



NLR-TP-98462

A tool kit for turning heterogeneous computer networks into application environments

E.H. Baalbergen and H. van der Ven



NLR-TP-98462

A tool kit for turning heterogeneous computer networks into application environments

E.H. Baalbergen and H. van der Ven

This report is based on an article to be published in NEC Research & Development, October 1998.

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

SPINeware is a tool kit that supports the development and operational use of user-oriented application environments on top of heterogeneous computer networks. Such an environment presents itself to its end users as a single and powerful "metacomputer" with a user-friendly interface that provides uniform and network-transparent access to the resources, such as computing power, data storage, and applications, available from the network. SPINeware also facilitates the tailoring of an environment to a particular application area. It provides experts in that area with tools to translate their computer experiences and expertise into a ready-to-use working environment for that area. In this paper, we describe a SPINeware-based working environment and how SPINeware supports the realization of such an environment. As a case example, we describe SX4DEV, a SPINeware tool pack in which the expertise of a developer of software targeted for the NEC SX-4 supercomputer has been consolidated in a form that is ready for use in a SPINeware working environment.



Contents

1	Introduction	5
2	Working Environment	8
3	Description of Spineware	12
	3.1 Product Overview	12
	3.2 Using Spineware for the Realization of Working Environments	13
4	Tool Pack SX4DEV	16
5	Concluding Remarks	18
6	References	20



1 Introduction

The application of computing power in today's enterprises is increasing rapidly. During the past five decades, the computing infrastructure has evolved from "programmable" calculator operated by specialists, into a powerful, general-purpose computer network, accessible to almost the entire staff of the enterprise. Today's networks have become key tools in supporting product-engineering processes, such as simulation and design-process control. An enterprise's engineering expertise is usually translated into information, in the forms of software, data, and documentation, available for immediate use, from the enterprise's computer network. The network has become a valuable competence management tool, supporting an enterprise in reusing and exploiting its own expertise (Ref. [1, 2]).

In order to lower the cost of product engineering, enterprises continuously invest in their computing infrastructure. New computers are added, existing computers are replaced by more powerful systems, network resources are replaced by faster and more advanced systems, and new applications - requiring even more computing and storage capacity - are installed. Although the potential computing power increases, the complexity also increases, at hardware and operating-system level as well as at end-usage level. Engineers using the computing power are typically faced with low-level details emerging from the use of the individual computers and the network involved. Even worse, they frequently need to familiarize with the ever-growing and ever-changing computing infrastructure. Consequently, the enterprise may suffer from the inefficiency arising from the (increasing) complexity.

Two characteristics of today's computing infrastructure contribute to the complexity. First, the infrastructure usually consists of a network comprising a heterogeneous set of interconnected computers, located at one or more sites. Unfortunately, integration of the network resources has taken place at the operating-system level, rather than at the end-usage level. End users must explicitly access the individual computers in a network, usually by using TCP/IP-based network utilities, such as TELNET, FTP, NFS, and RSH, available from the desk-top computer. Most of these utilities require explicit control by the end user. (NFS is an exception; it provides network-transparent access to remote files.) As a result, the user, although sitting behind a single desk top, still sees the individual computers (Fig. 1). The user is faced with low-level networking details, such as selection of remote computers for a specific application, remote login or execution, different command-line interpreters and commands, data conversions (resulting from

hardware and software heterogeneity), and file transfer between computers. The presence of fire walls may even make things more complex. It may require the user to select different utilities, such as SSH (Secure Shell) instead of RSH.

A second characteristic that contributes to the complexity is the potential access to an amount of information that is orders of magnitudes larger than found on a single computer. The user is often confronted with the need to organize and control access to the relevant information, to avoid information drowning.

SPINEware aims at reducing the complexity arising from today's computer networks (Ref. [3, 4]). Its development is motivated by two key requirements: the solution shall be *total and end-usage oriented*. Total means that the solution covers all networking and heterogeneity details, rather than only a subset of these. End-usage oriented means that the solution is easy to use by, and may be tailored for, end users from a particular application area, who are not computer experts.

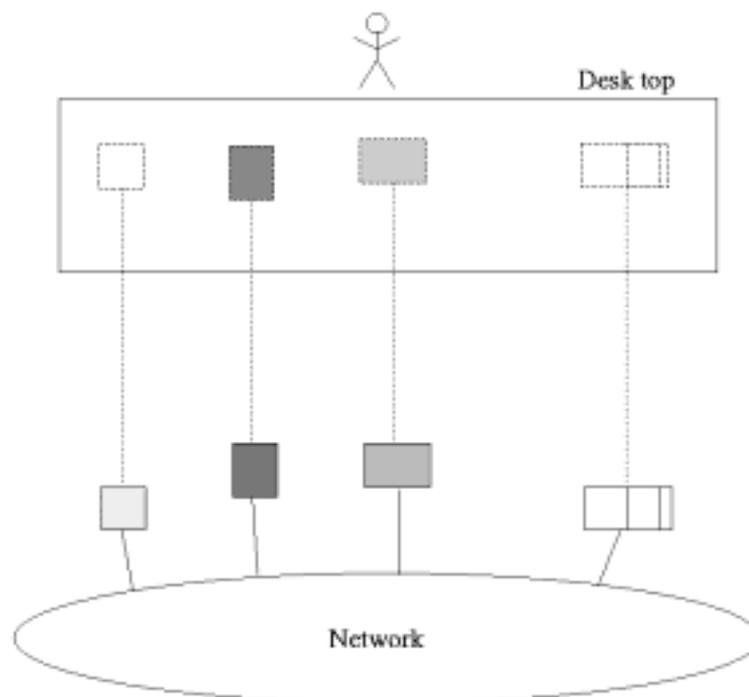


Fig. 1 The user of a computer network usually sees the individual computers from the desk top.



The SPINEware solution is based on the notion of *working environment*. A working environment provides the end user with a single computer (a ``metacomputer"), with a powerful and user-oriented graphical user interface. It gives uniform access to potentially all resources from the network, thereby making any details of network resources transparent. A working environment may be accommodated for use by a particular group of users and for specific application areas. SPINEware supports the construction and tailoring of such working environments.

In the remainder of this paper, we describe SPINEware, starting from a description of a SPINEware-based working environment in Section 2. In Section 3, we present the key components of SPINEware, and describe how these are used for development and usage of working environments. In Section 4, we describe the SX4DEV tool pack as a case example of how expertise in the field of developing software for the NEC SX-4 supercomputer has been made available for use in working environments. Section 5 contains concluding remarks.

2 Working Environment

A SPINeware working environment is a system that provides the users of a computer network with a single, user and/or application-area oriented, environment with a powerful graphical user interface (GUI), which gives uniform access to potentially all resources from the network. In this section, we will have a closer look at the working environment by considering its key aspects: the single computer look-and-feel, the tailorability for particular end usage, and the GUI.

A key aspect of a working environment is that it provides its end users with access to a computer network's resources as if the resources were available from a single computer. The resources include computing power, data storage, I/O facilities, and information in the form of software, data, and documentation. Details emerging from the network and the heterogeneity involved with access to the resources are hidden from the user. Instead of a set of loosely-coupled individual computers (cf. Fig. 1), the end user sees a single computer (Fig. 2).

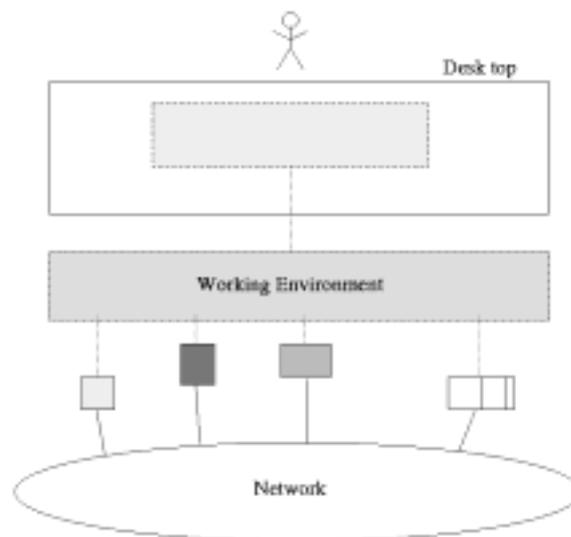


Fig. 2 The user of a working environment sees the resources from the computer network as if these were located on a single computer.

A working environment organizes the information from the network into a single virtual directory tree containing files, directories, and tools. The tree may span (parts of) the file systems of the interconnected computers, and hence provides easy access to native files and



directories. Although transparent to the user, the tree is implemented using off-the-shelf UNIX facilities such as symbolic links and NFS wherever applicable and possible. This tree makes it possible to arrange directories, data files, applicable tools, and documentation in a way that is most appropriate for specific end usage. The user can browse the tree, manipulate its contents, and launch tools using the file-browser functionality from the GUI, without dealing with machine boundaries and other implementation details.

Another key aspect is that a working environment can easily be tailored for particular application areas, users, and groups of users. The tailoring consists of defining a "view" on the network, through which the available resources may be organized, filtered, controlled, and presented in a way that is most appropriate for the intended end usage. A view basically defines the structure and contents of the virtual directory tree. It specifies how the GUI presents the contents to the user (that is, which icons to use), and how the GUI responds to icon operations (such as "open" and "drag-and-drop") by the user. The view also contains tools. SPINeware supports the integration of native programs and applications in a working environment as tools, to allow them to be manipulated and started in a uniform and easy way. SPINeware also provides a set of basic, general-purpose information-management tools, which are ready for installation in a working environment (see Section 3.1).

The third key aspect is the GUI, the so-called *User Shell*. The basic facility of the User Shell is the file browser. The browser enables the user to browse the virtual directory tree defined for the working environment (Fig. 3). It presents the contents of a directory as a set of icons in a window, and it enables the user to apply basic file and directory operations (for example, open, copy, move), and to activate tools, through point-and-click and drag-and-drop operations. A tool, which may or may not run on a different computer, and which hence may involve file transfer and data conversions, may be activated by the end user in a straightforward way, either by double-clicking its icon (if no input files are involved), or by dropping a selected set of input files on its icon. Options can be specified by filling out values in a "tool option form" that is popped up upon tool activation. On-line help information about a tool or file is displayed upon dropping the icon on an icon representing the SPINeware *help* tool.

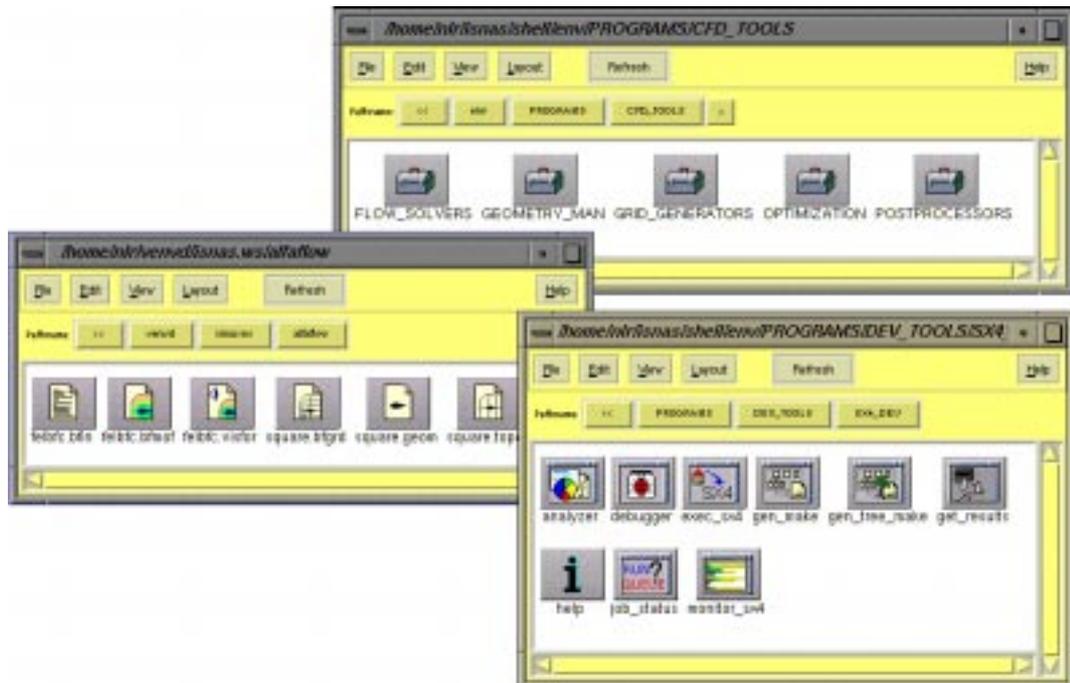


Fig. 3 Several file-browser windows of the working environment ISNaS. The upper window shows various tool boxes for the complete pipeline of flow simulation. The lower right window shows the tools in the SX4DEV tool pack. The left window shows the contents of a directory of the user.

The User Shell also provides the *Workflow Editor* for data-flow driven computing. It allows tools and placeholders for data files (*data containers*) to be organized into a *tool chain*. The user may interactively construct a tool chain by arranging icons representing tools and data containers on a canvas, and by drawing arrows between the icons to represent data flows. An example of a tool chain for a CFD (Computational Fluid Dynamics) application is shown in Fig. 4. The Workflow Editor also controls the execution of tools activated from the chain. In combination with appropriate information-management tools, it may be used as basic tool for defining work flows.

An example working environment is ISNaS (Ref. [5]), a working environment for CFD. SPINeware has been applied to ISNaS since its earliest versions. At first, ISNaS was aimed at supporting the use of CFD 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.

The success of ISNaS 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 analyze a specific flow configuration within one week. The analysis consisted of the entire pipeline from preprocessing up to visualization.

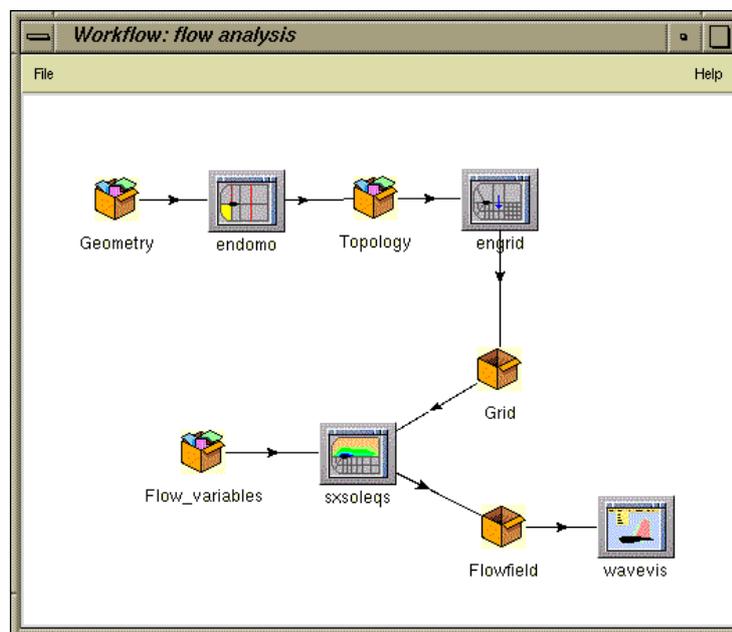


Fig. 4 An example of a tool chain for CFD: tools for pre- and postprocessing are connected with the flow solver.

ISNaS 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 that facilitates the translation of expertise and 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.



3 Description of Spineware

SPINeware supports the realization of the working environments described in the previous section. More precisely, SPINeware is a tool kit - a collection of tools and software modules - that support the development of working environments. In this section, we will give a global overview of the SPINeware product, and we concisely describe how SPINeware is used.

3.1 Product Overview

The SPINeware product comprises four components: User Shell, Working-Environment Management System, Middleware, and a collection of Information- Management Tools.

The *User Shell* is a collection of software modules that together constitute the GUI of a working environment. It provides a 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 provides tool-option forms for interactive specification of options at tool launching. It provides the Workflow Editor for specification and manipulation of tool chains.

The *Working-Environment Management System* supports the implementation of working environments. It has tools for creating an initial working environment (with minimum contents and functionality), and for extending and modifying working environments. Its basic purpose is to glue together the resources from the computer network (including native networking utilities) and the other SPINeware components (e.g., User Shell) into a single and coherent metacomputing system. The Working-Environment Management System provides the *Tool Integrator* to support the integration of tools in working environments, which is considered the prime tailoring activity. It also supports 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 role of the Working-Environment Management System is described in detail in section 3.2.

The SPINeware *Middleware* supports the network transparency required for the 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; 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; and SX4DEV for developing and targeting C and Fortran software for an NEC SX-4 supercomputer (see Section 4). In addition, a tool CONVERT is provided for automatic and execution of a series of conversion tools to convert a given data set into the specified format.

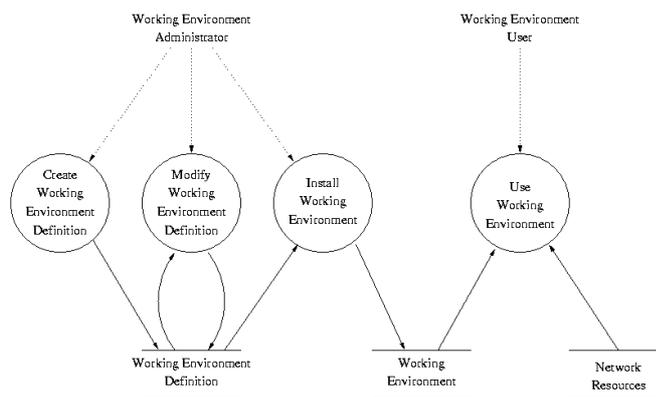


Fig. 5 The process of developing a working environment using SPINeware.

3.2 Using Spineware for the Realization of Working Environments

The process of developing a working environment is depicted in Fig. 5. The figure first shows the two different kinds of users involved. The *Working Environment Administrator* is responsible for creating the working environment, which can be used by the *Working Environment User*, who is referred to throughout this paper as "end user". The figure next shows, in the form of four circles, 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. The figure also shows the two information sets involved: the *Working*



Environment Definition and the *Working Environment*. The definition actually serves as "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 version compiled from the definition is the working environment as available to the end users.

The solid arrows from Fig. 5 indicate how a working environment is set up, starting from scratch. First, as represented by the "Create" circle, the definition of an initial working environment with minimum functionality is generated. The initial working environment provides a minimum virtual directory tree, containing 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 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 its users: a script, named after the working environment as determined during "Create", is present to start the User Shell of the working environment.

The key activity in development of a working environment is to "Modify" the definition of a working environment, in order to tailor it for end usage. First of all, it is necessary to "design" the working environment: to determine the contents and look-and-feel of the working environment for the end user. This requires expertise in the application field and the involvement of prospect end users. The design process is supported by SPINEware in that ideas can easily be translated into working-environment modifications and extensions, and hence can be demonstrated rapidly.

As indicated above, working-environment tailoring consists of modifying and extending its definition, which basically comprises changes in existing, and addition of new, text files in the definition. The modifications include:

- specification of host and site specific configuration parameters (e.g., path names of native programs, and internet names of specific computers);
- definition of the directory structure of the installed working environment (including the virtual directory tree);
- the definition of icons and their respective semantics, used for representing the directory entries;



- inclusion of tool packs (a tool pack is a semi-manufactured working environment in which tools and related information may be integrated, and which can be imported easily in working environments);
- the integration of programs and applications as tools.

The first four kinds of modification are accomplished easily by editing text files, which contain information in a straightforward and intuitive language. The latter - tool integration - deserves a little more attention.

As indicated in Section 3.1, tool integration is the prime tailoring activity. 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 (see Section 2), and which can be included 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. The SPINEware Tool Integrator facilitates tool integration by automatically generating a wrapper from information entered interactively by the user through a dedicated GUI. It is available from a working environment, and allows tools to be integrated in a working environment instantaneously (i.e., the result is immediately visible and usable), while also applying the necessary modifications to the working-environment definition for use in subsequent installations.



4 Tool Pack SX4DEV

The SX4DEV tool pack is a coherent set of tools for the development of simulation software on an NEC SX-4 parallel vector computer. In this paper the SX4DEV tool pack will serve as an example for the development of working environments. The tool pack combines the loosely coupled set of development utilities on the NEC SX-4 in an integrated set of tools.

A software developer on the NEC SX-4 requires correct and easy access to development tools such as compilers, source analyzers and debuggers. These tools, especially the compilers, usually come with a lot of options to accommodate any kind of user. For novice users the selection of useful options consumes a lot of time. The skills of experienced users would benefit these users greatly.

The SX4DEV tool pack presents the developer with an environment which

- contains expert knowledge on useful options,
- adjusts to the experienced user,
- allows development across LAN's and WAN's, and
- presents the tools in a uniform way, irrespective of platform and programming language.

The basic concept of SX4DEV is the Makefile. Given a source tree of directories containing source and/or include files, the user can automatically generate a Makefile for each directory in the tree. Targets are either libraries or executables. The Makefiles will contain the dependencies of the files in the directory on the files of the complete source tree. A top-level Makefile, generated by SX4DEV, links the different Makefiles in the (sub) directories and contains the compile options. Different languages can be mixed.

The top-level Makefile contains a complete description of the source through the Makefiles in the (sub) directories. The Makefile represents the source, and, through the compile options, the executable(s). This is exploited in the tool pack: whenever the source is handled, manipulation of the Makefile suffices. In order to compile the source, the user double-clicks the Makefile icon. In order to analyze or debug the source, the Makefile is dropped onto the analyzer or debugger icon. Hence, the representation of the source in the Makefile greatly facilitates the user interaction.



Whenever a tool is started, default options are presented to the user for, for example, efficient vectorization of a type of code analysis. These options are selected by expert developers on the NEC SX-4. For experienced users the tool pack allows for manipulation of the options.

Although the tools present themselves as a uniform set, behind the screens different tools are activated. For example, depending on the programming language of the main program, the analyzer starts the *fanp*, *fan90* or *ca* analyzer. In this way, even when the tools change, the user interface remains the same. If preprocessing of the source is necessary, by either the C or optimizing preprocessor, this is handled automatically. Needless to say, all network details are hidden from the user and the tool pack performs equally well on WAN's. Using SX4DEV, the computational kernel of a flow solver of TNO-TPD (the central institute for applied research in the Netherlands) was compiled, executed and analyzed within a quarter of an hour. Before that, TNO-TPD had never worked on an NEC supercomputer.

The reader should be aware that the above actually is a description of the end-users' wishes and the user interface for the solution. SPINeware provides a tool kit for the realization of the solution: a working environment for SX software developers.



5 Concluding Remarks

The development of SPINeware started early 1992 at the National Aerospace Laboratory NLR, as a product called ``SPINE". Since then, versions of SPINeware have been internally available at NLR, and several working environments have been realized. These include working environments for numerical flow simulation (ISNaS; Ref. [5]), for computer-aided control engineering (ISMuS; Ref. [6]), and for computer-aided software engineering (ISEnS). Efforts to hide the 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 also been used for the realization of several working environments by NLR and its partners in several national and international projects, such as NICE (Ref. [7]), FASTFLO (Ref. [8]), and MDO (Ref. [9]).

In 1995, NEC and NLR recognized SPINeware as a valuable tool for supporting the integration of their SX supercomputers in existing computer networks. Since early 1996, SPINeware has been developed jointly by NLR and NEC. Early 1998, NEC has 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 to 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 to 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. Other computers are accessed using the TCP/IP network utilities already found on most UNIX computers.

In this paper we presented SPINeware and its notion of working environment as solution for reducing the complexity emerging from the end usage of today's computer networks. The applicability of SPINeware has been proven by several existing working environments, 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 capability to also accommodate for security in wide-area networks even render it possible to use SPINeware for realization of so-called ``extended enterprises" (Ref. [10]). An important conclusion is that the total metacomputing



solution in combination with possibilities for end-usage tailoring is an ideal basis for using today's computer networks.

6 References

- [1] 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.

- [2] 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.

- [3] 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.

- [4] E.H. Baalbergen, H. van der Ven, SPINeware-A framework for user-oriented and tailorable metacomputers, submitted for publication in Special Issue on MetaComputing of the International Journal on Future Generation Computer Systems, Elsevier Publ., 1998.

- [5] 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.

- [6] 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.

- [7] 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.



- [8] 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.

- [9] 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.

- [10] M.E.S. Vogels, W. Loeve, Know-how management in distributed engineering environments, submitted to PROLAMAT'98 conference, Trento, Italy, 9-11 September 1998.