THE DESIGN AND IMPLEMENTATION OF A SOFTWARE TOOL FOR THE ELECTROTECHNICAL SIGNAL DISPLAY

Robert Porenta

KEYWORDS: Sun workstation, user interface, Sun View environment, circuit simulators, interactive program, algebraic expression syntax, software tools, software development, signal display, electrical signal, C programming language, UNIX.

ABSTRACT: This paper describes the design and implementation of a graphical tool for electrotechnical signal display. At the beginning of the paper some basic terms used throughout the text are briefly explained. In continuation the paper in turn describes the building blocks of the program.

NAČRT IN IZVEDBA PROGRAMSKEGA ORODJA ZA GRAFIČNI PRIKAZ ELEKTROTEHNIČNIH SIGNALOV

KLJUČNE BESEDE: delovna postaja Sun, uporabniški vmesniki, programsko okolje Sun View, simulator vezij, interaktiven program, sintaksa algebraičnega izraza, programska orodja, razvoj programske opreme, prikaz signalov, elktrični signali, UNIX, programski jezik C

POVZETEK: V članku sta opisana načrtovanje in izvedba programskega orodja za grafični prikaz elektrotehničnih signalov. Na začetku članka je na kratko razloženih nekaj osnovnih izrazov, ki so pozneje uporabljeni. Članek se nadaljuje z opisom posameznih enot programa.

1. Introduction

In the past few years the development of various electronic devices has become impossible without the support of software tools running on powerful workstations. By using tools such as circuit simulators, scheme editors, cell placers and others user can reduce development time and the number of errors significantly.

Reasons for development of a standalone tool are manifold. Many of similar programs offer only a limited graphical support on the level of simple character graphics, while more sophisticated ones use only a portion of graphical capabilities available on a workstation. Signal displaying programs are usually only a part of bigger software packages what makes such programs unsuitable for general use.

The most important program characteristics are the folowing:

- □ User friendly environment.
- □ Interactive operation.
- □ Ability to display data from different sources at the same time.
- □ New data can be obtained by means of algebraic expressions.
- □ The behaviour of program functions can be controlled by the user.

2. Programming tools

The program is implemented in the C programming language on the Sun workstation under UNIX operating system. For graphics independent functions only standard C library is used whereas user interface is supported by the SunView library.

2.1. The Sun workstation

Sun workstations are high-performance computers dedicated towards applications in the field of engineering,CAD,CAM,desktop publishing, software development and other tasks where high resolution graphics and high processing power is needed. The operating system is a heavily enhanced version of 4.2 BSD UNIX from the University of California at Berkeley.

Apart from standard UNIX utilities a user interface package based on multiple overlapping windows and associated libraries is added. Instead of typing commands on a keyboard the user can manipulate objects via a pointing device (mouse).

2.2 SunView library

SunView is an object oriented system. It consists of many building blocks which can be used to assemble a user interface for an application. Objects have different types according to their properties.

2.2.1 SunView objects

Windows are the most important SunView objects. Apart from windows other objects also include panel items, scrollbars, menus, cursors and icons.

Window objects include frames and subwindows. Frames are used to join nonoverlapping subwindows into one window. There are four types of subwindows:

Panel contains various panel items.

Text subwindow contains text.

Canvas is used for drawing area.

TTY subwindow serves as a terminal emulator.

Other visual objects are less general and thus more suited for their specific functions:

Panel item is part of a panel used for interaction between user and application. Some of predefined items are buttons, message items, choice items, text items and sliders.

Scrollbar is attached to a subwindow and serves to control which part of a subwindow is visible.

Menu is used for user to issue commands when one of menu's items is selected. By convention menus pop up on the right mouse button pressed and this is also true for menus in this program.

Cursor locates the mouse position on the screen.

Icon is a small image which represents various information.

3. Data organization

The program is designed to work with data obtained from simulation software. In order to handle data from various sources efficiently, a simple data model has been introduced.

3.1. Signal

Signal is an abstraction for a data which can be handled as one unit. Most often the signal represents information obtained in one node of a circuit for one simulation cycle (for example voltage in a node).

3.2. Signal group

Signal group is a data object consisting of data which are mutually related. For example signals which represent node voltages of a circuit can be associated to a signal group. Each signal group has a unique name for identification which is usually the name of the file from which the data were read. It is also possible to assign a list of parameters to each group. These parameters are usually the conditions under which the simulation has been executed (such as temperature for example).

3.3. Working signal group

Working signal group is a special signal group selected by the user. It is possible to load many signal groups from different files in order to display data from different analyses. Each of these signal groups can be selected to be the working group which basically serves as an identifier for functions operating on one signal group at the time.

4. User interface

The user interface is built from these basic SunView objects

□ Control panel
□ Signal window
□ Manual window
□ Expression window

Any of them may or may not be visible in particular situation. All these objects in turn consist of other more fundamental SunView objects to form a functional unit.

4.1. Control panel

Control panel (Figure 2) consists of many buttons, choices and text items which control the behaviour of the program as a whole. These buttons are used to select the particular action:

QUIT Regular exit from the the program.

REDRAW Redraw the contents of the signal window.

SAVE STATE Selecting this button causes program state to be written to a predefined file. The program state includes the settings of toggles, file pattern, window positions and other information.

LOAD STATE This button is used to set a new program state from a previously saved state.

READ FILE New data are read from file named in text item labeled "Read data from file"

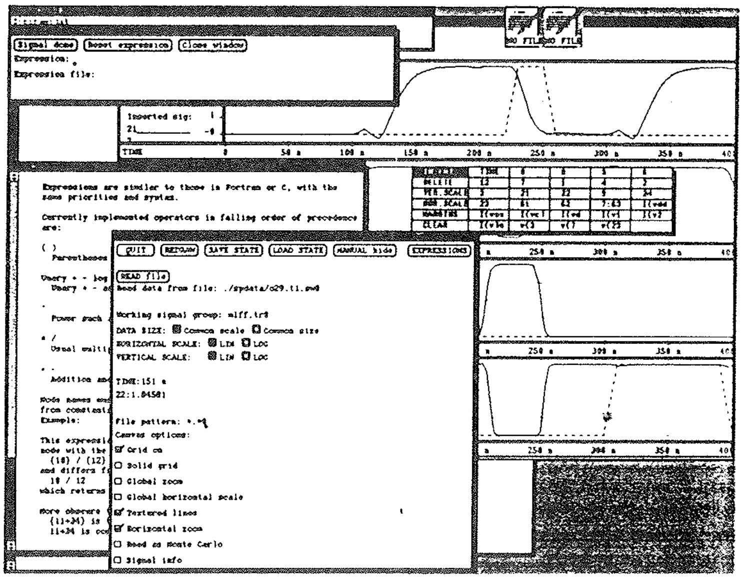


Figure 1: An example where the control panel, signal window, manual window and expression window are all open.

MANUAL popup Make manual text window visible when invisible and vice versa.

EXPRESSIONS Make expression window visible or invisible according to its current visibility.

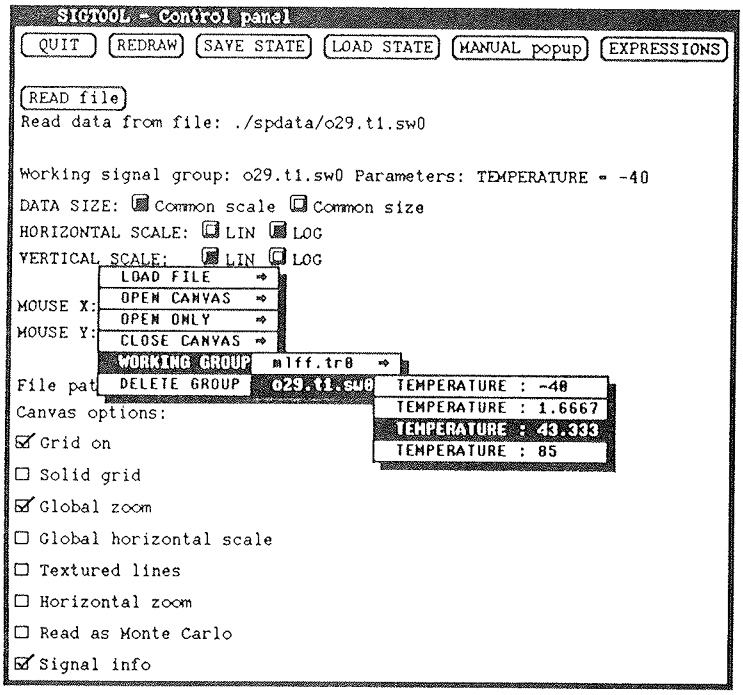
Choices serve to select one of many options. The chioces include:

DATA SIZE This choice decides whether data will be displayed with common scale or will be of the same size.

Common scale means that data in each signal canvas will be displayed according to common vertical scale and as a consequence their sizes will be proportional.

Common size denotes that each signal will be expanded from top to bottom of the signal canvas. For this reason the shapes of signals with different order of magnitude can be compared.

HORIZONTAL SCALE defines type of horizontal scale. User can select either linear or logarithmic scale.



```
Read data from file: ./spdata/o29.t1.sw0

Working signal group: o29.t1.sw0 Parameters: TEMPERATURE = -40
DATA SIZE: ▨ Common scale □ Common size
HORIZONTAL SCALE: □ LIN ▨ LOG
VERTICAL SCALE: ▨ LIN □ LOG

MOUSE X:
MOUSE Y:

File pat
Canvas options:
☑ Grid on
□ Solid grid
☑ Global zoom
□ Global horizontal scale
□ Textured lines
□ Horizontal zoom
□ Read as Monte Carlo
☑ Signal info
```

Figure 2: The control panel with its menu expanded.

VERTICAL SCALE sets vertical scale type which can also be linear or logarithmic.

Text items and messages are used to enter and display text strings:

Read data from file: This text item contains pathname of the file to be read by pressing the "READ file" button. Pathname can be entered from the keyboard or by selecting the path through a control panel menu which will be described later.

Working signal group: Because user can load many signal groups into memory, he has to decide which of these signal groups he wants to work with. He selects one of these groups via a control panel menu. The selected group is called "Working signal group". Many functions for data selection and manipulation operate on this signal group.

MOUSE X: MOUSE Y: These two items display the last cursor position in signal window. The position is interpreted according to vertical and horizontal scale.

File pattern: contains regular expression used in control panel menu for file selection. Only those files which match the regular expression are visible in the menu. Filename matching is similar to the file expansion mechanism in C–shell.

To control various parameters these toggles can be used:

Grid on toggle is used to set grid in signal canvases on.

Solid grid Default lines for grid are dotted while this flag sets their texture to solid.

Global zoom The zooming function is available in each signal canvas separately. With this toggle the zooming action in one signal canvas is effective in all canvases simultaneously.

Global horizontal scale similar to zooming function when this toggle is set, the change of horizontal scale in one signal canvas changes the scale in all canvases.

Textured lines By default all signals are drawn with solid lines. Different signals can therefore be distinguished only by their color. This is however not sufficient for black and white screens and other output devices such as laser printers. In that case the user can select this toggle for signals to be distinguished also by texture.

Horizontal zoom toggle controls the behaviour ob zooming function. With this toggle set the user defines only the left and right margin of the zooming rectangle while top and bottom margins are left unchanged.

Read as Monte Carlo This option is specially designed for HSPICE data files where more analyses can be written into one file. With toggle set the file data are read into one signal group and thus treated as a single unit.

Signal info In signal window each signal canvas includes a list of signals currently displayed. The toggle causes not only signal name but also other information to be displayed.

4.2. Menu in control panel

This menu (Figure 2) pops up when user selects right mouse button in the control panel window space. The control panel menu contains various functions concerning filename selection and signal window control.

LOAD FILE This item is at the root of the hierarchical menu which resembles the filesystem structure. The user can search for desired file either from filesystem root "/" or from current working directory ".". Because each directory can contain more files than a menu is able to display and to make it more readable, filename expansion mechanism is provided. Each menu item for one directory can represent either a regular file name or a directory name. Filenames are only visible when their name matches the file pattern from the "File pattern:" item present in the control panel. By setting this item the user can control the visibility of different file names. Directory names are always visible regardless of their names. Each directory is presented by a pullright menu item indicated by a little arrow to its right. If the user acctually pulls right crossing the arrow, the contents of affected directory are displayed in a submenu. The selected pathname is copied into the "Read data from file:" text item in the control panel.

OPEN CANVAS is a pullright item where the pullright menu contains the indexes of all available signal canvases. Each signal canvas has a unique index associated to it which serves for its identification. On selection the canvas with the selected index is made visible.

OPEN ONLY item makes the selected signal canvas the only one which is open. It opens the selected signal canvas and closes all other open canvases.

CLOSE CANVAS item closes the selected signal canvas.

WORKING GROUP is used to select the working signal group which is displayed in the control panel in "Working signal group:" item. The working signal group can be chosen among signal groups currently loaded into the memory.

DELETE GROUP deletes the selected signal group from the memory.

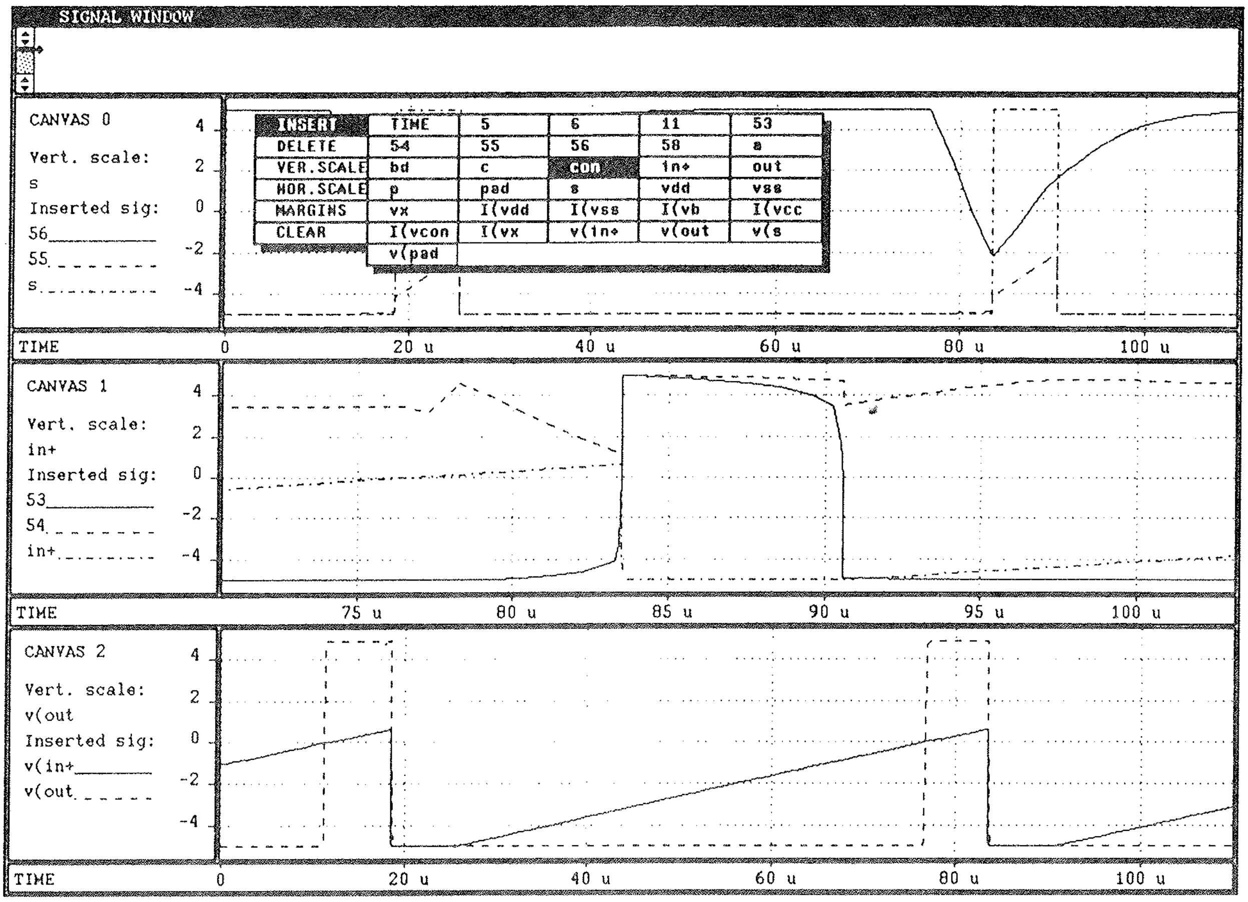


Figure 3: The signal window with three open signal canvases.

4.3. Signal window

Signal window (Figure 3) is used to graphically display data of different types. It consists of several subwindows all of which need not be visible at the same time.

The text subwindow used for comments made by the user is positioned at the top of the signal window and is always visible.

After the text subwindow one or more signal canvases with associated vertical and horizontal scales can be positioned from top to bottom. Each signal canvas has an index associated to it starting from 0. Signal canvas can be either open or closed depending on user settings from the menu in the control panel. Signal window has fixed vertical and horizontal dimensions. As a consequence of that the height of currently open signal canvases is set so that heights of open canvases are equal and that they occupy all available space. Because vertical dimensions of canvases change whenever the user opens or closes one, the contents of each canvas are scaled so that they fit the new dimensions.

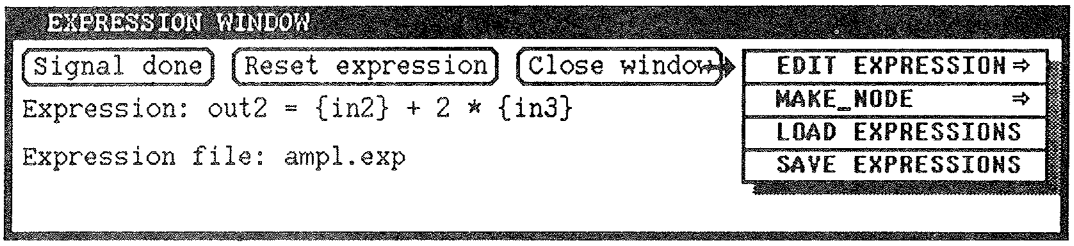
In each of the signal canvases a zooming function is available. In order to zoom in, two corners of a rectangle have to be defined. The first click of the left mouse button sets the first corner while the second click of the same button expands the rectangle to the whole signal canvas thus completing the zooming operation. In order to cancel the zooming when the left corner has already been set, the user has to press the middle mouse button. Zooming out can be done either by the menu in signal window or by selecting one of "DATA SIZE" items in the control panel.

4.4. Menu in signal window

Like the menu in the control window this menu pops up on the right mouse button pressed. It contains functions for signal selection and signal canvas control. Menu is active in signal canvases. For each canvas the menu items are dynamically set according to that canvas state.

The menu items are:

INSERT menu item has a pullright menu containing signals from working signal group. By selecting a signal it is displayed in signal canvas.



Expression: out2 = {in2} + 2 * {in3}

Expression file: ampl.exp

Figure 4: The expression window and associated menu.

DELETE deletes previously inserted signal from the signal canvas.

VER.SCALE The selected signal is used as a reference for the vertical scale.

HOR.SCALE The selection sets signal used as a reference for the horizontal scale. On toggle "Global horizontal scale" in the control panel set, the effect is as if the horizontal scale would have been set in all signal windows simultaneously to the selected signal.

MARGINS On this selection the margin window pops up enabling the user to set new top, bottom, left and right margins for the signal canvas. These margins are used to zoom in or out so that they overlap with the signal canvas margins. When the toggle "Global zoom" in the control panel is set, the margins are set in all signal canvases simultaneously.

CLEAR Clears the desired signal canvas by deleting all signals.

4.5. Expression window

The expression window (Figure 4) is used to construct new signals from those already belonging to some signal group. Newly constructed signals are added to the working signal group. The expression window contains the folowing items:

Signal done By pressing this button the user adds the signal made by the "Expression:" item into the working signal group. When any errors are detected the user is requested to make corrections.

Reset expression clears the text in the "Expression:" item.

Close window Button closes the expression window.

Expression: is the text item where the user enters the algebraic expression for the constructed signal. Syntax checking is postponed until the signal is constructed.

Expression file: text item accepts the pathname for a file where the expressions are to be saved or read from.

4.6. Menu in expression window

The menu (Figure 4) is poped up by the right mouse button. It contains items for expression control:

EDIT EXPRESSION Every signal expression that was succesfully done is stored in a list. This item's pull-right menu contains all these expressions in inserting order. The selected expression is copied into the "Expression:" item where it can be edited.

MAKE NODE item is used to explicitly make a signal from the selected expression. This is useful when more than one signal has to be made from one expression.

LOAD EXPRESSIONS Loads expressions from file in "Expression file:" and appends them to those already made.

SAVE EXPRESSIONS Saves expressions into the filename found in the "Expression file:" item.

4.7. Algebraic expressions

Each signal created in the expression window is defined by an algebraic expression. The expression consists of operands, operators and parentheses for grouping with the syntax reminiscent to Fortran or C. The operators are divided into:

unary operators are unary + and -. They together with functions have the highest precedence.

operators in order of decreasing precedence are

- □ exponentiation(∧).
- □ multiplication(*) and division(/).
- □ addition(+) and subtraction(-).

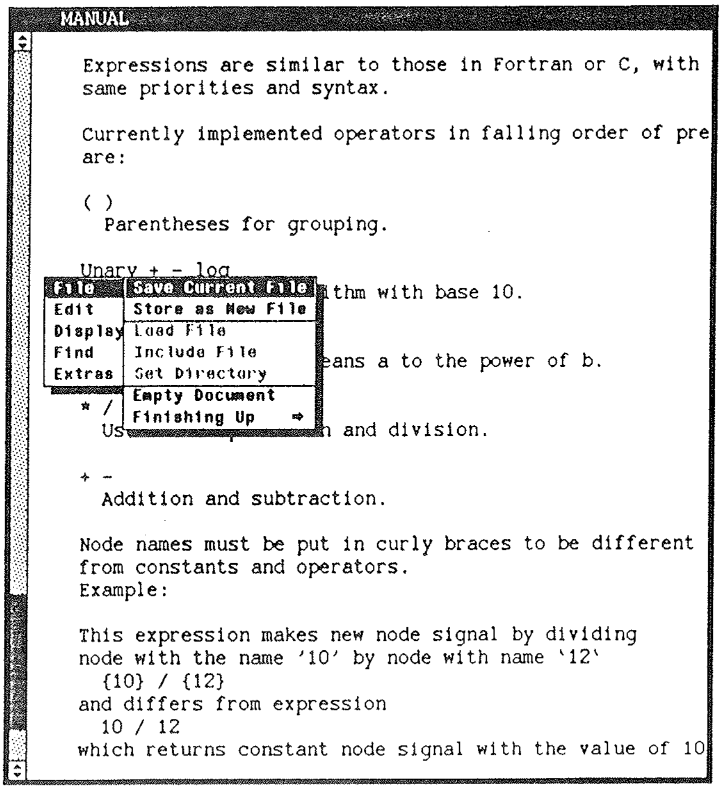
functions The precedence of functions is the same as for the unary operators. This is the complete list of available functions:

trigonometric functions – sin, cos, tan , asin, acos, atan

hyperbolic functions – sinh, cosh, tanh

square root – sqrt

exponential function – exp



```
Expressions are similar to those in Fortran or C, with
same priorities and syntax.

Currently implemented operators in falling order of pre
are:

( )
  Parentheses for grouping.

Unary + - log
                                   thm with base 10.

                                   eans a to the power of b.

* /
  Us                                 and division.

+ -
  Addition and subtraction.

Node names must be put in curly braces to be different
from constants and operators.
Example:

This expression makes new node signal by dividing
node with the name '10' by node with name '12'
  {10} / {12}
and differs from expression
  10 / 12
which returns constant node signal with the value of 10
```

Figure 5: The manual window with its menu.

logarithm – log, log10,db(20 * log10(X))

minimum of two expressions X and Y – min(X,Y)

maximum of two expressions X and Y – max(X,Y)

absolute value – abs

Operands can be of the folowing types:

constant Constants can have any integer or real value.

signal name Signal name is treated similar to variables in Fortran or C since each signal really represents a set of values. Signal names must be used in curly braces to be distinguised from constants and operators.

4.8. Online manual

The online manual (Figure 5) is available in a text window which can be opened or closed at any time. The window contains a short reference guide for the entire program.

5. Conclusion

The design of a program for graphic signal display poses a number of problems. User friendly interface with simple and yet flexible display is by experience most difficult to achieve. Using the SunView system has proved to be very effective. By combining a number of visual objects very compact user interface can be built with much less effort compared to more conventional graphic libraries. The program has been in use for several months very successfully. Most of the credit is due to its interactive operation which enables the user to concentrate attention to the data displayed rather than to the program operation. In practical use some new ideas have emerged which may be included in the subsequent releases of the program.

References

1.) Brian W. Kernighan, Dennis M. Ritchie: The C programming language. Prentice Hall, 1983.

2.) Bourne S. R.: The UNIX system. Addison Wesley Publishing Co., 1982.

3.) Meta-Software, Inc.: HSPICE User''s Manual. Meta-Software, Inc., 1990.

4.) Sun Microsystems, Inc.: SunView Programmer''s Guide. Sun Microsystems, Inc., 1988.

Robert Porenta, dipl. ing.
Fakulteta za elektrotehniko in računalništvo
Tržaška 25, 61000 Ljubljana

Prispelo: 23.9.91 *Sprejeto: 15.11.91*