responsible for creating the working environment, which can be used by the *Working Environment User*, who is the actual end user as referred to throughout this paper. The figure next shows, in the form of four bubbles, the basic activities involved in working-environment development and usage. The “Create”, “Modify”, and “Install” activities are supported by the Working Environment Management System; the “Use” activity is covered by the other SPINEware components presented in Section 3.2. The figure also shows the two information sets involved: the *Working Environment Definition* and the *Working Environment* proper. The definition is the “source code” of the working environment, and consists of a small tree of directories with text files which may be edited by the administrator. The working environment proper is the installed version of the working environment as available to the end users.

The solid arrows from Figure 4 globally describe how a working environment is set up, starting from nothing. First, as represented by the “Create” bubble, the definition of an initial working environment with minimum functionality is generated. The initial working environment provides a minimum virtual directory tree, containing some basic tools such as a text-file editor. Without any modification, the initial working environment can be installed (cf. the “Install” bubble) by simply typing “`make install`” in the top-level directory in the definition. Upon termination of this command, the working environment is installed, and is readily available for “Use” by the end user. The installed working environment consists of a tree of directories, which comprises all information to realize the metacomputer. The subdirectory `bin` contains the entry point for users of the working environment: a script, which is named after the working environment as determined during “Create”, is present to start the User Shell of the working environment.

The key activity in working environment development is “Modify”, which consists of modifying and extending the definition, and hence - upon “Install” - the working environment, in order to tailor it for the end user. First of all, it is necessary to “design” the working environment: to determine the contents of the working environment for end usage. This requires expertise in the application field, and the involvement of prospect end users. This “design” process is supported by SPINEware in that ideas can rapidly be translated into extensions and modifications of the definition, and hence can rapidly be demonstrated. Section 5 describes an example design process and its results: the ISNaS working environment tailored for numerical flow simulation.

Modification of a definition primarily consists of changing existing, and adding new, text files in the definition. The possible modifications are:

- The specification of host and site specific *configuration parameters*, such as path names of native programs, and internet names of specific computers. The configuration parameters are centralized in a few text files.
- The setting up and modification of the directory structure of the installed working environment, which includes the virtual directory tree as presented in Section 3.1. The directory structure is defined in a single text file, using a straightforward and intuitive language.
- The definition of icons to be used by the User Shell’s file browsers to represent entries in a directory, together with the “semantics”: a specification of the actions to initiate upon user operations on the icons. The icons with the semantics to be used for certain classes of directory entries (specified in terms of file type and file-name pattern) are specified as rules in a text file.
- The integration of tools. The purpose of tool integration is to make native - commercial as well as home-brew - programs available in a working environment as tools, which can be represented as such by the User Shell, and which can be contained in tool chains. SPINEware uses the “wrapper” method, which means that a native program is integrated by putting a wrapper around it in order to make it fit into a working environment, while leaving the program proper intact (i.e., without having to modify and recompile it). A wrapper takes care of all execution details, and usually uses the SPINEware middleware to deal with

possible remote execution.

To facilitate tool integration, SPINEware provides the Tool Integrator. This utility has a GUI that enables the user to specify the information needed to further automatically generate the wrapper. The Tool Integrator is available from a working environment, and enables a tool to be integrated in the working environment instantaneously (i.e., the result is immediately visible and usable in the working environment), thereby also applying the necessary modifications to the working-environment definition for use in subsequent installations.

- The import of tool packs. A tool pack is some sort of semi-manufactured working environment in which tools and related information may be integrated, and which can be imported easily in working environments, in order to integrate the contents of the tool pack in those working environments. The import is achieved easily by adding a single command to the `Makefile` in the top-level directory of a working environment.

4 Current status of SPINEware

Versions of SPINEware have been internally available at NLR since 1992. Since then various working environments have been realized, among which for numerical flow simulation (ISNaS [6]; see also Section 5), for computer-aided control engineering (ISMuS [7]), and for computer-aided software engineering (ISEnS). Efforts to hide the broad heterogeneity of the NLR computing network, the complexity arising from this, and feed back from users of these environments contributed to the general applicability of the product.

In the past years, SPINEware has been used for the realization of several working environments by NLR and its partners in several national and international projects, such as NICE [8], FASTFLO [9], and MDO [10].

In 1995, NEC recognized SPINEware as a valuable tool for supporting the integration of their SX supercomputers - running its own variant of UNIX: SUPER-UX - in existing computer networks such as described in Section 2. Since early 1996, SPINEware has been further developed jointly by NLR and NEC. Early 1998, NEC announced SPINEware as one of their program products.

SPINEware is currently available for most popular UNIX workstation systems, such as IRIX, HP-UX, and SunOS. The code is targeted for the de-facto UNIX standard System V Release 4, which answers for portability to other UNIX systems as well. Although the present version of SPINEware (2.0) is targeted for UNIX, it is possible to operate a working environment (running on some UNIX host system) from a PC, which has an X server (and preferably NFS software) installed.

Use of a SPINEware-based working environment in a computer network requires SPINEware and the working environment to be installed on the desk-top computer (or a UNIX host available from the desk-top system) only; the other computers are accessed using the TCP/IP network utilities already found on most UNIX computers nowadays.

5 A case example: CFD working environment ISNaS

SPINEware is a tool kit that supports the implementation of working environments. SPINEware does not provide a ready-to-use metacomputer for the end-user: using SPINEware such a metacomputer can be created, tailored to the specific wishes of the end users. The working environment administrator, responsible for the development of a specific working environment, thus requires both knowledge on how to use the tools in the tool kit SPINEware, *and*, as important, awareness on the requirements of the end users.

The development of a working environment is dominated by the following two principles:

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

Hence, the user is the central theme in the development of the working environment. This theme also implies that an integrated tool does not stand alone in the working environment: it is surrounded by tools that either supply input or use the output of the specific tool. This uniform integration of several tools greatly facilitates their use. SPINEware provides the working environment administrator with a toolkit to conform the working environment to the end users' wishes.

Since the earliest developments of SPINEware it has been applied to ISNaS, a working environment for flow simulations (CFD: Computational Fluid Dynamics). At first, the working environment was aimed to support the use of flow simulation packages across a local area network consisting of a supercomputer, a mainframe, workstations and terminals. Because of its success, more functionality was - and still is - added.

ISNaS now supports both use and development of simulation software (see Figure 5 for a partial view of the working environment). The use of the simulation software is supported by integrating the entire pipeline of simulation analysis: geometry modelers, 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 SPINEware facility for implicit file conversion. Exchange of documents and data is made possible through easy-interface databases. Feedback to the developer of the working environment is made easy by the use of electronic error reports.

SPINEware supports the development of simulation software by providing a software version management tool and a framework for setting up a regression test suite. Development on, and use of the NEC SX-4 is supported by the SX4DEV tool pack. This tool pack integrates tools for compiling,

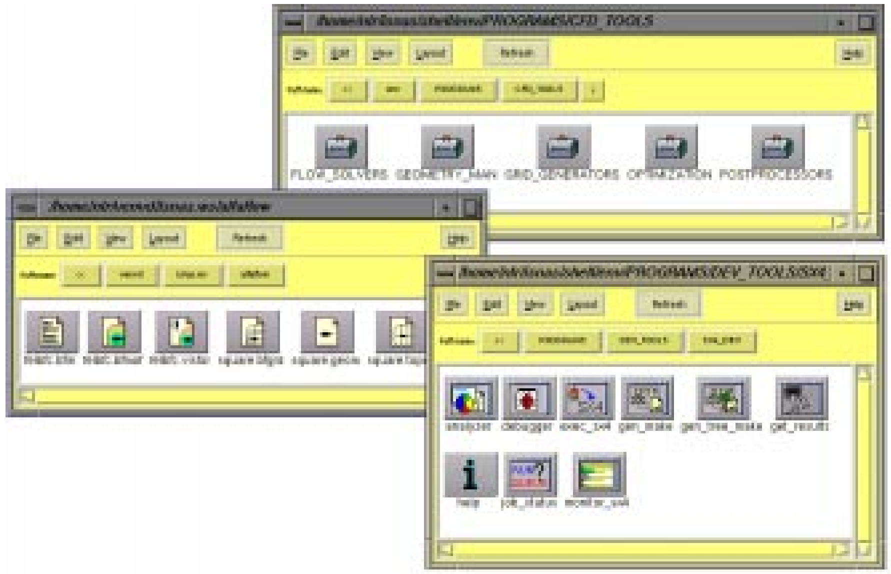

Fig. 5 A view of some browser windows of the working environment ISNaS. The upper window shows the various tool boxes for the complete pipeline of flow simulation. The right window shows the tools in the SX4DEV tool pack. The left window shows the contents of a directory of the user.

executing, analyzing and debugging source on the NEC SX-4. Automatic `Makefile` generation allows for easy manipulation of the source. Specific options of the various tools are based on the experience of experts in the field of supercomputing. Using this tool the computational kernel of a flow solver from the Institute of Applied Physics of the Netherlands Organization for Applied Scientific Research (TNO-TPD) was compiled, executed and analyzed within a quarter of an hour. Before that, TNO-TPD had never worked on an NEC supercomputer.

For both users and developers project management systems can be set up using the tools in SPINEware. 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.

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.

The working environment has proven to be highly portable as well. For a nation-wide project, NICE [8], which promotes the use of CFD through HPCN to industry, ISNaS has been ported to a wide-area network consisting of companies' sites and NLR's NEC SX-4. Only a small set of configuration parameters needed to be modified: re-use was immediate.

The working environment has developed over the years, through requests from the end users. SPINEware has made this development easy by providing the working environment administrator with a toolkit which greatly facilitates the translation of user demands into the working environment. Development and user support for a specific working environment typically takes four man weeks a year. This is a small investment compared to the improved efficiency of the end users.

6 Concluding remarks

In this paper we presented SPINEware and its notion of working environment as solution for reducing the complexity emerging from the end usage of computer networks found nowadays in most enterprises. The applicability of SPINEware has been proven by the several working environments that exist, and the various projects that use, or plan to use, SPINEware for the realization of intra-enterprise as well as inter-enterprise working environments. Its capabilities to also accommodate for security in wide-area networks even render it possible to use SPINEware for realization of so-called “extended enterprises” [4].

An important conclusion of the project is that the total metacomputing solution in combination with possibilities for end-usage tailoring is an ideal basis for using today’s computer networks.

Near-future developments in the project will focus on

- the application of object-based techniques and CORBA concepts and products to increase the flexibility and interoperability of the product
- full availability and functionality of SPINEware in networks that include UNIX systems as well as PCs running Windows 95 or Windows NT, and
- end-user oriented job management.

References

1. W. Loeve, M.E.S. Vogels, H. van der Ven, E.H. Baalbergen, Network middleware for enterprise enhanced operation, in: Computer Applications in Production and Engineering, proceedings of IFIP TC5 International Conference on Computer Applications in Production and Engineering (CAPE'97, 5-7 November 1997, Detroit, Michigan, USA), Chapman and Hall 1997.
2. W. Loeve, E.H. Baalbergen, M.E.S. Vogels, H. van der Ven, High performance computing in simulation that fulfills user-driven quality criteria, in: Proceedings of the DASIA'97 Conference Data Systems in Aerospace (Sevilla, Spain, 26-29 May 1997), European Space Agency 1997.
3. W. Loeve, M.E.S. Vogels, Competence management in engineering environments with HPCN, in: Proceedings of the international conference and exhibition on high-performance computing and networking (HPCN Europe 1998), Springer Lecture Notes in Computer Science 1401, April 1998, pp. 13-22.
4. M.E.S. Vogels, W. Loeve, Know-how management in distributed engineering environments, Prolamat'98 conference, 9-11 September 1998, Trento, Italy.
5. E.H. Baalbergen, SPINEware: a practical and holistic approach to metacomputing, in: Proceedings of the international conference and exhibition on high-performance computing and networking (HPCN Europe 1998), Springer Lecture Notes in Computer Science 1401, April 1998, pp. 1008-1011.
6. M.E.S. Vogels, W. Loeve, Development of ISNaS: an information system for flow simulation in design, in: Computer applications in production and engineering, F. Kimural ed., North Holland, 1989.
7. M.J.H. Couwenberg, R.J. Cazemier, NLR's CACE working environment ISMuS, in: Proceedings of the IEEE int. symp. on computer-aided control system design, Dearborn, MI, USA, 15-18 September 1996.
8. R. Groothuizen, H. van der Ven, A NICE HPCN centre for flow simulation, in: Proceedings of the international conference and exhibition on high-performance computing and networking (HPCN Europe 1998), Springer Lecture Notes in Computer Science 1401, April 1998, pp. 23-29.
9. J.W. van der Burg, B. Oskam, FASTFLO - automatic CFD system for three-dimensional flow simulations, 7th international conference on industrial technologies and 3rd aero days, Toulouse, France, 27-30 October 1997.
10. M.E.S. Vogels, P. Arendsen, R.J. Krol, M. Laban, G.W. Pruis, From a monodisciplinary to a multidisciplinary approach in aerospace: as seen from an information and communication technology perspective, 21st ICAS Congress, Melbourne, Australia, 1998.