Computer Aided Software Reliability Estimation

(CASRE)

User's Guide

Version 3.0

March 23, 2000

Prepared by:
Allen P. Nikora
Autonomy and Control Section
Jet Propulsion Laboratory
4800 Oak Grove Drive
Mail Stop 125-209
Pasadena, CA 91109-8099

tel: (818)393-1104
fax: (818)393-4085

e-mail:
Allen.P.Nikora@jpl.nasa.gov

Table of Contents

APPENDICES

Table Of Figures

Tables

1. Introduction

CASRE (Computer Aided Software Reliability Estimation) is a software reliability measurement tool that runs in the Microsoft Windows™ environment. Although there are several software reliability measurement tools currently available, CASRE differs from these in the following important respects:

1. Plots of the failure data used as inputs to the models are displayed simultaneously with the text. Failure data text is shown in one window, while plots are shown in another. Changes made to the text of the failure data are automatically reflected in the displayed plots.

2. The results given by the software reliability models included in CASRE are displayed graphically, in terms of failure counts per test interval, times between successive failures, and the cumulative number of errors discovered. Plots of various model evaluation criteria, such as model bias and bias trend, can also be made.

3. Users can define so-called combination models, which are linear combinations of modeling results. The developers of CASRE have found that combining the results of several models in a linear fashion tends to yield more accurate results overall than relying on any single component in the combination. CASRE allows users to define several types of combination models, store them as part of the tool's configuration, and execute them exactly in the same way as any component model.

The modeling and model applicability analysis capabilities of CASRE are provided by the public-domain software reliability package SMERFS (Statistical Modeling and Estimation of Reliability Functions for Software). SMERFS was developed by Dr. William H. Farr at the Naval Surface Warfare Center, Code B-35, Dahlgren, VA, 22448. SMERFS is available for approximately $70.00 from Automated Sciences Group, Inc; 16349 Dahlgren Road; P.O.Box 1750; Dahlgren, VA; 22448-1750. In implementing CASRE, the original SMERFS user interface has been discarded, and the SMERFS modeling libraries have been linked into the user interface developed for CASRE. The combination modeling capabilities, however, are new for CASRE.

1.1. Applicability

CASRE can be applied starting after unit test and continuing through system test, acceptance test, and operations. **You should only apply CASRE to modules for which you expect to see at least 40 or 50 failures. If you expect to see fewer failures, you may reduce the accuracy of your estimates. Experience shows that at the start of software test, modules having more than about 2000 source lines of executable code will tend to have enough faults to produce at least 40 to 50 failures.**

1.2. Limitations

The limitations associated with CASRE have to do with the amount of available RAM and the size of memory segments. The limitations are:

1. A file of failure data must be no larger than 64K bytes, the maximum size of a RAM segment. This is because CASRE holds the entire failure data file in RAM. For time between failures data, this translates to a maximum of about 3000 failures per file. For failure count data, this means that a file can record information for up to about 2000 test intervals.

2. The number of models you can run at any one time depends upon the amount of RAM in your system. For any one model, the results can occupy up to 213,000 bytes of RAM. Since CASRE does all of its operations in memory, the more RAM you have in your system, the more models you can run simultaneously.

2. Installing CASRE

2.1. Required Operating Environment

CASRE has been designed to run in a Microsoft Windows™ 3.1 or higher environment. Computers on which CASRE will run must have the following characteristics:

1. Operating Environment - Microsoft Windows 3.1, Windows 3.11, Windows95, or WindowsNT.

2. CPU - 80386 with an 80387 math coprocessor, 80486 DX, or Pentium. Since you will be doing a lot of floating-point calculations when running models, a 66MHz or faster 80486 DX or Pentium based system is **highly recommended.**

   NOTE: This version of CASRE runs on computers having EISA motherboards, but DOES NOT NECESSARILY RUN on machines using the "local bus" architecture. If you're planning on acquiring a computer having the "local bus" architecture, make sure that CASRE runs on it before you purchase the machine.

3. Disk space - You should have at least 2 MB of free space on your hard drive to install CASRE. In addition, the data files used by CASRE can be up to 64KB long.

4. Pointing device - two-button Windows-compatible mouse. CASRE will not run without a mouse or equivalent pointing device (e.g. Windows-compatible trackball, touch pad, or digitizing tablet). The term "mouse" is used in this user's guide to indicate any appropriate pointing device.

5. Memory - considering the volume of modeling results that may be generated in a single CASRE session, at least 8MB of RAM is recommended.

6. Monitor - a 17" or larger VGA or better quality monitor supported by Windows is expected. Although CASRE is implemented to allow you to distinguish one variable from another on a black and white monitor, the best results will be obtained using a VGA or higher quality color monitor. Since there may be several other windows open at any time in addition to on-screen menus and control panels, a 19" or larger monitor is highly recommended.

7. Printer - a printer supported by Windows is assumed. It is assumed that the available printer will be capable of printing high-quality graphics as well as text. Since CASRE allows users to draw high-resolution plots on a printer as well as on-screen, a 300dpi or better resolution laser printer is highly recommended.

2.2. Installation Procedure

You should follow these instructions if you're installing CASRE for the first time, or if you're upgrading to this version from an earlier version. If you've defined any combination models with the earlier version, or if you've linked external applications to the earlier version, you'll retain those combination definitions and links to external applications, with the following exceptions:

- **Any combinations that depend on one of the Brooks and Motley models will not be retained in version 3.0. This includes combinations for which one of the components is itself a combination depending on one of the Brooks and Motley models.**
- **If you're changed the name or location of an external application, or if you're removed it from your disk between the last time you used an earlier version of CASRE and the time you're installing version 3.0, the link to that application will not be retained.**

Finally, if you've modified any of the sample data files distributed with an earlier version, and left them in the "c:\casre\data" subdirectory, you'll lose those modifications if you follow the steps below.

The CASRE executable and sample data files are packaged in a self-extracting WinZip file, which is shipped on one 3.5-inch high-density (1.44 Mb) diskette. This self-extracting file contains 28 files. Six of these files are directly associated with the executable load. One of these is the executable file itself, CASRE.EXE. There is also the "stub" file, "CASRSTUB.EXE. This is the executable file, copied into the Windows directory during installation, which notifies users trying to invoke CASRE from the DOS prompt that CASRE must be invoked from within Windows. Besides the executable file, there are three help files: CASRE.HLP, PLOT.HLP, and RSLTTABL.HLP. Each of these help files corresponds to one of the three CASRE windows. The sixth file, CASREV3.INI, defines the tool configuration as far as external applications and user-defined model combinations are concerned. The "README.TXT" file provides a summary description of the new features for CASRE, version 3.0. The installation program which you will use to install CASRE on your hard disk, "INSTALL.EXE", is also included in the self-extracting WinZip file. Finally, the self-extracting file includes a subdirectory, "\data", which contains sample data files that you can use to explore CASRE's capabilities.

CASRE must be installed on the type of platform described in section 2.1. If Microsoft Windows has not already been installed on the hardware, install it prior to installing CASRE. After you've installed Windows, start Windows and then start the install program:

1. In Windows 3.1 or Windows 3.11, bring up the File manager and find the icon for "casre30.exe". This icon will be found in the root directory of the distribution diskette in drive "a:\". Double-click the icon to extract the CASRE files. You will be prompted for the name of a subdirectory into which the files will be extracted –

enter “c:\temp”, “c:\windows\temp”, or the name of another subdirectory that holds temporary working files.

2. Using the File Manager (Windows 3.1), or Windows Explorer (Windows95), go to the temporary subdirectory into which the CASRE files have been extracted. This is the subdirectory you identified in step 1. Find the icon for the installation program, “install.exe”. Double-click the icon to start the install program.

Once the installation program has been started, select the “Start” item in the “Install” menu. CASRE will be installed on your machine. INSTALL.EXE, assumes that your hard disk is drive C:. It will make the “c:\casrev3” and “c:\casrev3\data” subdirectories on your hard disk. CASRE.EXE, the three *.HLP files, and the installation log, “install.log”, will be copied to c:\casrev3. The installation program also assumes that the directory in which Windows is installed is c:\windows. The "CASREV3.INI" file will be copied to that subdirectory, as will the program CASRSTUB.EXE. The installation program finishes by making a CASREV3 program group if run in the Windows 3.1 and Windows 3.11 environments. In Windows95, you can create your own desktop folder for CASRE, or add CASRE to your Start menu programs. See Windows95 help for further details. After installation has been completed, exit the installation program by selecting “Exit” on its “Install” menu.

The installation log, INSTALL.LOG, is a text file that gives you details of the CASRE installation. If you’re upgrading from previous versions of CASRE, external applications that you’ve installed will be retained in the CASREV3.INI file, as will model combination definitions. **However, model combinations that depend on the Brooks and Motley models will not be retained, as CASRE version 3.0 no longer includes these models. The developers are of the opinion that data collection for these models is sufficiently troublesome that they will be used very infrequently. Eliminating these models simplifies the structure of the input files for failure counts data and makes it much more understandable**. A typical install.log file is shown below. Annotations are prefaced with “//”.

```
Installing CASRE.EXE in \CASREV3.
        Copied application file CASRE.EXE to \CASREV3   // States that the CASRE application file
                                                        // was successfully copied to the target directory.

Installing README.TXT file in \CASREV3.
        Copied README file README.TXT to \CASREV3.          // States that the README file was
                                                            // successfully copied to the target directory.

//      The next section of the install.log file names the data files that have been
//      successfully copied to the c:\casre\data subdirectory on the hard disk.

Installing sample data files in \CASREV3\DATA.
        Copied data file fc_temp.dat to \CASREV3\DATA.
        Copied data file fc_test.dat to \CASREV3\DATA.
        Copied data file fc_test1.dat to \CASREV3\DATA.
        Copied data file fc_test2.dat to \CASREV3\DATA.
        Copied data file fc_test3.dat to \CASREV3\DATA.
        Copied data file fc_test4.dat to \CASREV3\DATA.
```

Copied data file fc_test5.dat to \CASREV3\DATA.
Copied data file s1.dat to \CASREV3\DATA.
Copied data file s2.dat to \CASREV3\DATA.
Copied data file s3.dat to \CASREV3\DATA.
Copied data file tbetst2a.dat to \CASREV3\DATA.
Copied data file tbe_nhpp.dat to \CASREV3\DATA.
Copied data file tbe_test.dat to \CASREV3\DATA.
Copied data file tbe_tst1.dat to \CASREV3\DATA.
Copied data file tbe_tst2.dat to \CASREV3\DATA.
Copied data file tbe_tst3.dat to \CASREV3\DATA.
Copied data file tbe_tst4.dat to \CASREV3\DATA.
Copied data file tbe_tst5.dat to \CASREV3\DATA.
Copied data file test1.dat to \CASREV3\DATA.

Installing CASREV3.INI file in Windows directory.
Copied INI file CASREV3.INI to C:\WINDOWS.

Creating CASREV3.GRP group file in Windows directory.
Copied group file CASREV3.GRP to C:\WINDOWS.

Installing CASRE stub file CASRSTUB.EXE in Windows directory.
Copied stub file CASRSTUB.EXE to C:\WINDOWS.

Installing help files in CASRE directory.
Copied help file CASRE.HLP to C:\CASREV3.
Copied help file PLOT.HLP to C:\ CASREV3.
Copied help file RSLTTABL.HLP to C:\ CASREV3.

Integrating CASRE program group into Program Manager.

// In the following section, model combinations defined in previous versions of CASRE
// are retained for use in version 3.0.

Installing the following user-defined combinations that were
defined in previous versions.
Changed component Littlewood-Verrall to Quadratic LV in user-defined combination SLC_17.

// In the combination SLC17, the Littlewood-Verrall model was specified as a component.
// Users of earlier versions of CASRE could select the Littlewood-Verrall model as a
// combination model component; version 3.0 allows users to select two versions of this
// model as components in a combination. The installation program substitutes the
// quadratic form of this model in combination definitions retained from earlier versions.

Deleted the user-defined combination SLC_18.
One of its components depends on the Brooks and Motley models.
Deleted the user-defined combination SLC_19.
One of its components depends on the Brooks and Motley models.

// Version 3.0 of CASRE no longer implements the Brooks and Motley models. Model
// combinations depending on this component model are not retained for use in version 3.0.

Changed component Generalized Poisson to Schick-Wolverton in user-defined combination SLC_20.

//   Unlike earlier versions, version 3.0 explicitly identifies the varieties of the Generalized Poisson
//   model available to users. For combination model definitions from earlier versions that include
//   the Generalized Poisson models, the installation program substitutes the Schick-Wolverton variety
//   of this model in those definitions.

Changed component Schneidewind to Schneidewind: all in user-defined combination SLC_21.

//   For those combination model definitions that included the Schneidewind model in
//   earlier versions, the installation program substitutes a specific form of this model
//   in those definitions.

Installing the following links to external applications
that were defined in previous versions.
Transferring external application Write from previous version.

//   The link to the external application "Write" (WordPad in Windows95) that was made in
//   an earlier version is retained for version 3.0.

External application AARGH was not found.

//   A link to an external application "AARGH", made with a previous version, is deleted because
//   the external application cannot be found on disk (it was deleted).

Installation of CASRE version 3.0 was successfully completed.
One or more external applications specified in previous versions were not found.
Details are given in the log information above.

//   Installation was successfully completed, and CASRE version 3.0 can now be run.
//   Some of the links to external applications made in earlier versions of CASRE were
//   not retained, because the applications could not be found on disk.

2.3. Support

If problems are encountered in running CASRE, notify the developers at the address or telephone number shown on the front of this manual. If possible, please have the following information available:

1. A description of the hardware and software configuration of your computer, including memory managers and any TSR programs that were in memory when CASRE was running.

2. The sequence of steps executed up until the problem occurred.

3. A description of the problem.

4. The CASRE version number. You'll find this number in the dialog box that appears when you select the "About..." item in the main window's "Help" menu.

We will then attempt to duplicate the problem, determine its cause, and develop a solution. If the problem cannot be duplicated, we may ask you if the input data that you were using can be sent to us to help identify the problem. If the solution involves a repair to CASRE, the repair will be prioritized for inclusion in the next scheduled release of CASRE. Requests for changes and/or additional features will be handled in the same fashion.

3. CASRE Menu Trees

This section of the user's guide describes the CASRE menu trees. CASRE has three windows, which are:

1. The MAIN WINDOW, which is the window that appears when CASRE is invoked, containing a text representation of the failure data used as input to the models.
2. The GRAPHIC DISPLAY WINDOW, in which graphical representations of the failure data as well as modeling results are displayed.
3. The MODEL RESULTS TABLE, in which detailed modeling results are shown in tabular form. The items shown in this table include model parameters, model estimates, model applicability statistics, and the residuals (actual data minus model estimates).

Each of these windows has an associated menu tree. The next three sections show the structure of each tree, and briefly describe each menu item. An underlined letter indicates that the letter is an accelerator key for that menu item. For instance, the "Open" menu item in the main window's "File" menu could be invoked by using the mouse, or by using an "Alt-F" keystroke, followed by an "o". If the word "GRAYED" appears after a menu item, that item initially appears in grayed-out lettering after CASRE is started, and is unavailable until a specific action has been performed. For instance, the "Select models" menu item is initially grayed-out. You cannot select models to be run until a file has been opened. After a file has been opened, the "Select models" menu item is enabled, and is shown in normal printing. Finally, you can end the current CASRE session either by selecting the "Exit" menu item under the "File" menu, or by pressing the F3 function key. There are a few other CASRE functions that can be performed by using the function keys.

3.1. CASRE Menu Structure

The overall CASRE menu structure is given below. The top-level menu for each window is shown.

CASRE

Main Window	Graphic Display Window	Model Results Table
File	Plot	File
Edit	Results	Results
Filters	Display	Help
Trend	Settings	
Model	Copy	
Setup	Help	
Plot		
Help		

The menus for each window are further described below.

3.2. Main Window

The menu tree for the main window is given below. The text of the failure data is displayed in this window.

- "File"
  - "Open..."
  - "Save", GRAYED
  - "Save as..." , GRAYED
  - "Setup printer..."
  - "Print", GRAYED
  - "Exit" can also use F3 function key
- "&Edit"
  - "Change data type...", GRAYED
  - "External application"
    - "Notepad"
  - "Escape to DOS"
- "Filters"
  - "Shaping and Scaling", GRAYED
    - "Scaling and offset...", GRAYED
    - "Power...", GRAYED
    - "Logarithmic...", GRAYED
    - "Exponentiation...", GRAYED
- "Change time units...", GRAYED
  - "Smoothing", GRAYED
    - "Hann window", GRAYED
  - "Subset data", GRAYED
    - "Select severity...", GRAYED
- "Round", GRAYED
  - "Remove last filter", GRAYED
  - "Remove all filters", GRAYED
- “Trend”
  - “Running average”
  - “Laplace Test”
  - “Undo trend test”, GRAYED
- "Model"
  - "Select models...", GRAYED
  - "Define combinations"
    - "Static weights..."
    - "Result based weights..."
    - "Evaluation based weights..."
  - "Edit/remove models...", GRAYED
  - "Parameter estimation..."
  - "Select data range...", GRAYED

"Predictions...", GRAYED
"Setup"
"Add application..."
"Remove application...", GRAYED
“Remember Most Recent Files...”
“GOF Significance…”, GRAYED
“Laplace Test Sig…”
"Plot"
"Create plot window", GRAYED
"Help"
"Help index"
"Keys help"
"Help for help"
"About..."

3.2.1. FILE Menu

The items in the "File" menu are used to open and save failure data files, select and set up printers, print the contents of the main CASRE window, and end the current CASRE session.

Open... - Select a failure data file to be opened. The data will then be displayed in tabular form in the main window. The data in the opened file can then be used as input to one or more reliability models.

Save - Replace the original file contents with the modified data.

Save as... - Save the contents of the main CASRE window as a new file on disk, or rename the existing disk file.

Setup printer... - Select a printer on which the contents of the CASRE main window will be printed, and specify its resolution and the orientation (landscape or portrait) in which text will be printed.

Print - Print the contents of the CASRE main window to the printer selected by the "Setup printer..." menu item.

Exit - End the current CASRE session.

3.2.2. EDIT Menu

The items in the "Edit" menu allow you to run an external application (e.g. an editor or word processor to edit failure data files), to open a window in which to execute DOS commands, or to change the way in which the failure data is represented.

Change data type... - Changes failure data from time between failures (TBF) to failure counts (FC) or vice versa. Some of the models built into CASRE accept only TBF data, while the others accept only FC data. This menu item is included to allow use of both types of models on the same set of failure data.

External application - Brings up a submenu which gives the names of up to 65 Windows or DOS applications that you can run from inside CASRE. These applications provide the text editing capability for this version of CASRE. CASRE allows you to add external applications to or remove them from this submenu.

Notepad - This is a permanent item in the "External editor" submenu described above. This editor, which is included with Windows 3.1, cannot be added to or removed from the "External editor" submenu.

Escape to DOS - Brings up a window in which DOS commands can be executed. When you're finished running DOS commands, remove the DOS window from the screen by entering "exit" at the DOS prompt within the window.

3.2.3. FILTERS Menu

The "Filters" menu items allow you to change the shape of the curve that can be drawn through the failure data by applying one or more transformations to that data. It also allows you to remove failure data noise by applying a Hann window, and allows you to define a subset of the data based on the severity classification of the observed errors. Finally, you can remove the effects of either the most recent filter that was applied, or the effects of all filters that have been applied.

3.2.3.1. Shaping and Scaling

The shaping and scaling transformations included in CASRE allow the shape of a curve drawn through the failure data to be changed. The filters are implemented such that the output of any filter remains in the first quadrant of the plot shown in the graphic display window. This is to prevent filters from producing physically meaningless results (e.g., negative times between failures, failure counts less than 0). These filters can help you spot trends in the data and more easily identify appropriate models.

Scaling and offset... - Multiply each observation (time elapsed since the last failure, or failure count for a test interval) by a scaling factor and then add an offset.

Power... - Multiply each observation by a scale factor, then raise the result to a user-specified power.

Logarithmic... - Multiply each observation by a scale factor, add an offset, then take the natural log of the result.

Exponentiation... - Multiply each observation by a scale factor, add an offset, then raise the base of natural logarithms to the result.

3.2.3.2. Other Filter Operations

The remaining filter operations allow you to remove failure data noise, change the time units of the failure data, form subsets of the failure data, or remove the effects of filters that have been applied to the data.

Change time units... - Failure data has time units associated with it. Failure count data measures the length of each test interval in terms of seconds, minutes, hours, days, weeks, months, or years. Time between failures data measures the time between successive failures in terms of the same units. This filter allows you to change the time units associated with the currently open set of failure data. For instance, in a failure count data set, you can express the test interval lengths in terms of minutes instead of hours. If the length of the test intervals is given as 40 hours in the data file, the test interval lengths will be shown as 2400 minutes after this filter has been applied.

Smoothing - Applies a Hann window (a triangular moving average) to the failure data shown in the main CASRE window. This filter is designed to reduce noise in the failure data.

Subset data - Allows a subset of the data to be created by selecting observations having a severity in a user-specified range between 1 and 9. This allows you to separately model failure rates for each severity category in a set of failure data, if desired.

Round - Rounds the failure data to the nearest whole number. For failure counts data, the counts for each severity class are rounded separately. They are then summed to yield the total number of failures in each test interval.

Remove last filter - Removes the effects of the last filter applied to the failure data shown in the CASRE main window. For example, if a Hann window was applied to smooth the data, and a logarithmic filter was then applied, removing the last filter would remove the effects of the logarithmic filter, but not the Hann window.

Remove all filters - Removes the effects of all filters that have been applied.

3.2.4. TREND Menu

The “Trend” menu lets you run two different trend tests against the failure history data to see if it exhibits reliability growth. If the data exhibits reliability growth, software reliability models can be applied to the data. If the data does not exhibit reliability growth according to these tests, then software reliability models should not be applied.

Running average – Computes the running average of the time between successive failures for time between failures data, or the running average of number of failures per interval for failure count data. For failure count data, this test is available only if the test intervals are of equal length. For time between failures, if the running average increases with failure number, this indicates reliability growth. For failure count data, if the running average decreases with time (fewer failures per test interval), reliability growth is indicated. The results of the test are shown in the main window and the graphic display window.

Laplace test – Computes the Laplace test for either data type. As above, for failure count data, the test intervals must be of equal length. The null hypothesis for this test is that occurrences of failures can be described as a homogeneous Poisson process. If the test statistic decreases with increasing failure number (test interval number), then the null hypothesis can be rejected in favor of reliability growth at an appropriate significance level. If the test statistic increases with failure number (test interval number), then the null hypothesis can be rejected in favor of decreasing reliability. The results of the test are shown in the main window and the graphic display window.

Undo trend test – If a trend test has been performed, this menu item undoes the trend test, restoring the main window and the graphic display window to the state in which they were prior to performing the trend test.

3.2.5. MODEL Menu

The "Model" menu lets you select one or more software reliability models to run, or define combination models and store them as part of CASRE's configuration. You can also specify modeling ranges, how far into the future predictions should be made, and parameter estimation methods.

Select models... - Allows selection of one or more software reliability models to be run, using the failure data shown in the CASRE main window.

Define combinations - Allows you to define combinations of model results and include the definitions as part of CASRE's configuration. All such "combination models" can be selected using the "Select models" menu item above. Descriptions of ways to form combination models are given in section 4.13.3.

Edit/remove models... - You can modify or remove combination model definitions that have been added to CASRE's configuration. This menu item does not allow you to remove any of the models that are a permanent part of CASRE's configuration.

Parameter estimation... - Brings up a dialog box to select between the maximum-likelihood and least-squares parameter estimation methods. The default is maximum likelihood.

Select data range... - Selects the range of observations to use as input to the models. The default data range is the last 1000 observations in the data file.

Predictions... - Lets you specify how far into the future predictions should be made, using model results. Depending on the data type, this represents either the next "n" intervals for which the number of failures should be predicted, or the next "n" failures for which the times between failures should be predicted. If you're working with failure count data, you'll also need to specify how long you expect future test intervals to be in order for the models to be able to make predictions. The default future test interval length is the same as the length of the last interval in the data set.

3.2.5.1. Define Combinations

The "Define combinations" sub-menu of the "Model" menu allows combinations of model results to be defined and stored as part of CASRE's configuration. Once created, these definitions can be executed in exactly the same way as the individual component models can be selected to be run.

Static weights... - In creating statically weighted combination models, each component of the combination is assigned a specific, constant weight. If users assign a value of $\mathbf{w_i}$ to a component's weight, and that component's result is represented as $\mathbf{r_i}$, the combination result is given as:

$$\sum_{1}^{n}(w_i r_i / \sum_{1}^{n} w_i)$$

Result-based weights... - Weights are dynamically assigned to the components of a result-based combination. The weights for each component are determined by comparing the component model results to one another. The weights are re-evaluated every "n" observations, where "n" is a number that you specify. For a combination model with three components, for instance, the weights 1, 2, and 3 could be chosen for the model predicting the lowest failure rate, the model predicting the next lowest failure rate, and the model predicting the highest failure rate, respectively. At each step in the prediction process, these weights would be reassigned to the model according to whether its predicted failure rate was highest, lowest, or in the middle.

Evaluation-based weights... - Weights are dynamically assigned to the components of an evaluation-based combination. Weights are assigned on the basis of changes in the prequential likelihood statistic (details in section 4.9.2) over a small number of observations.

<u>3.2.6. SETUP Menu</u>

Items in the "Setup" menu allow users to add and remove external applications from the "Edit" menu's "External editors" sub-menu, as well as to set significance levels for goodness of fit and trend tests.

Add application... - Lets you add the name of an external application to the "External applications" submenu in the main window's "Edit" menu. To add an external application to this sub-menu, you specify the name of the application, the subdirectory in which it is found, and the name that should appear on the "External editors" submenu. The application can then be invoked from the "External applications" submenu.

Remove application... - Allows you to remove external applications from the "External applications" submenu in the main window's "Edit" menu. If you remove an application, its name no longer appears on the "External applications" submenu. The "Notepad" entry on the "External applications" submenu is a permanent part of the CASRE configuration, and cannot be removed.

Remember Most Recent Files... – After you open a file for the first time, or save it under a new name with the "Save as" File menu item, the name of that file appears at the top of a list of file names kept at the bottom of the File menu. This capability allows you to specify the length of that list, from 0 to 9 entries.

GOF Significance… - Allows you to set the significance value, α, for the goodness of fit tests. When goodness of fit tests are run, there will be an indicator whether the model fits the data at the α% significance level.

Laplace Test sig… - Allows you to set the significance value, α, for the Laplace trend test. When this trend test is run, there will be a key in the graphic display window telling you the value that the test statistic must have to reject the null hypothesis at the α% significance level.

3.2.7. PLOT Menu

The "Plot" menu contains only one item, allowing you to create a new graphic display window.

Create plot window - Allows you to create a new graphic display window if you have destroyed the previously existing graphic display window. This capability can be thought of as a safety feature in that even if the graphic display window is destroyed, another one can always be created to replace the one that was destroyed.

3.2.8. HELP Menu

The "Help" menu provides on-screen help for users.

Help for help - Gives general information on how to navigate through the help system. No CASRE-specific information is given here.

Keys help - Describes function keys and key combinations that can be used to perform specific CASRE operations.

Help index - Gives help on individual menu items for each of the two CASRE windows. The index is organized in the same fashion as the menu trees for the windows.

About... - Identifies CASRE's authors and gives the CASRE version number.

3.3. Graphic Display Window

The menu tree for the graphic display window is given below. Plots of the failure data as well as model results are displayed in this window.

- "Plot"
  - "Save plot as..."
  - "Draw from file..."
  - "Setup printer..."
  - "Print plot...", GRAYED
- "Results"
  - "Select model results...", GRAYED
  - "Model results table", GRAYED
- "Display "
  - "Data and model results"
    - "Time between failures",
    - "Failure counts", GRAYED for time between failures data
    - "Failure intensity"
    - "Test interval lengths",GRAYED for time between failures data
    - "Cumulative failures",
      - "Over all data"
      - "From model start point"
    - "Reliability...", GRAYED
  - "Model evaluation", GRAYED
    - "Goodness of fit", GRAYED
    - "Prequential likelihood", GRAYED
    - "Relative accuracy", GRAYED
    - "Model bias", GRAYED
    - "Model bias trend", GRAYED
    - "Model bias scatter plot", GRAYED
    - "Model noise", GRAYED
    - "Model rankings...", GRAYED
      - "Rank summary...", GRAYED
      - "Ranking details...", GRAYED
- "Settings"
  - "Scale axes..."
  - "Show entire plot"
    - "Current plot"
    - "All plots"
  - "Draw data only"
  - "Draw results only"
  - "Draw data and results"
  - "Color"
    - "Choose colors..."

"Draw in color"
"Draw in B and W"
"X-Axis labeling"
"Test interval number", GRAYED for time between failures data
"Elapsed time", GRAYED for time between failures data
"Draw predictions as:"
"Scatter points"
"Line plot"
"Copy"
"Copy plot"
"Help"
"Help index"
"Keys help"
"Help for help"

3.3.1. PLOT Menu

Save plot as... - Allows you to save the currently-displayed contents of the graphic display window to a new disk file. It can also be used to rename an existing plot file.

Draw from file... - Draws the contents of a previously saved plot file in the graphic display window.

Setup printer... - Allows users to select and configure a printer in preparation for printing the contents of the graphic display window. The printer chosen with this menu option does not have to be the same as that chosen for the main window.

Print plot... - Prints the contents of the graphic display window on the printer chosen with the "Setup printer" option described above.

3.3.2. RESULTS Menu

The "Results" menu has two entries, the "Select model results..." and the "Model results table" items.

Select model results - This brings up a dialog box showing the models that have been run. You can select up to three of these models from a list of the models that were run. The results of these models will then be plotted in the graphic display window.

Model results table - Once you've run one or more models, you can bring up a window which will show the results in tabular form (the model results table). This detailed display complements the plots shown in the graphic display window.

3.3.3. DISPLAY Menu

The "Display" menu contains items allowing users to plot raw failure data, model results, and model evaluation statistics in a variety of ways. There are two main groupings in this menu - data and model result displays, and model evaluation displays.

3.3.3.1. Data and Model Results

The items in this portion of the "Display" menu allow users to display the failure data and model results in a variety of ways.

Time between failures - This menu item can be chosen for either type of failure data. Selecting this item produces a plot of time since the last failure as a function of failure number. Both the failure data and any selected model results are replotted if this item is selected.

Failure counts - This menu item is enabled only if a failure count data file was opened. Selecting this item produces a plot of the number of failures observed in a test interval as a function of the test interval number. Both the failure data and any selected model results are replotted if this item is selected.

Failure intensity - This menu item can be selected for either type of failure data. The failure intensity (failures oboserved per unit time) as a function of total elapsed testing time is displayed in the graphic display window. Both the failure data and any selected model results are replotted if this item is selected.

Test interval lengths - This menu item is enabled only if a failure count data file was opened. If this item is selected, a plot of the lengths of each test interval as a function of test interval number appears in the graphic display window. If model results are being displayed, the length of future test intervals that you've specified (see **Predictions...**, paragraph 3.2.4 above) will be shown as well as the test interval lengths that have actually been observed.

Cumulative failures - This menu item can be selected for either type of failure data. The total number of failures observed as a function of total time elapsed is displayed in the graphic display window. Both the failure data and any selected model results are plotted if this item is selected.

Reliability - This menu item can be chosen for either type of failure data. The way in which the reliability of the software changes as more failures are observed and corrected is plotted as a function of time for any selected model results as well as for the failure data. Most software reliability models assume that the reliability of software increases as failures are found and repaired.

3.3.3.2. Model Evaluation

The items in this portion of the "Display" menu are used to display various model evaluation criteria.

Goodness-of-fit - Selectable for both types of failure data. If failure count data is used, the Chi-square test is used to compute goodness-of-fit. For time between failures data, the Kolmogorov-Smirnov test is used. For each model executed, the goodness-of-fit statistic is computed and displayed in a table.

Prequential likelihood - Selectable for both types of failure data. This capability plots the negative of the natural log of the prequential likelihood for selected model results. The ratio of the prequential likelihood values for two models indicates how much more likely it is that one model is more applicable to the failure data than the other model.

Relative accuracy - Displays a scatter plot of the prequential likelihood ratio for the models selected for display. Given two models, this ratio indicates how much more likely it is that one model will produce more accurate predictions than the other model. The plot, then, tells you how much more likely it is that one model will produce more accurate predictions than the others.

Bias - Selectable for time-between-failures data only. This capability draws a plot which indicates whether the selected models tend to predict higher or lower times between failures (or failure counts) than are actually observed.

Bias trend - Selectable for time-between-failures data only. A plot is drawn which indicates any trends in the selected models' bias over time.

Bias scatter plot - Selectable for time-between-failures data only. This capability draws a scatter plot of the probability of failure before the next observed error vs. error number (test interval number), indicating the direction of the selected models' bias as well as the range of failure data in which the bias is observed.

Model noise - Selectable for time-between-failures data only. This menu item draws a table displaying the model noise measurement for each model that was run. The higher the noise figure, the less accurate the predictions made by the model.

Model rankings... - Selectable for both types of failure data. This menu item brings up a dialog box in which the user assigns weights to the following five criteria: goodness-of-fit, prequential likelihood, bias, bias trend, and model noise. Based on the weights for each criterion, each model that was executed is ranked, and a table of the model rankings is displayed. For failure count data, the weights for the bias, bias trend, and model noise criteria are locked at a value of 0.

You have a choice of two types of ranking displays. Choosing the "Rank summary" menu item displays the overall rank of each model with respect to the selected criteria, along with the current reliability predicted by the model. Choosing the "Ranking details" menu item gives the ranking details, in which the rank of each model with respect to each criterion is displayed, as well as the overall rank.

3.3.4. SETTINGS Menu

Scale axes... - Lets you set the origin of the plot in the graphic display window, and set the extent of the x and y axes. Once a plot has been rescaled, those settings remain with the plot until they are cancelled by using the "Show entire plot" capabilities described below, or until a new failure data file is opened.

Show entire plot - Brings up a sub-menu which allows you to cancel any scaling that was done on either the currently-displayed plot or on all plots. The two options on this sub-menu are "Current plot" and "All plots".

Current plot - If the currently-displayed plot has been rescaled, this option cancels the rescaling that has been done and shows the current plot in its entirety. Suppose you show one type of plot, rescale it, show a second plot, and then return to the first plot. You'll then see the scaled version of the first plot.

All plots - This option cancels rescaling that has been done on any plot. Suppose you display one type of plot, scale it, and then display a second type of plot. If you use the "All plots" option before redisplaying the first plot, the entire first plot will be shown, rather than the scaled version.

Redraw data - Redraws the failure data in the graphic display window. No modeling results are displayed.

Redraw results - Redraws only modeling results in the graphic display window.

Redraw data and results - Redraws both failure data and modeling results in the graphic display window.

Color - The three items in this section of the menu allow you to assign one of 25 colors to the model results and choose whether to draw in black and white or in color.

> **Choose colors** - This menu item brings up a dialog box which allows you to choose one of 25 colors for the raw data as well as three sets of model results.
>
> **Draw in color** - This menu item tells CASRE to draw the raw data and model results in color. This is CASRE's default setting.
>
> **Draw in B and W** - This menu item tells CASRE to draw the raw data and model results in black and white.

X-Axis labeling - There are two options in this section of the SETTINGS menu, "Test interval number" and "Elapsed time". These options determine how the x-axis for plots of failure count data should be labeled. These options are not available for time-between-failures data.

Test interval number - Use this option to label the x-axis for failure count data plots with test interval numbers. For the following types of plots, the x-axis will show the test interval number:

- Time between failures (assumes equal time between failures within a test interval)
- Failure counts
- Failure intensity
- Test interval lengths
- Cumulative number of failures
  - Over all data
  - From model start point
- Reliability

Elapsed time - Use this option to label the x-axis for failure counts data plots with the total elapsed time. This option affects the same types of plots as the "Test interval number" option.

Draw predictions as - This section of the "SETTINGS" menu determines the way that model predictions are drawn. There are two settings - "Scatter points" and "Line plot".

Scatter points - Use this option to draw model predictions of future times between failures, future reliability, etc., as scatter points. This is the default setting.

Line plot - Use this option to draw model predictions of future times between failure, future reliability, etc., as lines through the predicted points. In this case, the scatter points are not drawn.

3.3.5. COPY Menu

There is only one item in this menu, the purpose of which is to create an additional copy of the graphic display window.

Copy plot - Use this option to make multiple copies of the graphic display window. Each time you use this option, an additional graphic display window will appear on the screen. Initially, this window is the same as the window from which it was copied. However, you can change it to display a different set of model results, or you can change the way in which the data and model results are shown.

3.3.6. HELP Menu

The items in the graphic display window's "Help" menu perform the same functions as those in the main CASRE window's "Help" menu. The only differences are that:

1. The help index is organized around the graphic display window's menu tree rather than that of the CASRE main window.
2. The topics under "Keys help" are directed toward controlling the contents of the graphic display window.
3. There is no "About..." menu item.

3.4. Model Results Table

The menu tree for the model results table is given below. This window appears after you've selected the "Model results table" in the graphic display window's "Results" menu. Detailed modeling results, such as reliability estimates, model parameter estimates, and residuals (actual data minus model estimates), are displayed in this window in tabular form.

"File"
- "Save as..."
- "Print setup..."
- "Print", GRAYED

"Results"
- "Select results..."
- "Previous model", can also use F9 function key
- "Next model", can also use F10 function key

"Help"
- "Help index"
- "Keys help"
- "Help for help"

3.4.1. FILE Menu

The options in the "File" menu let you save the model results table to a file, or let you print it out.

Save as - This menu item works in the same way as the "Save as" item in the main window's "File" menu, with the following addition. In the "Save as" dialog box, you can select one of two radio buttons to decide whether you'll be saving the tables for all of the model results (default), or only for the currently-displayed model. The model results tables are saved as ASCII text files.

Print setup - This menu item works in the same way as the "Print setup" menu item in the main window's "File" menu, except that instead of choosing a data range to print, you select one of two radio buttons to choose whether you'll be printing the results tables for all models that were run (default), or only the table for the currently displayed model.

Print - Print out the selected model results table(s) on the selected printer.

3.4.2. RESULTS Menu

The three items in this menu control which set of model results will be displayed in the model results table.

Select results - Selecting this item brings up an alphabetized list of all of the models that have been run. You can scroll through this list and select the model whose results you want to display in the model results table.

Previous model - You can use this item to step backward through the list of models that were run. You can also press the "F9" function key to do this.

Next model - You can use this item to step forward through the list of models that were run. You can also press the "F10" function key to do this.

3.4.3. HELP Menu

The items in the model result table's "Help" menu perform the same functions as those in the main CASRE window's "Help" menu. The only differences are that:

1. The help index is organized around the model result table's menu tree rather than that of the CASRE main window.
2. The topics under "Keys help" are directed toward controlling the contents of the model result table.
3. There is no "About..." menu item.

4. A Sample CASRE Session

Figure 1 shows a flowchart of a typical CASRE session. Typically, you'll select a set of failure data, choose how far into the future you want to predict reliability, select and run models, look at model results, and determine which model is most appropriate to the data.

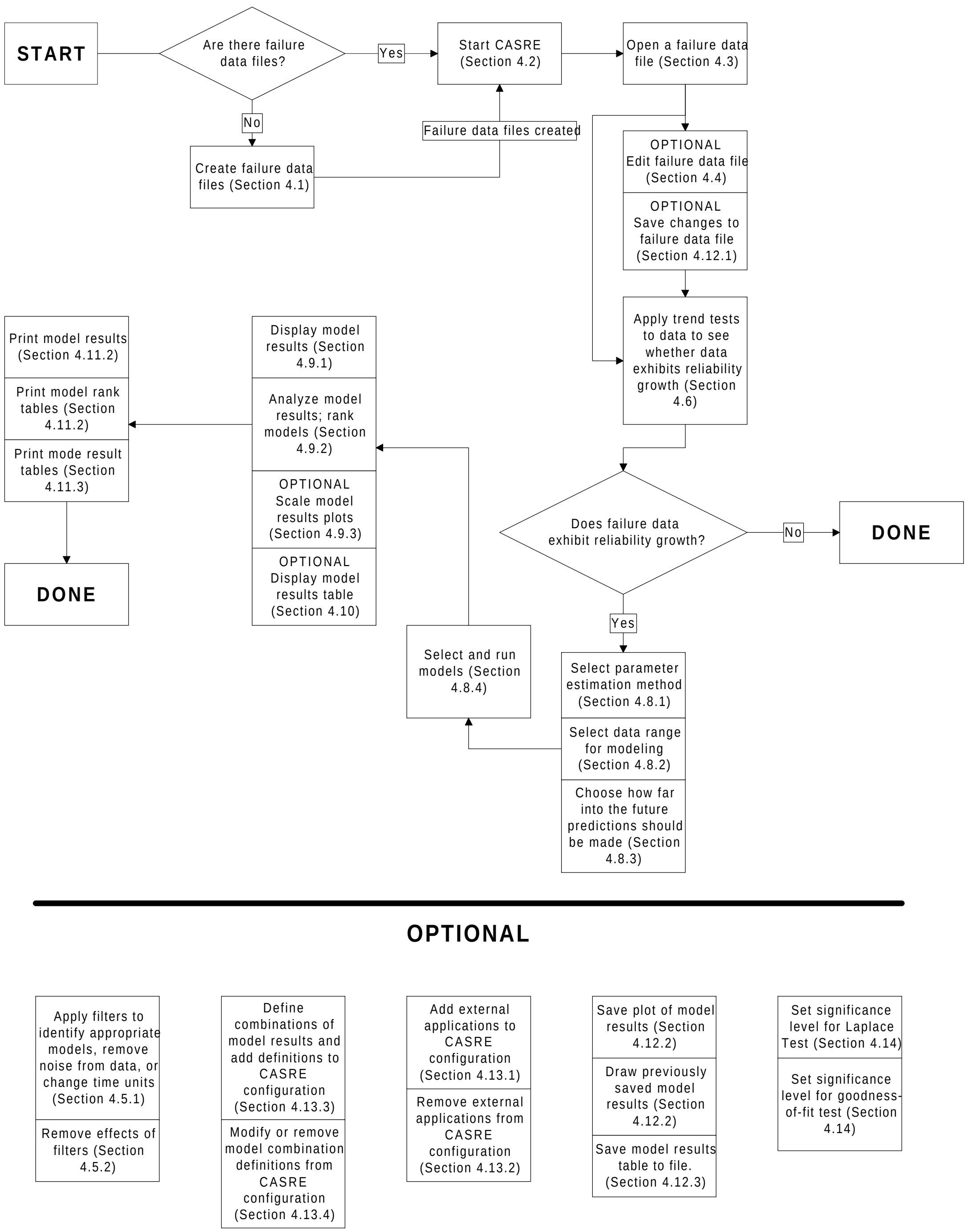

Figure 1 - Typical CASRE work session

This section of the user's guide presents a work session with CASRE. Using test data from actual software development efforts, we show you how to do the following:

1. Create a set of failure data.
2. Start CASRE.
3. Open a set of failure data.
4. Use an external application to change the failure data.
5. Apply filters and smoothing operations to the data.
6. Apply trend tests to the failure data to determine whether or not software reliability models should be applied.
7. Apply models to the failure data.
8. View the model outputs.
9. Print failure data and model results.
10. Save failure data and model results to disk.
11. Add external applications and combination model definitions to the CASRE configuration, or remove selected items from the CASRE configuration.
12. Set significance levels for goodness-of-fit tests and trend tests.
13. Set the length of the list of most recently used files that is kept in the File menu.
14. End the CASRE session.
15. Get help.

4.1. Creating a Failure Data File

Failure data files used by CASRE must be in the format summarized below and detailed in Appendix C. One way to create these files is to use a word processor or text editor to create ASCII text files of the appropriate format. Text editors, word processors, and other applications can be invoked as stand-alone applications, or you can run them from within CASRE (see paragraph 4.4 for more details). If you have a database in which information about failures is tracked during testing, however, you can use whatever capabilities your database program has to create a delimited ASCII file having the format and containing the information described in Appendix C. Field delimiters can be commas, tabs, or blanks. Data formats are summarized below:

For time between failures data, the data file format is as follows:

1. The first line represents the time units for the data file. You have a choice of seven keywords - Seconds, Minutes, Hours, Days, Weeks, Months, and Years. This specifies the time units for the times between successive failures given in the file.

The following points apply to the second and subsequent lines of the data file. Note that the second line in the file (the first line of actual failure data) represents the first failure observed, the third line represents the second failure observed, and so forth.

2. The first column is the current failure number.

3. The second column represents the time that has passed since the last failure was observed. Suppose that the value in the second row's second column has a value of 14,400. This means that 14,400 units of time have elapsed since the start of testing to the discovery of the first failure. If the value in the third row's middle column has a value of 14,400, this means that 14,400 units of time have elapsed since the first failure was observed. In general, the entry for error number "n" indicates the number of time units that have passed since failure "n-1" was observed.

   The values in the second column are measured in the time units given in the first line of the file. Suppose that the first line is the word "Seconds". For the time between failures values given above, this would mean that 14,400 seconds had elapsed from the start of test to the first failure, and that 14,400 seconds would have elapsed between the first and second failures.

4. The third column indicates the severity of the failure on a scale of 1 to 9. CASRE assigns no meaning to these severity categories; 1 could represent the least serious type of failure just as easily as it could represent the most serious type.

For failure count data, the format is as follows:

1. The first line of the file specifies the time units for that file, as for times between failures data.

The following applies to the second and subsequent lines of the data file.

2. The first column gives a sequential test interval number.

3. The second column specifies the number of failures that were observed during a given test interval.

4. The third column gives the length of the test interval. Test interval lengths do not have to be equal. They are measured in the time units given in the first line. Suppose that the first line specifies "Hours", and that for a specific test interval, the length is given as 48.2. This would a test interval 48.2 hours long.

5. The fourth column specifies the severity of the failures that were observed during a given test interval, on a scale of 1 to 9. **If no failures were seen in an interval, use a failure count of 0 and a severity of 0. CASRE will show the severity as "N/A". Also, CASRE assigns no meaning to severities.**

For failure count data, one test interval can occupy more than one line in the data file. For instance, it could happen that 14 failures of severity 1, 3 of severity 2, and 1 of severity 4 were seen in the first interval. This would mean that there would be three lines for test interval 1, looking like this:

```
1   14.0   56.0   1
1   3.0    56.0   2
1   1.0    56.0   4
```

If there were no failures seen in the second interval, the fourth line of the file would be as follows:

```
2   0.0   56.0   0
```

In the main window, this line would be shown as follows:

```
2   0.0   56.0   N/A
```

4.2. Starting CASRE

Before starting CASRE, make sure that CASRE has been installed as described in section 2, and that you've started Windows by entering "win<CR>" at the DOS prompt. The CASRE icon will then appear in the CASRE window that you created in Section 2. If the CASRE window is not shown, perform the following steps to choose CASRE's program group and display it:

1. Choose the Program Manager's "Window" menu.
2. On the menu that appears, select the name "CASRE" if it appears.
3. If the name "CASRE" does not appear, select the last item in the menu, "More windows...". This brings up a dialog box with a scrolling list of available windows. Use the mouse to scroll through the list until you find "CASRE", Click on the name CASRE, then click on the "OK" button in the dialog box.

Once the CASRE window is on your screen, start CASRE by using the mouse to double-click on the CASRE icon within that window. The CASRE main window should then appear as shown in Figure 2 below.

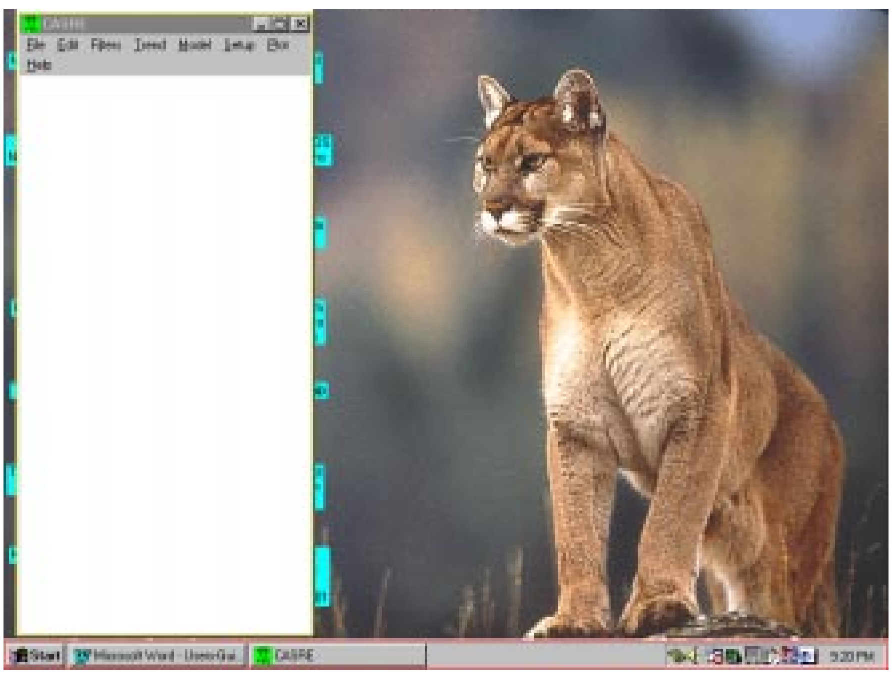

Figure 2 - Initial CASRE Display - CASRE main window

4.3. Opening a Data File

To open a file of failure data, first use the mouse to select the main window's "File" menu, then use the mouse to select the "Open" item in that menu. The dialog box shown in Figure 3 on the next page will then appear, showing the directories accessible from the current directory in the left-most scrolling list ("Directories"), and the files in that currently selected directory in the right-most scrolling list ("Files"). For the examples shown in the next few figures, the data we'll be looking at is in the subdirectory "c:\casre\data." The "Directory is:" field in the dialog box in Figure 3 shows the current subdirectory. To change directories, use the mouse to double-click on the name of the subdirectory in the "Directories" list to which you want to go.

The parent of the current subdirectory is shown in the list as "[..]". You can think of each double-click as being equivalent to entering "cd <directory>" at the DOS prompt. When you've reached the subdirectory in which the data is, you can open a file in one of the three following ways:

1. Click the mouse on the name of the desired file in the "Files" scrolling list. Then use the mouse to select and click on the "Open" pushbutton. The file will then be opened, resulting in the plot shown in Figure 4 below, **or**
2. Enter the name of the file in the "File Name" field and then use the mouse to click on the "Open" pushbutton, **or**
3. Use the mouse to double-click on the name of the desired file in the "Files" scrolling list.

In this case we've opened the file "tbe_tst2.dat" in the subdirectory "c:\casrev3\data".

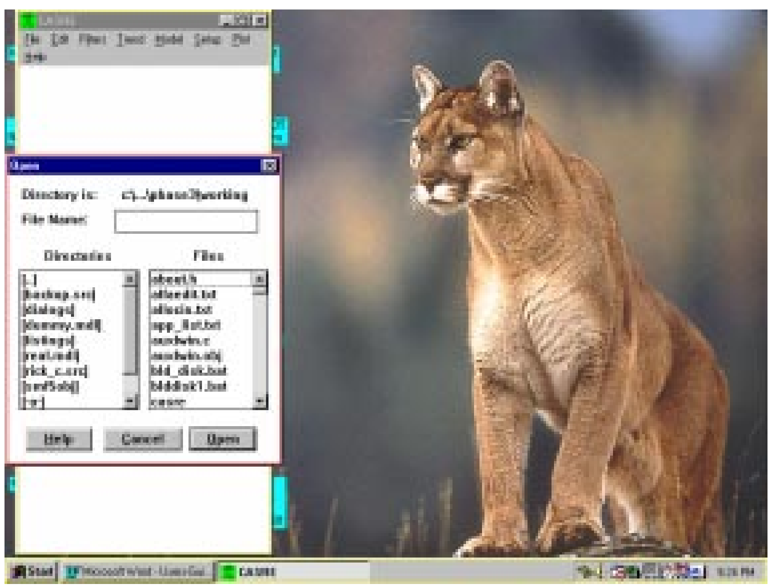

Figure 3 - Opening a failure data file

When a failure data file is opened, the text of the file is shown in the main window, while a plot of the data is shown in the graphic display window. If the file contains times between failures, the default plot in the graphic display window is that shown in Figure 4 on the next page. For files containing failure counts data, the default contents of the graphic display window is a plot showing the number of failures observed in each test interval as a function of the test interval number, as shown in Figure 5 on the next page. The following types of plots are available after a data file has been opened, but before any models have been run:

Times between Failures Data	Failure Counts Data
	Failure counts
	Test interval lengths
Failure Intensity	Failure Intensity
Times between failures	Times between failures
Cumulative number of failures	Cumulative number of failures
Cumulative number of failures from modeling start point	Cumulative number of failures from modeling start point
Reliability	Reliability

Table 1 – Available Plot Types

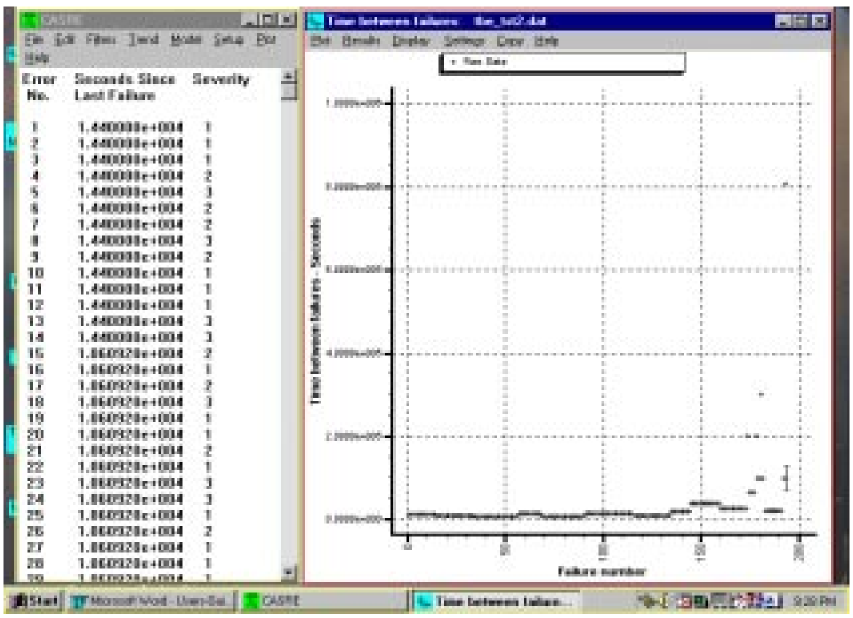

Figure 4 - Initial display for time between failures data - CASRE main window and graphic display window

Examples of these plots are shown in Figures 5-15. Here we explain how to display each of these types of plots.

Failure counts - choose the "Failure counts" item in the graphic display's "Display" menu. This plot, shown in Figure 5, displays the number of failures observed vs. test interval number for failure count data only. Although we don't discuss the details here, you can change the way in which the x-axis is labelled for failure count data. The default labelling is test interval number, as shown.

However, you can choose to have the x-axis show the total elapsed time since the start of testing. See paragraph 4.9.3 for more details.

Failure intensity - choose the "Failure intensity" item in the graphic display's "Display" menu. This plot, shown in Figure 6 on the following page, displays the failure intensity (failure observed per unit time) for both time between failures and failure counts data. Although we don't discuss the details here, you can change the way in which the x-axis is labelled for failure count data. The default x-axis labelling for failure counts data is test interval number. However, you can choose to have the x-axis show the total elapsed time since the start of testing. The default failure intensity display for failure count data is identical to that shown in Figure 5, except that the y-axis is labelled in terms of the number of failures observed per unit time interval. See paragraph 4.9.3 for more details.

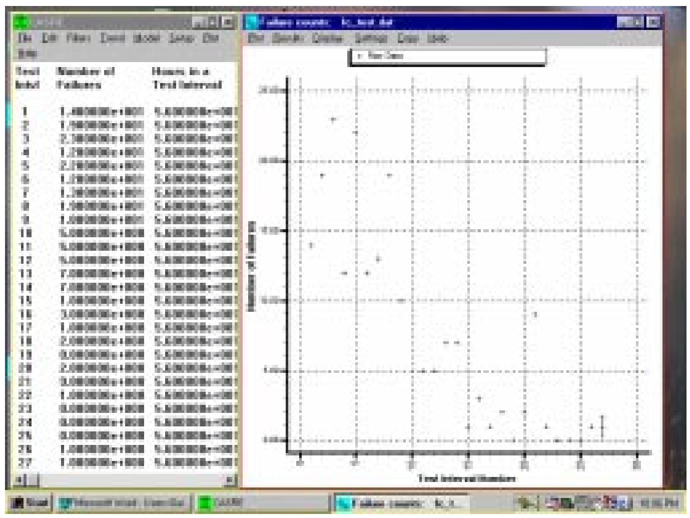

Figure 5 - Initial display for failure counts data

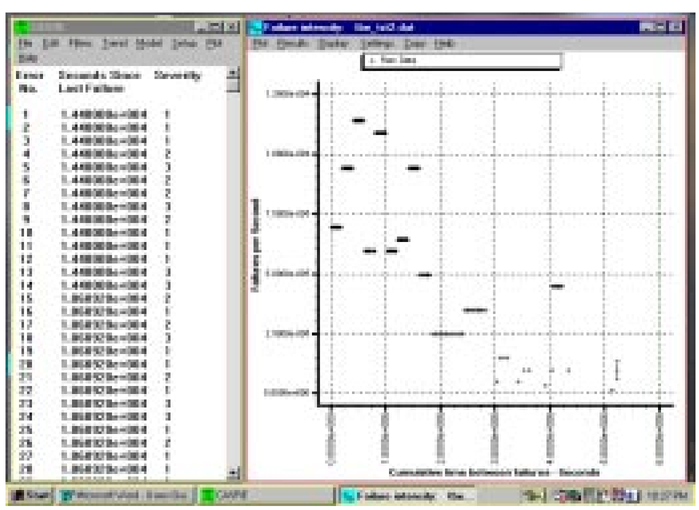

Figure 6 - Failure intensity for time between failures data

Cumulative number of failures - First select the "Display" menu for the graphic display window. Next, select the "Over all data" item in the "Cumulative failures" sub-menu. This shows a plot of the cumulative number of failures vs. total elapsed time for time between failures data, and a plot of cumulative number of failures vs. test interval number of failure count data. These plots are shown in Figures 7 and 8 below.

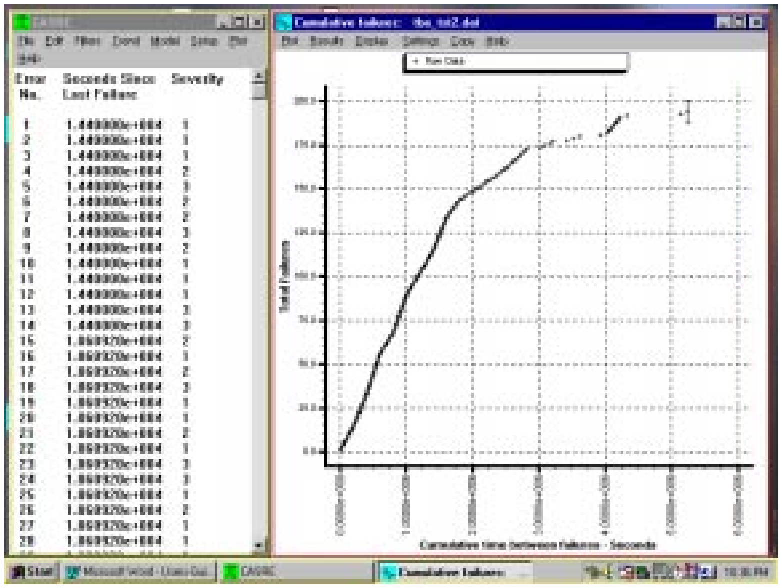

Figure 7 - Cumulative number of failures plot for time between failure data

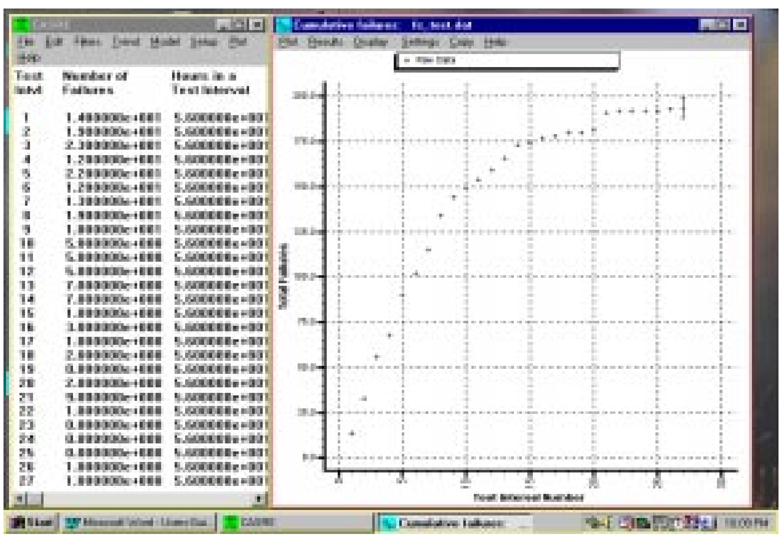

Figure 8 - Cumulative number of failures display for failure count data

Cumulative number of failures from model start point - First select the "Display" menu for the graphic display window. Next, select the "From model start point" item in the "Cumulative failures" submenu. For time between failures data, this shows a plot of the total number of failures, **starting at the point you've specified as the model starting point**, vs. total elapsed time. For failure count data, the x-axis shows the test intervals. These plots are shown in Figures 9 and 10 below. In Figure 9, we've set the model starting point to be 20, while in Figure 10, we've set the model starting point to be 5. See how these two figures are different from Figures 7 and 8.

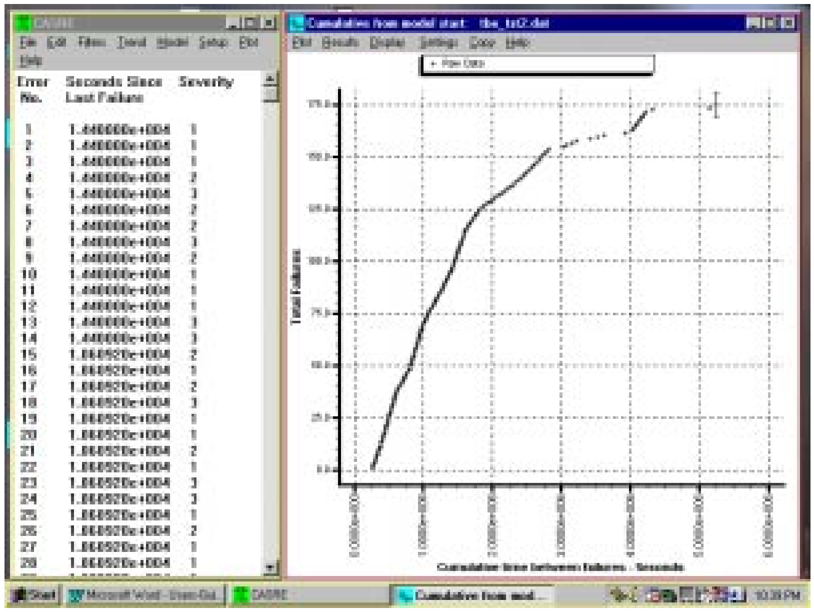

Figure 9 - Cumulative number of failures from model start point for time between failures data

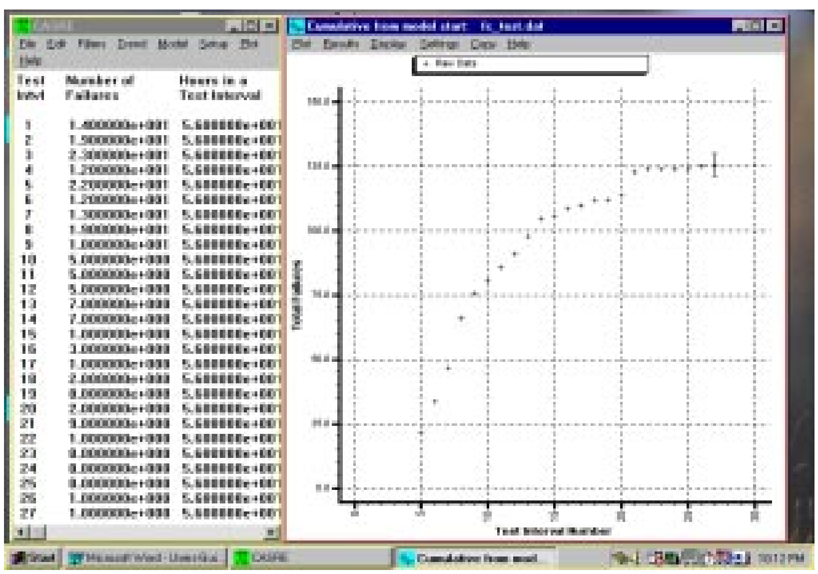

Figure 10 - Cumulative number of failures from model start point for failure count data

Reliability - You can display the reliability indicated by the raw data. Do this by selecting the "Reliability..." menu item in the graphic display window's "Display" menu. For time between failures data, the failure rate is taken as the inverse of the time between successive failures. For failure count data, the failure rate is computed as the number of failures per test interval divided by the test interval length. To compute reliability, you have to specify the amount of time for which you want to determine reliability. This amount of time might be specified by a reliability requirement. For example, a system might be required to run for 14,400 seconds and have a 98% probability of not failing during those 14,400 seconds. You might want to see the current probability for running for 14,400 seconds without failures, as well as how that probability has changed from the start of test. For this example, you'd enter "14,400" into the dialog box shown in Figure 11 below. This dialog box appears after you've selected the "Reliability..." menu item. After you've entered this value into the dialog box, select and click on the "OK" button. The reliability indicated by the raw failure data will then be drawn. Figures 12 and 13 on the following page show the reliability curves for time between failures data and failure count data, respectively. Here's what Figure 12 tells us. If we'd stopped testing after 1,000,000 seconds, the probability of running successfully for another 14,400 seconds would be about 0.2. If we'd stopped testing after 2,000,000 seconds instead of 1,000,000, the probability of running without failure for another 14,400 seconds would be about 0.7. Finally, if we stop testing right now, after having observed 194 failures (about 5,250,000 seconds of testing time), the probability of running without failure for another 14,400 seconds is about 0.85. Figure 13 is identical to Figure 12, except that it's for failure count data. In Figure 13, we're looking at the probability of running for another 28 hours without failure. This value of 28 hours was specified using the same dialog box as shown in Figure 11. Note that in Figure 13, there are some test intervals (19, 23, 24, and 25) for which a reliability of 1.0 is plotted. No failures were observed during these intervals.

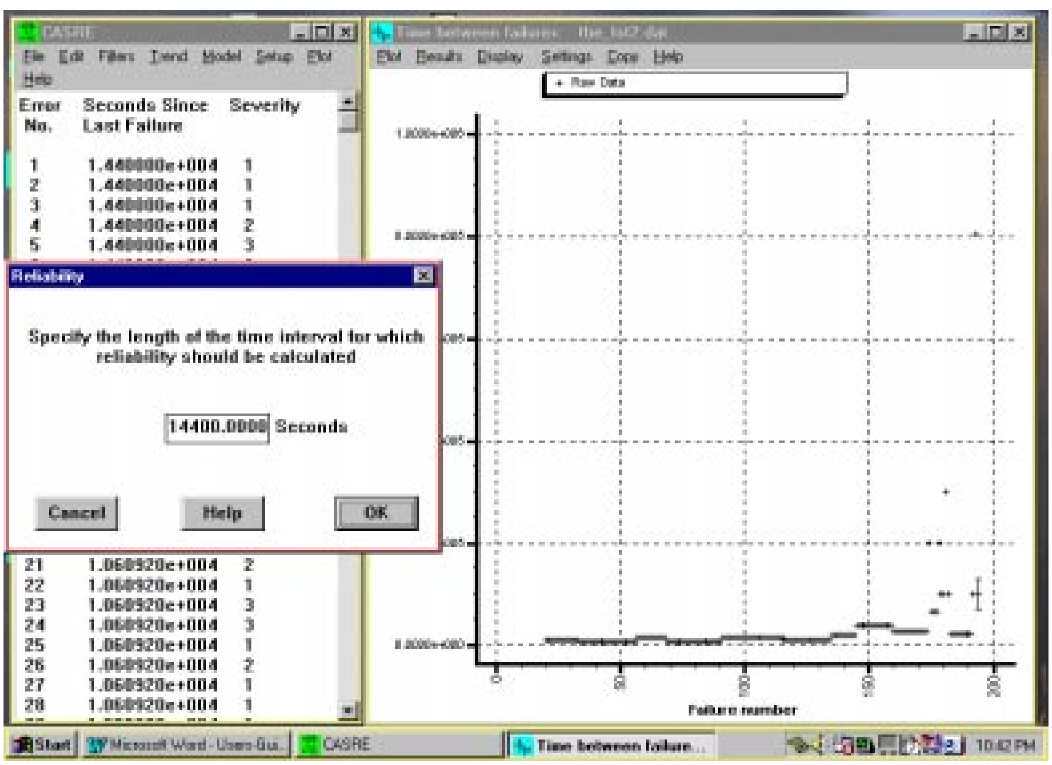

Figure 11 - Specifying the amount of time for which reliability will be computed

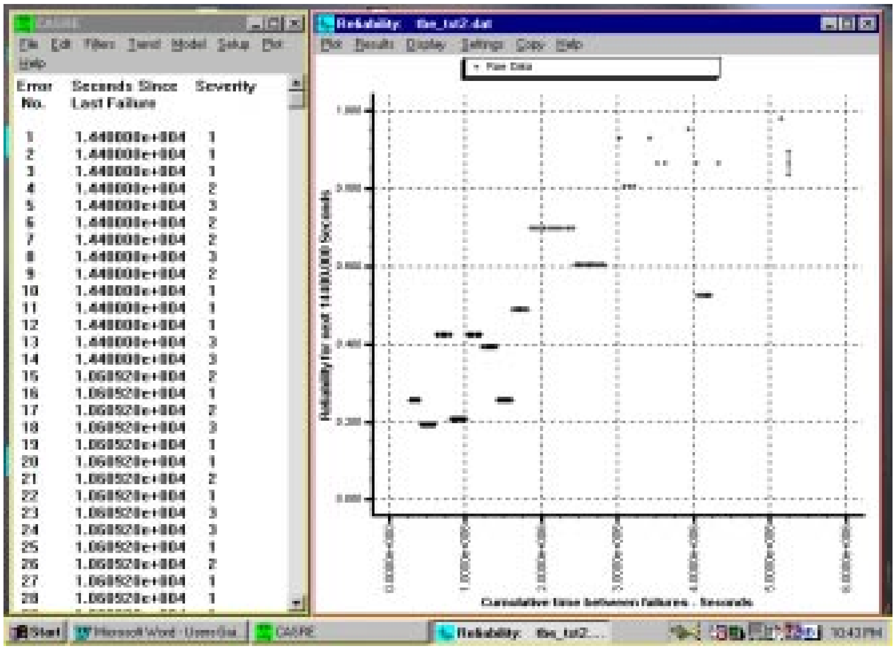

Figure 12 - Reliability for time between failures data

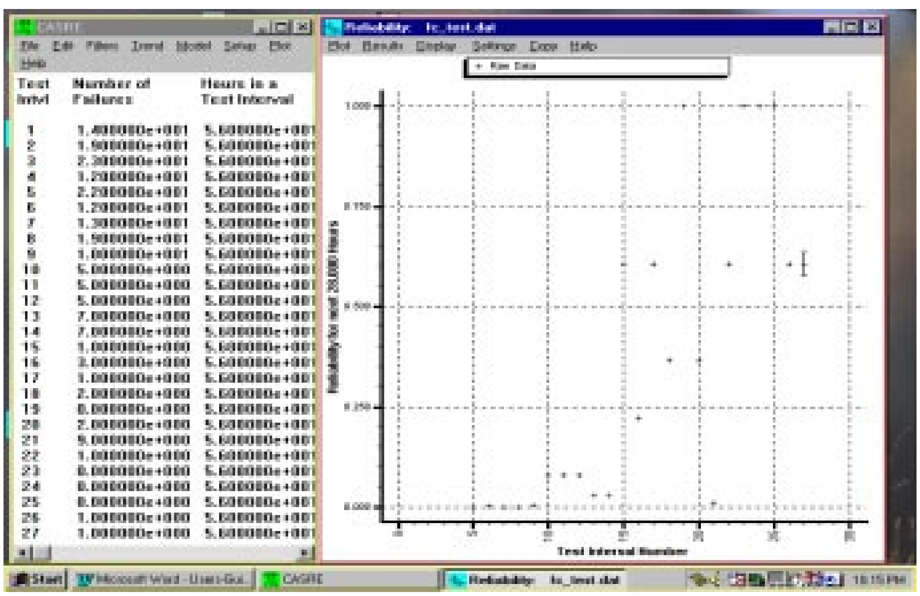

Figure 13 - Reliability for failure count data

Time between failures - Produce this display by selecting the "Time between failures" item in the graphic display window's "Display" menu. For time between failures data, you'll get the plot shown in Figure 4. For failure count data, you'll get the plot shown in Figure 14 on the following page. For failure count data, we assume that failures are evenly distributed within each test interval. In Figure 14, notice that the times between failures for test intervals 19, 23, 24, and 25 are represented by different symbols than for the other intervals. In this type of plot, this symbol means that the time between failures is infinite. For the data shown in Figure 14, the failure counts for these test intervals were 0, as shown in Figure 5.

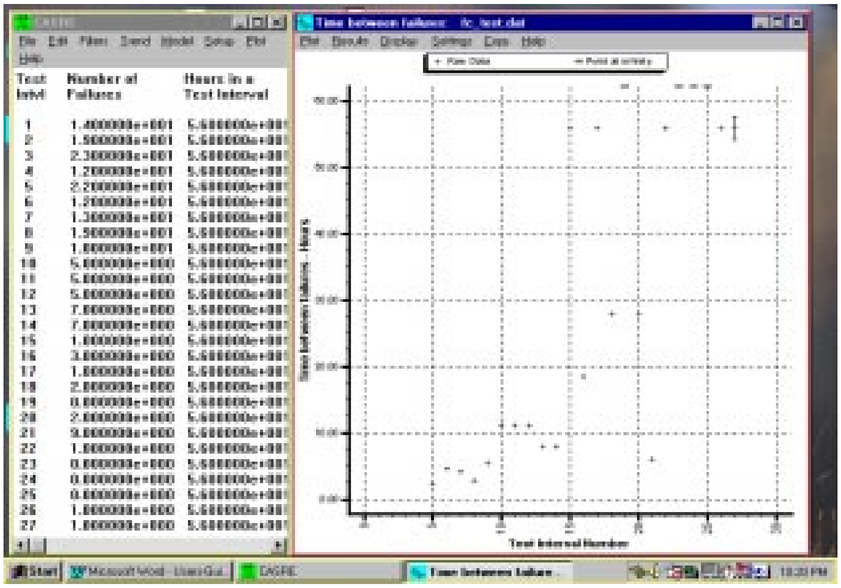

Figure 14 - Time between failures for failure count data

Test interval lengths - This plot, shown in Figure 15 below, displays the lengths of each test interval for failure count data. This plot is produced by selecting the "Test interval lengths" menu item in the graphic display window's "Display" menu. For each test interval, the length of the test interval is shown. For the data shown in Figure 15, each test interval was 56 hours long. However, it is not necessary that all test intervals be of equal lengths.

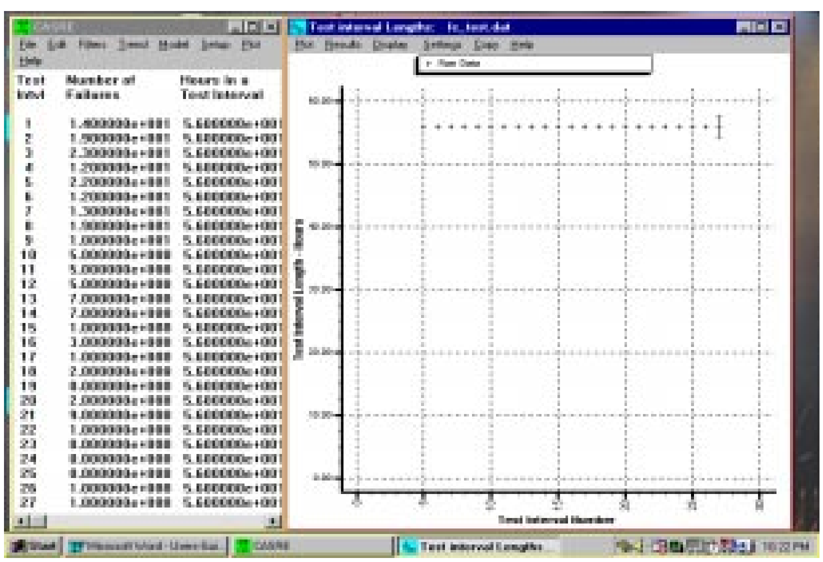

Figure 15 - Test interval lengths for failure count data

After selecting and opening a set of failure data, you can modify it in several ways. For instance, you can invoke an external application, such as Word or Excel, to edit the data set. You can also apply global filters to the data, and you can use the filtered data as input to software reliability models that you choose. In the next section, we'll describe how to run external applications to modify the failure data.

You can also open a file by selecting its name from the list of filenames that appears at the bottom of the “File”menu, as shown in Figure 16.

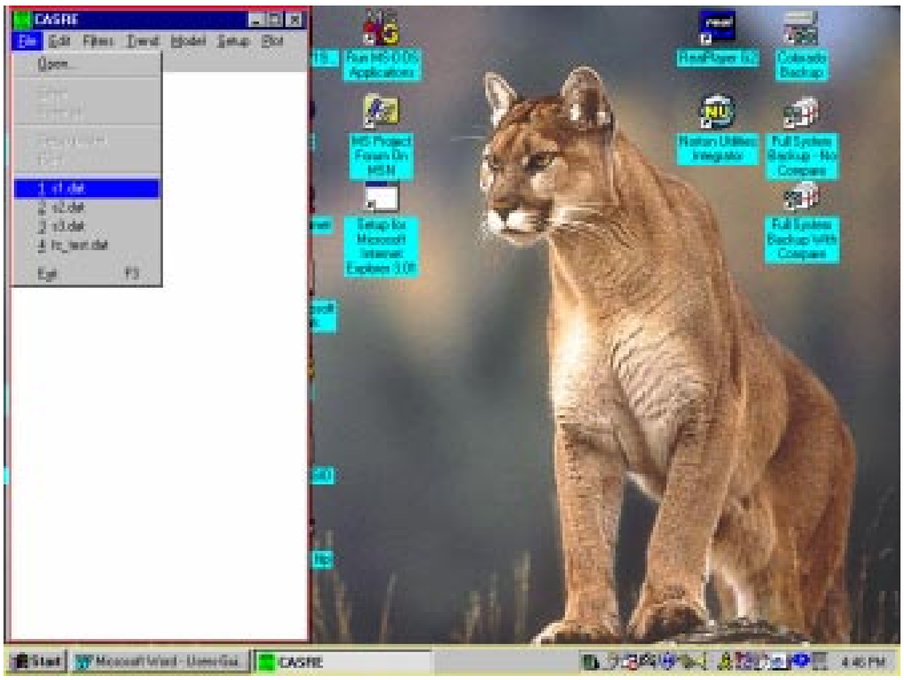

Figure 16 – Opening a file by selecting its name from the File menu list

In Figure 16, we see that the four data files “s1.dat”, “s2.dat”, “s3.dat”, and “fc_test.dat” were previously opened. Selecting the name “s1.dat” from the File menu will open that file. The “Remember Most Recent Files” item in the Setup menu allows you to specify how many filenames will be kept in the list. See section 4.15 for further details.

NOTE: CASRE version 3.0 will open failure count data files created for earlier versions of CASRE. However, it will not display the data required to run the Brooks and Motley models, since those models have been removed from version 3.0. In addition, if you use CASRE’s data modification capabilities (e.g., applying a filter) to change an old failure counts data file, and then save it under its original name by doing a “Save”, the data file will no longer be usable by earlier versions of CASRE – the data required to run the Brooks and Motley models will have been removed from the file. If you want to save modifications you’ve made to an old failure counts data file, use the “Save as” capability to save it under a new name. See the user’s guide for CASRE version 2.0 for a detailed description of the old failure counts data file format.

4.3.1. Moving and Sizing Windows

At times, you may want to make one window larger, reduce its size, remove it from the screen altogether, or move it from one place in the display to another. We'll explain briefly how to do this. For more details, see the Windows user's guide.

There are two ways to increase the size of a window. You can make a window cover the entire screen by using the mouse to click on the button in the upper-right hand corner of the window having the up-arrow symbol. This is the right-most button along the top edge of the window. Once you've done this, the symbol on the button will change to an up-arrow/down-arrow symbol. When you want to return the window to its former size, use the mouse to click on the same button. You can also stretch or shrink a window to make it a different size. To do this, first use the mouse to move the arrow-shaped cursor (**mouse cursor**) until it's on top of one of the window borders. If you want to stretch or shrink the window horizontally, move the mouse cursor to the left or right window border. If you want to stretch or shrink the window vertically, move the mouse cursor to the top or bottom border. If you want to stretch or shrink in both directions at once, move the mouse cursor to the border at one of the corners. At this point, the mouse cursor should have turned into a double-headed arrow, telling you that you can start resizing the window. Now push and hold down the left mouse button, at the same time moving the mouse cursor in the direction you want to expand or shrink the window. A heavy, shaded border will appear, showing you how the window changes shape as you resize it. When the window is the size you want, release the left-most mouse button. The window will then be redrawn in its new size.

You can reduce a window to such a small size that the information in it is displayed in only a stylized form, and the menu bars are no longer visible. This stylized representation is called an **icon**. If you want to temporarily stop displaying a window, use the mouse to click on the button along the window's top edge that has the down-arrow symbol on it. The window will then shrink in size and be represented by an icon. To redisplay the window, move the mouse cursor on top of the icon, and double-click the left mouse button.

Sometimes a window is partially on top of another one that you want to see. For instance, the main window may be partially on top of the graphic display window. If this is the case, move the mouse cursor on top of the window you want to see (e.g. the graphic display window), then click on the left mouse button. The graphic display window will now be on top of the main window.

To move a window, just move the mouse cursor so that it's somewhere on the top bar of the window. For the main window shown in Figure 2, this would be in the shaded portion labeled "CASRE". While holding down the left mouse button, move the mouse cursor to where you want the window to go. A heavy, shaded border the size of the window will move with the mouse cursor, showing you how the window will change position. When the window is where you want it, release the left mouse button, and the window will be redrawn in its new position.

4.4. Editing the Data File

The editing capabilities in the current version of CASRE are mainly limited to the ability to invoke external applications from the "Edit" menu, although there is the ability to convert time between failures data to failure counts, and vice versa. To invoke an external application, choose the "Edit" menu's "External applications" sub-menu. This menu, shown in Figure 17 below, can have up to 65 external applications.

In Figure 17 the "Write" editor, included with Windows 3.1 and Windows 3.11, (Wordpad replaces Write in Windows95), is being selected as the external application to be invoked. See Appendix G for more details on using Wordpad. Once the selection has been completed, Wordpad will appear in its own window, as shown in Figures 18 and 19 below. The message box shown in Figure 18 telling you to click the "OK" button after you have finished using the external application also appears. CLICK THE "OK" BUTTON ONLY AFTER EXITING THE EXTERNAL APPLICATION. IF THE "OK" BUTTON IS CLICKED BEFORE THIS, YOU WON'T BE ABLE TO USE THE EXTERNAL APPLICATION TO MAKE CHANGES TO THE FILE CURRENTLY SHOWN IN THE MAIN WINDOW. In Figure 19, the message box shown in Figure 18 is still on-screen; it's just been placed behind the Wordpad window by using the mouse to click on the Wordpad window. In Figure 19, note that the file being edited with Wordpad is the same one currently being displayed in the main window and the graphic display window.

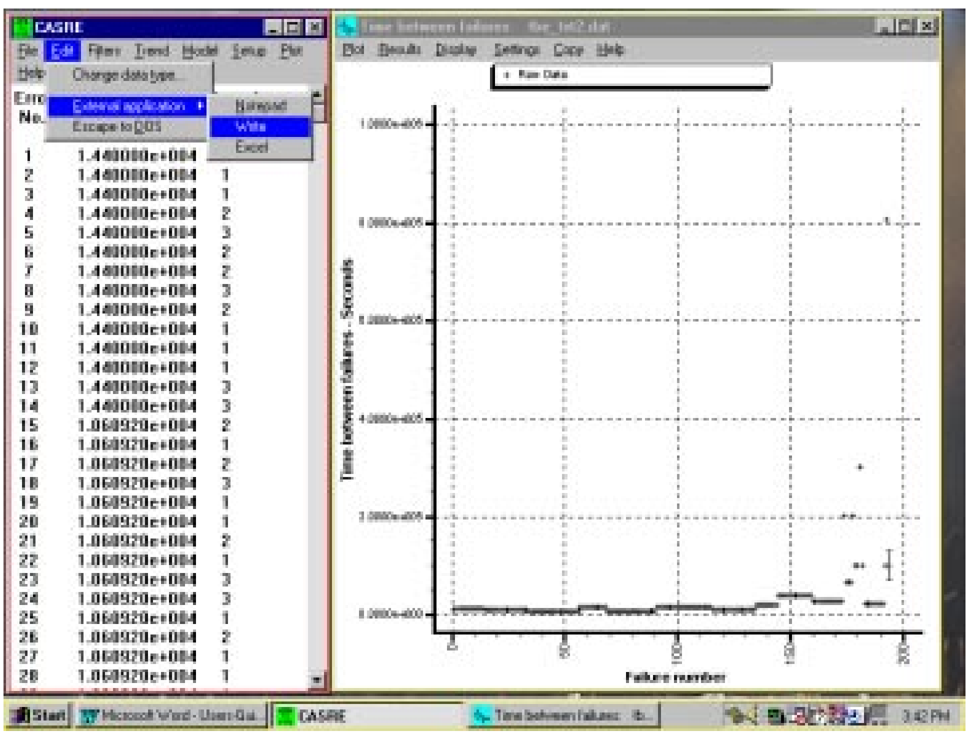

Figure 17 - Invoking an external application

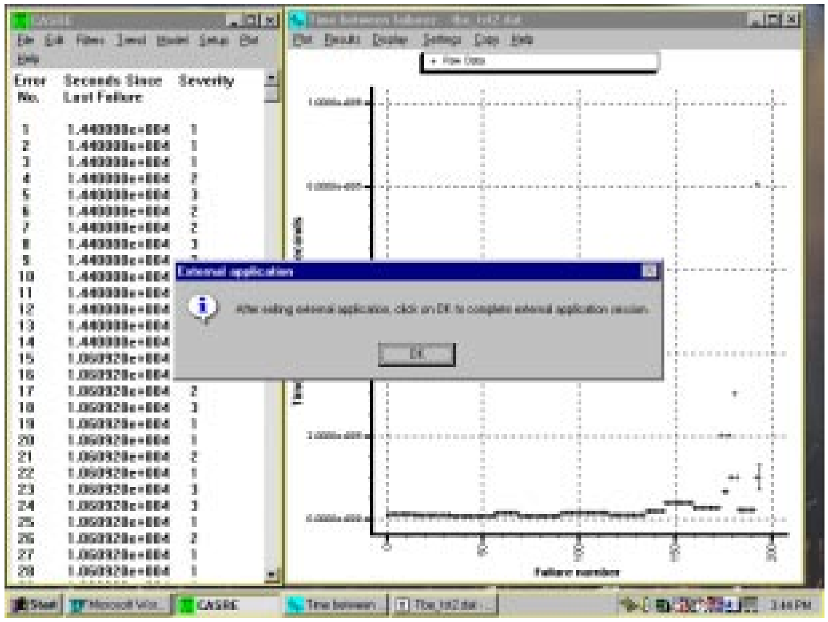

Figure 18 - Invoking an external application (cont'd)

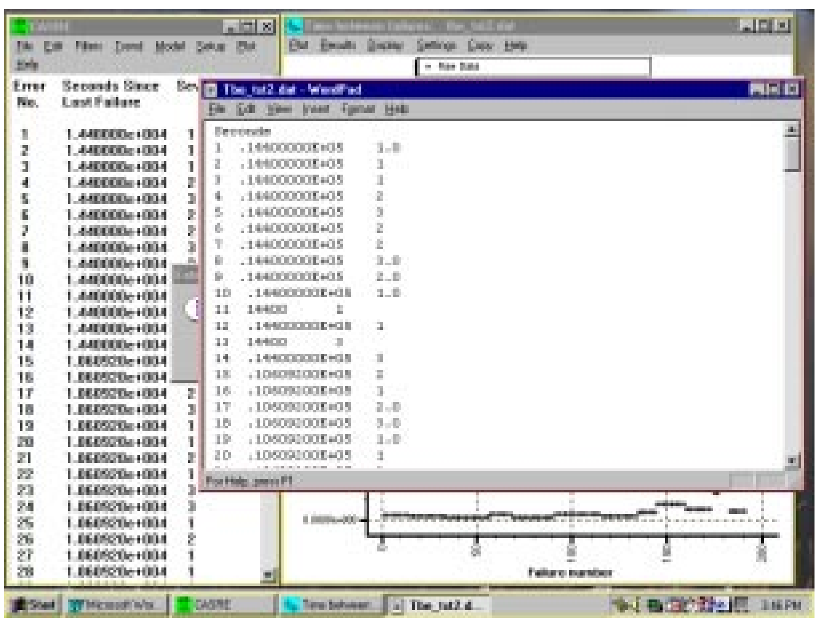

Figure 19 - Using an external application to edit the current set of failure data

After an external application has been invoked, there are no restrictions on its use other than those imposed by Windows, such as the amount of memory available on the system. If an editor or word processor has been invoked, any file on the system can be edited, including the failure data file that you're currently working with. If you delete the file while working in the external application, you will be prompted to open a new file.

The other main capability available in the "Edit" menu is the ability to convert time between failures data to failure count data, or vice versa. Two types of models are implemented in CASRE - those that take time between failures data as input, and those that take failure counts. These models are listed in paragraph 4.8.4. The capability to convert from failure counts to time between failures is included because you may have collected failure counts data for a development effort, but experience on other efforts indicates that the models using time between failures input give more accurate predictions. Another reason for having this capability is that you might have collected failure data in terms of time between failures, but you feel that the data collection process introduced too much noise into the data. Converting time between failures to failure counts may remove some of the noise in the data by effectively integrating it out as part of the data grouping operation.

The dialog box for converting from time between failures to failure counts data is shown in Figure 20 below. To do the conversion, the length of each test interval is entered into the dialog box's edit window as a floating point number. If the value typed into the edit window is some floating point number "X", the time between failures data is converted into failure count data for which the length of each test interval is shown as "X", and the failure counts for each test interval represent the number of failures observed for each interval of "X" units length. Figure 21 on the following page shows the results of converting the data set tbe_tst2.dat (first seen in Figure 4) from time between failures to failure counts, using a test interval length of 100,800 seconds. Note that the original file, "tbe_test.dat", is closed, and a new unnamed failure counts file, "<UNTITLED>", is created. In converting from time between failures to failure counts, you'll want to make sure that you don't specify a test interval length that is too short or too long. The following guidelines will help you choose an appropriate test interval length.

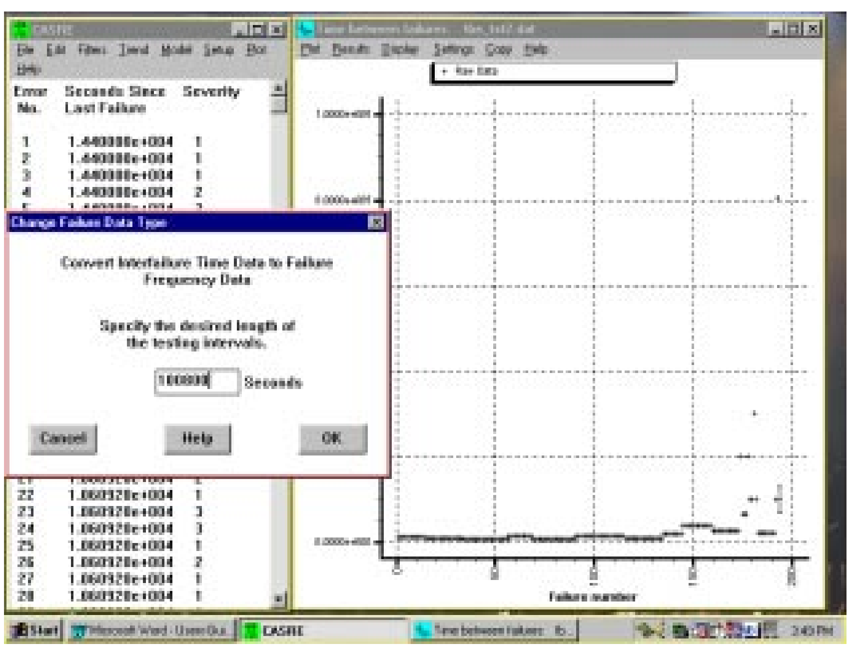

Figure 20 - Converting time to failures data to failure counts - selecting a test interval length

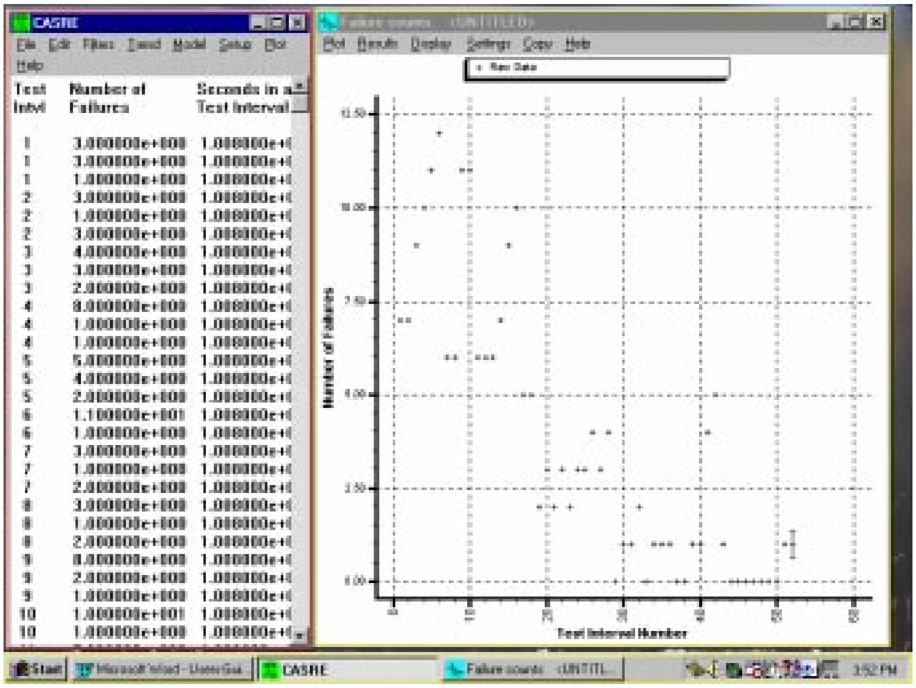

Figure 21 - Converting time between failures data to failure counts - conversion complete

a. First, display the time between failures data in the form of cumulative number of failures, as previously shown in Figure 7. By looking at the x axis, you'll be able to see the total amount of time that has elapsed since you started testing. You'll want to choose a test interval length such that the total elapsed testing time divided by the test interval length comes out in accordance with the following guidelines.

b. The test interval length must be chosen such that you have no more than 3000 test intervals. Otherwise, you'll get a message saying that not enough memory could be allocated to do the conversion.

c. Choose the test interval length such that you get at least 30 or more test intervals after the conversion is done.

d. Choose a test interval length such that no more than 10-20% of the test intervals have failure counts of 0.

Figure 22 on the following page shows the dialog box that appears if you want to convert from failure count data to times between failures. You can choose whether the times to failure for each test interval should be randomly assigned (default setting), or if they should be of equal length. Experience indicates that random assignment yields more accurate model results. Figure 23, on the next page, shows the results of converting failure counts data to time between failures data. As with converting times between failures to failure counts, the original file is closed, and a new, unnamed data file designated "<UNTITLED>" is created.

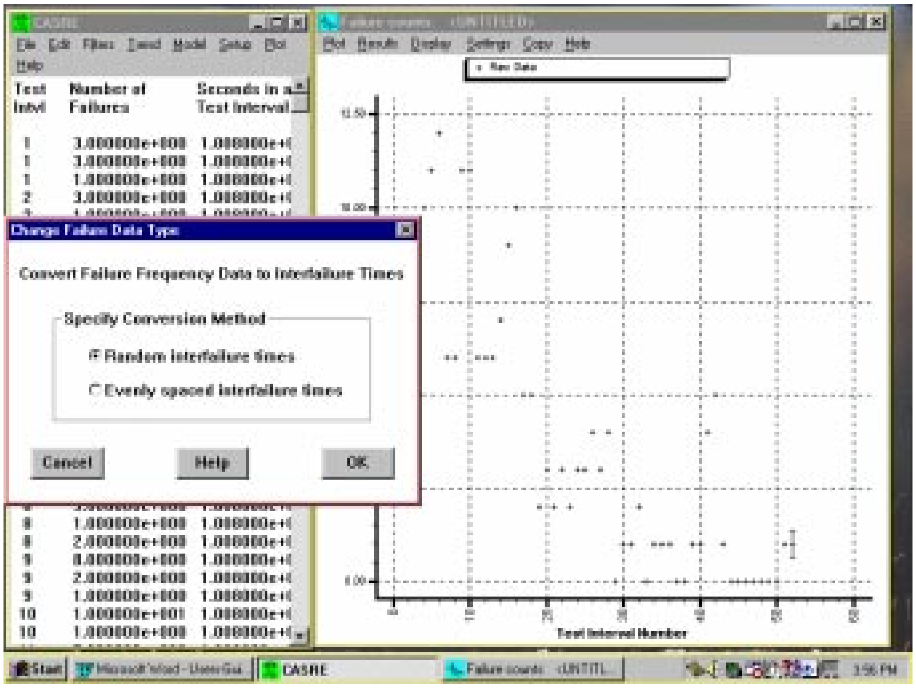

Figure 22 - Converting failure counts to time between failures - choosing conversion method

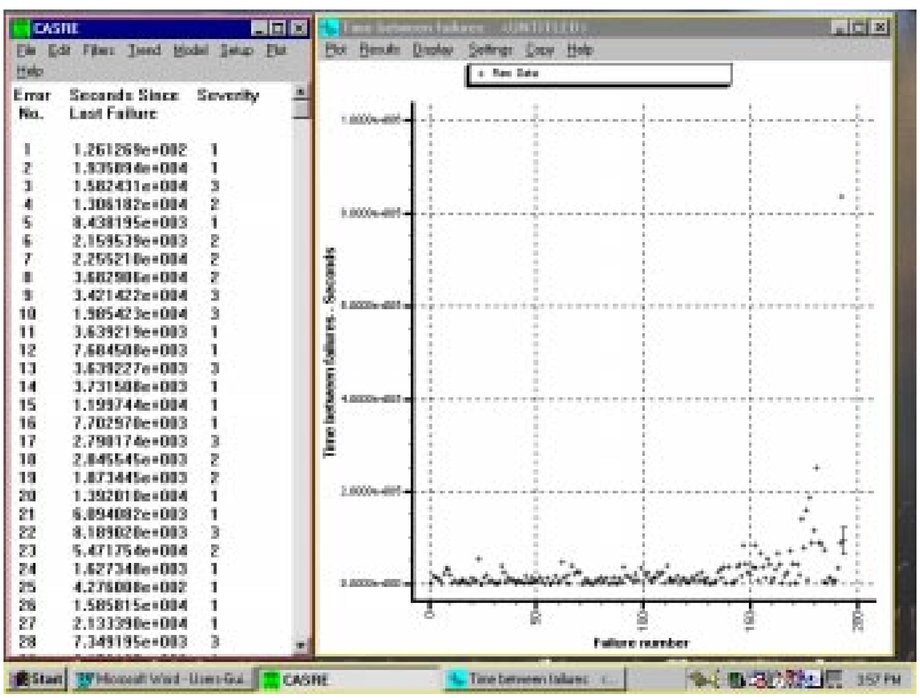

Figure 23 - Converting failure counts to time between failures - conversion complete

IMPORTANT: When you're converting from failure counts to time between failures data, there are two problems that you can encounter:

1. **If the failure counts data have any fractional values, the conversion won't work because of the way memory allocation handles fractional values. For instance, you might have applied a filter (see 4.5), or perhaps you're working with a file that was filtered, then saved to disk. Don't try the conversion if your failure counts data is not all whole numbers. You can use the "Round" filter (see paragraph 4.5.1) to make the failure counts into whole numbers.**

2. **If your failure counts data are all whole numbers, don't apply any filters (except "Select Severity" - see paragraph 4.5.1 for more details) until you've done the conversion. This is to avoid any chance of having fractional failure counts.**

You should also note the following behavior of the random number generator used in converting failure counts to time between fail-ures data. Suppose that during a CASRE session, you're converting a failure counts data file, "A", to times between failures, and that this is the first time you've done this conversion during the session. The results, which we'll call "A_1", will be a data set similar to Figure 23. If you repeatedly convert the file "A" during the session, you'll produce more time between failures results we'll call $A_2, A_3, \ldots, A_n$. Now suppose that in a later session, you again convert "A" to times between failures. The first time you convert "A", the result will be identical to the A_1 you got during the earlier session. If you convert "A" n-1 more times during the later session, the results you get will be identical to the results $A_2, A_3, \ldots, A_n$ you got during the earlier session.

4.5. Filtering and Smoothing the Data

CASRE includes five types of filtering capabilities:

1. Shaping and scaling filters for changing the shape of the failure data curve.
2. A filter for changing the time units for a failure data set. For instance, if a time between failures data set records time between failures in seconds, you can apply this filter to change the times between failures to minutes.
3. A Hann window for removing noise from the failure data.
4. The capability of selecting a subset of the failure data based on severity classification.
5. A filter for rounding the failure data to the nearest whole number.

The menu items in the "Filter" menu's "Shaping and Scaling" section may be used to change the shape of the failure data curve to make it easier to predict future behavior. These filters can be useful for more easily spotting trends in the data and identifying appropriate models. For instance, a logarithmic transformation can be used to make the time between failures data shown in Figure 4 into the nearly horizontal line shown in Figure 24 below. This indicates that models assuming an exponential distribution of time between failures, such as the Musa Basic or the Nonhomogeneous Poisson Process models, may produce good predictions. **THE ONLY CIRCUMSTANCE IN WHICH IT IS NECESSARY TO APPLY FILTERS PRIOR TO APPLY MODELS IS IF YOU'RE WORKING WITH FAILURE COUNTS DATA FOR WHICH SOME OF THE VALUES ARE NOT WHOLE NUMBERS. IN THIS CASE, YOU MUST APPLY THE ROUNDING FILTER BEFORE YOU CAN APPLY MODELS. IN GENERAL, FILTERS ARE USED TO MORE EASILY SPOT TRENDS IN THE DATA, OR TO REMOVE NOISE.** Guidelines for applying filters are given in paragraph 4.5.1.

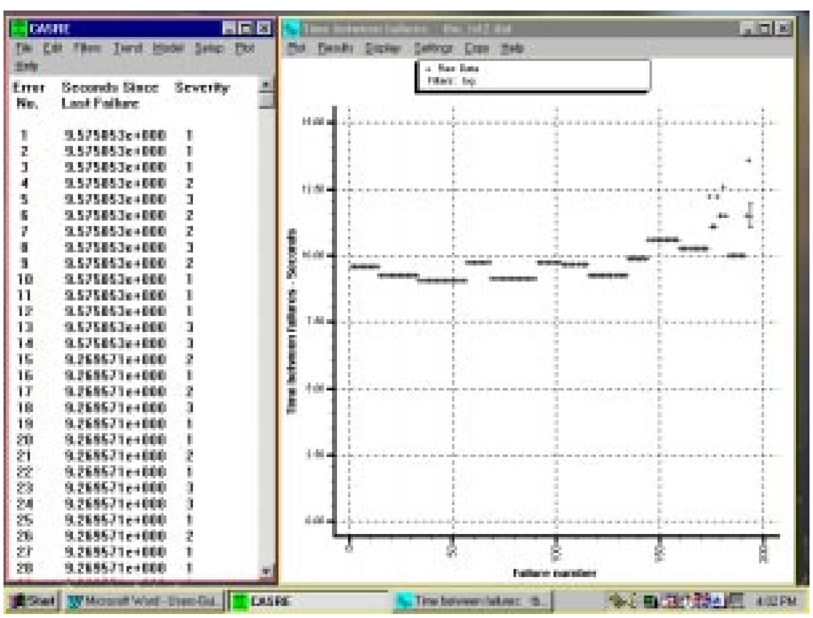

Figure 24 - Application of scaling and shifting filter (logarithmic filter)

After a filter has been applied, the filter(s) applied are shown in the small box near the top of the graphic display window. In Figure 24, the filter that was applied is indicated by the "Filters: log" text in that box. As many filters as desired can be pipelined; this is shown in Figure 25 below. Once more than about ten filters have been pipelined, there is not enough room in the plot's legend box to show the entire pipeline. In this case, only the most recently applied filters are listed in the plot's legend box.

Figure 25 illustrates the application of multiple filters. As with Figure 24, the filters are applied to the time between failures data first shown in Figure 4. The legend box at the top of the graphic display window reads "Filters: sever:1-3 | log | hann | hann", which means that the following sequence of filters was applied:

1. A subset of the failure data was created by looking only at failures having severity ratings between and including 1 and 3.
2. A logarithmic transform was applied to turn the time between failures curve into a nearly straight line.
3. A Hann window was applied to remove noise from the observations.
4. A second Hann window was applied to remove additional noise from the observations.

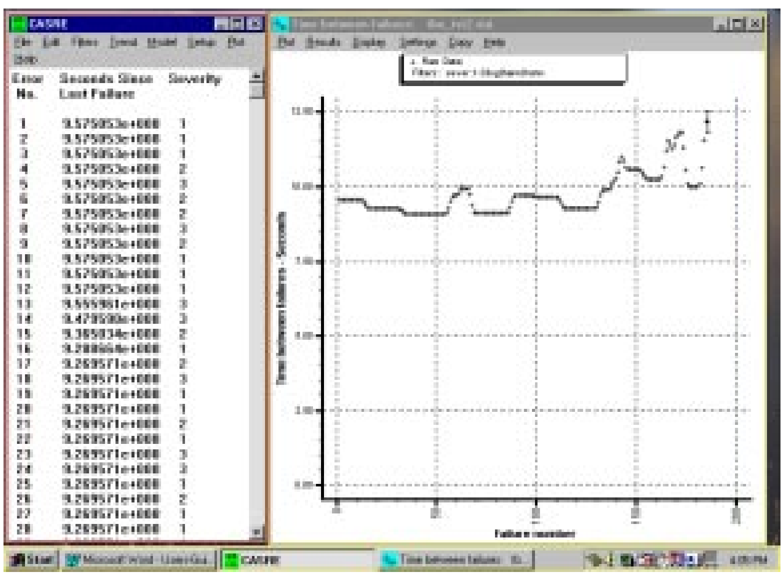

Figure 25 - Applying multiple filters to a set of failure data

The operation of the "Change time units..." filter is shown in Figures 26 and 27 on the following page. Once again, we'll be operating on the set of failure data shown in Figure 4. Notice that the times between failures are given in seconds. Suppose you want to change the time units to minutes. To do this, choose the "Change time units..." item in the main window's "Filters" menu to bring up the dialog box shown in Figure 26. The dialog box tells you what the current time units are. In this case, the units are "Seconds". To change the time units to minutes, select and click on the button that says "Minutes", then click on the "OK" button. This will change the time units to minutes. The times between failures shown in the main window are converted to minutes and redisplayed, and the graphic display window is changed. These changes are shown in Figure 27 on the next page. In the graphic display window, the label for the y-axis reads "Time between failures - Minutes" instead of "Time between failures - Seconds", and the values on the y-axis are obtained by dividing the values on the y-axis in Figure 4 by 60. Note that in Figure 27, the shape of the times between failures curve has not changed from that shown in Figure 4. In Figure 27, the last line of the graphic display window's legend box reads "Filters: time", which indicates that the "Change time units..." filter has been applied.

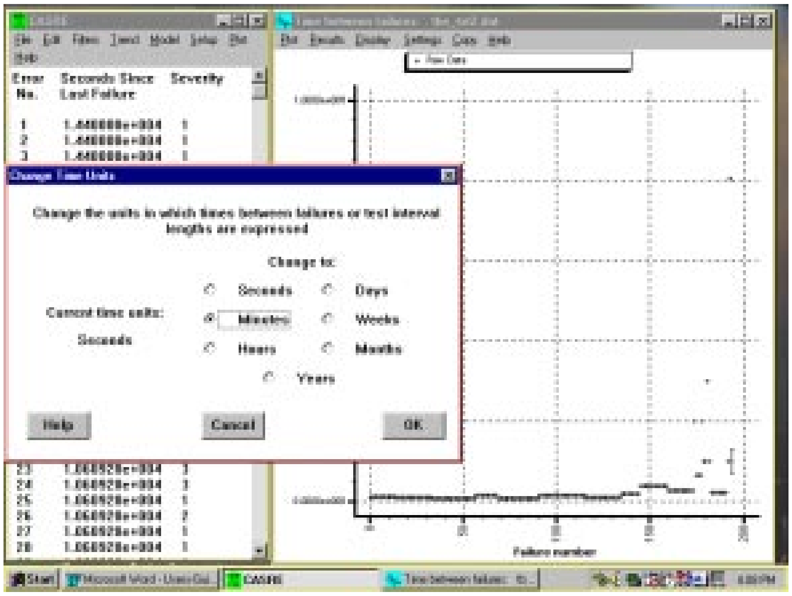

Figure 26 - Changing time units - choosing the new time units

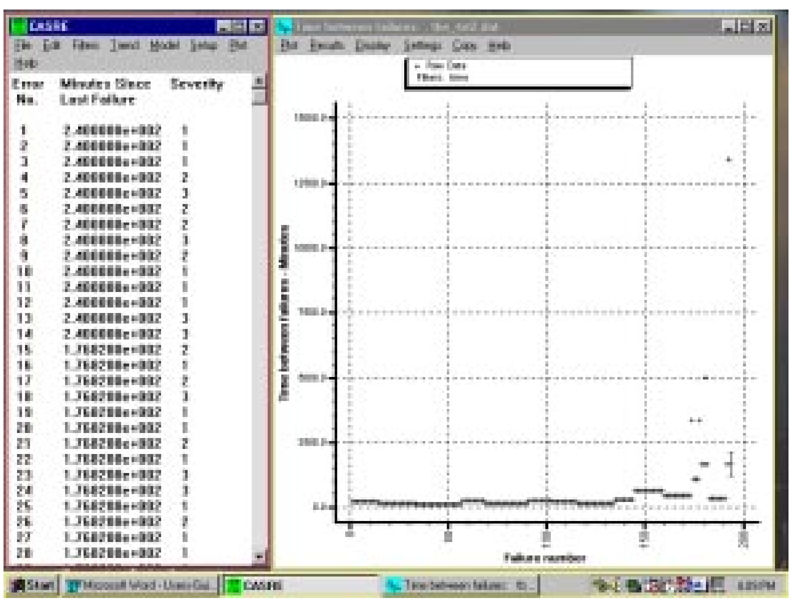

Figure 27 - Changing time units - conversion complete

4.5.1. Filter Application Guidelines

Guidelines for applying filters are given below.

1. The Hann window is applied to remove noise in the observed times between failures or failure counts. This filter should be applied only if a model has already been run using the unfiltered failure data as input, and the model's predictions could not be displayed in the graphic display window. Application of the Hann window can then be repeated until the first time that model predictions can be displayed. At that point, there should be no further applications of the Hann window to the failure data. Based on our experience during CASRE development, after more than about three applications, the Hann window will not appreciably change the shape of a curve drawn through the plotted raw data. We do not recommend applying it more than three times, then, since that would usually not result in any significant reduction of noise.

2. The shaping and scaling transformations can change the shape of the failure data curve, making it easier to make predictions about future failure behavior. For instance, the time between failures curves shown in the preceding figures can be transformed with a logarithmic transform, as shown in Figure 24. The "Cumulative failures" item in the graphic display window's "Display" menu can then be used to draw the cumulative failures curve using the transformed data. This curve, shown in Figure 28 below, is a line with a nearly constant slope. If the testing method remains the same, it can be expected that this straight-line trend will continue.

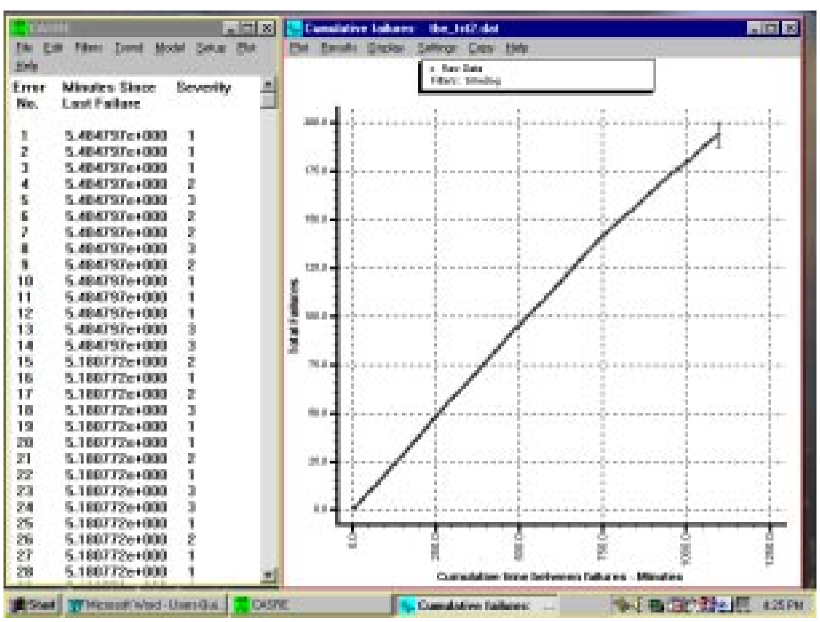

Figure 28 - Logarithmically transformed display of cumulative number of failures

3. The "Select severity" item allows a subset of the failure data to be formed based on the severity classifications of the observed failures. The default is for failures of all severity classifications to be displayed. To look at a subset of the failure data based on severity classifications, choose the "Select severity" option in the "Filters" menu. This will bring up the dialog box shown in Figure 29 below. Once you've entered the upper and lower severity limits and clicked the "OK" button in the dialog box, only the failure data between and including the chosen severity classifications will be displayed in the main window and the graphic display window. The resulting display is shown in Figure 30 on the next page. Times between failures and failure counts are automatically adjusted to have the correct values. This capability is used to model failure counts or times between failures separately for each severity classification.

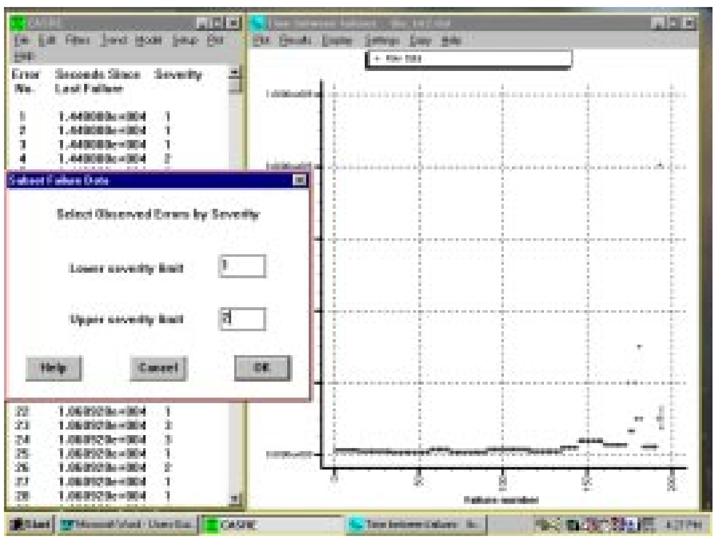

Figure 29 - Selecting a subset of failure data based on severity classification

The example in Figures 29 and 30 starts with the time between failures data shown in Figure 4. In Figure 29, we choose to look at only those failures having severity classifications of 1 or 2. Figure 30 shows the resulting subset of the failure data. In forming this type of subset, times between failures are to be adjusted to take into account those failures which do not appear in the subset. For instance, consider the following times between failures within a set of failure data:

Failure number	Time since last failure
n	100
n+1	150
n+2	262

If failures "n" and "n + 2" were of severity 1 or severity 2, while failure "n + 1" was not, failure "n + 1" would not be included in the subset, while failure "n + 2" would be relabelled as "n + 1" and given a time since last failure of 150 + 262 = 412.

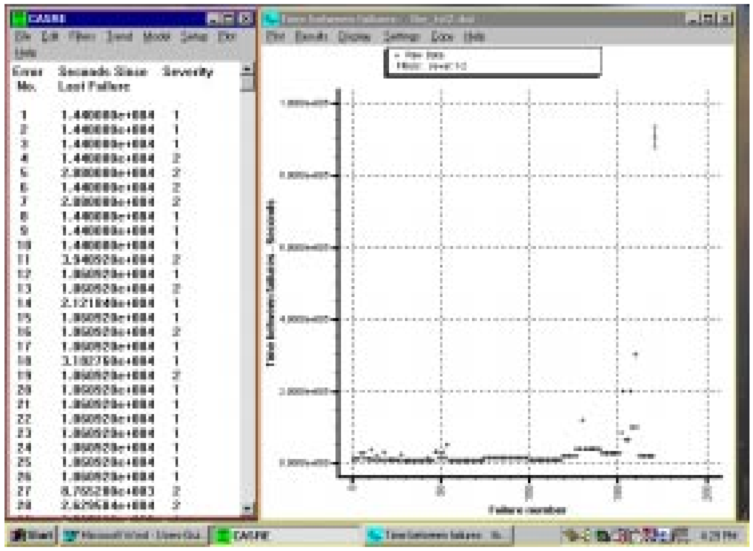

Figure 30 - Creating a subset of failure data based on severity classification - completion

> **For failure counts data, recall from paragraph 4.2 that if no failures have been observed during a test interval, CASRE shows a failure count of 0 and a severity of 0. When selecting a subset of severity categories, intervals having failure counts of 0 will be shown in addition to the intervals containing failures having the selected severities. This is done because when you start applying models, it is necessary to use those intervals having failure counts of 0 as well as those having no failure counts. Suppose you have a failure counts data set in which no failures were observed during interval 11. If you apply the "Select severity" filter, specifying severity levels from 2 through 7, you'll still see a failure count value of 0 for interval 11, as well as a severity of "N/A" for that interval after the filter has been applied.**

4. The "Round" filter rounds the failure data to the nearest whole number. There are several ways in which you may get failure data that has fractional values. For time between failures data, for instance, the times between successive failures may have been recorded as fractions of a second (day, hour, etc.). For failure counts data, you may have applied a filter such as a Hann window after opening the file, which may have changed the failure counts to fractional values.

   For time between failures data, this filter simply rounds the time between successive failures values to the nearest integer. For failure counts data, the failure counts for each severity class are rounded separately, then summed to yield the total number of failures in each test interval. You may want to use this filter in the following situations:

a. You've applied a series of filters to a set of failure counts data, and you want to apply one or more models to the data. If the failure counts aren't whole numbers, you won't be able to run any models. You can use this filter to turn the failure counts into whole numbers, after which you can apply the models you want.

b. You want to convert a set of failure counts data to time between failures data, but some or all of the failure counts aren't whole numbers. You can use this filter to turn the failure counts into whole numbers, then do the conversion.

You should note that for failure counts data, using the "Round" filter may change the data such that the total number of failures in the data set after application of the filter is not the same as the total number of failures in the original data set.

Note that it would be possible to look at those failures for only one severity classification. In the example of Figures 29 and 30, if we had wanted to look at only those failures of severity 1, we would have entered a "1" into both the "Lower severity limit" and "Upper severity limit" edit boxes in the dialog box shown in Figure 29.

In general, filters should not be applied to failure data unless a model has been run and the predicted time between failures or failure rate curves cannot be displayed. If this occurs, a Hann window or a scaling and shaping transformation can be applied to remove noise or change the shape of the failure rate curve, increasing the likelihood that model predictions for subsequent modeling runs will be displayed. The exception to this guideline is the "Filters" menu "Select severity" item, which allows a subset of the failure data to be formed based on the severity classification of the observed failures.

4.5.2. Removing Filters

Once filters have been applied to a failure data set, there are two ways of removing their effects. The filter most recently applied to the failure data may be removed by selecting the "Filters" menu's "Remove last filter" item. This will remove the effects of the most recent filter and redisplay the failure data as it was prior to the application of that filter. To remove all of the filters that have been applied to the failure data, choose the "Remove all filters" option of the "Filters" menu. This will redisplay the failure data as it was before the application of any filters. The only way to remove a filter in the middle of a chain of filters, such as that illustrated in Figure 25, is to first remove all filters and then reapply all of the removed filters except for the one that is no longer wanted.

4.6. Identifying Trends in Failure Data

Before applying any models to a set of failure data, it is advisable to determine whether the failure data does, in fact, exhibit reliability growth. If a set of failure data does not exhibit increasing reliability as testing progresses, there is no point in attempting to estimate and forecast the system's reliability. CASRE provides two types of trend tests that can be applied to both time between failures data and failure counts data. These are the running arithmetic average and the Laplace test.

4.6.1. Running Arithmetic Average of Time Between Failures/Failure Counts

The running arithmetic average is one of the simplest trend tests that can be applied to determine whether a set of failure data exhibits reliability growth. This test may be applied to both time between failures data and failure counts data. **For failure counts data, the test may only be applied to data in which the test intervals are of equal length**. To apply the test, simply select the "Running average" item in the main window's "Trend" menu. The test is applied to the data between the first data point and last data point, as specified by the dialog box that's brought up by selecting the "Select data range" item in the main window's "Model" menu. The results of the test are shown in the main display window as well as the graphic display window. Running arithmetic average test results are interpreted as follows:

- For time between failures data, if the running arithmetic average increases as the failure number increases, the time between failures is increasing. Hence the system's reliability is increasing. Reliability models may be applied if the running arithmetic average is increasing. Conversely, if the running arithmetic average decreases with increasing failure number, the average time between failures is decreasing, meaning that the system is becoming less reliable as testing progresses.

- For failure counts data, if the running arithmetic average decreases as the test interval number increases, the number of failures observed per test interval is decreasing. Hence the system's reliability is increasing. For failure counts data, reliability models may be applied if the arithmetic average is decreasing with increasing test interval number. Conversely, if the running arithmetic average increases with increasing test interval number, the average number of failures observed per test interval is increasing, meaning that the system is becoming less reliable as testing progresses.

For time between failures data, the running arithmetic average after the ***i^th^*** failure has been observed, ***r(i)***, is given by:

$$r(i) = \sum_{j=1}^{i} \theta_j \Big/ i \,,$$

where θ_j is the observed time between the ***(j-1)^st^*** and the ***j^th^*** failures. For failure counts data, the running arithmetic average after the ***i^th^*** test interval has been completed, ***r(i)***, is given by:

$$r(i) = \sum_{j=1}^{i} n_j \Big/ i \,,$$

where n_j is the number of failures that have been observed in the j^{th} test interval. Figures 31 and 32 show the results of applying the running arithmetic average to two sets of failure data. The data set shown in Figure 31 is a time between failures data set, while that shown in Figure 32 is a failure counts data set.

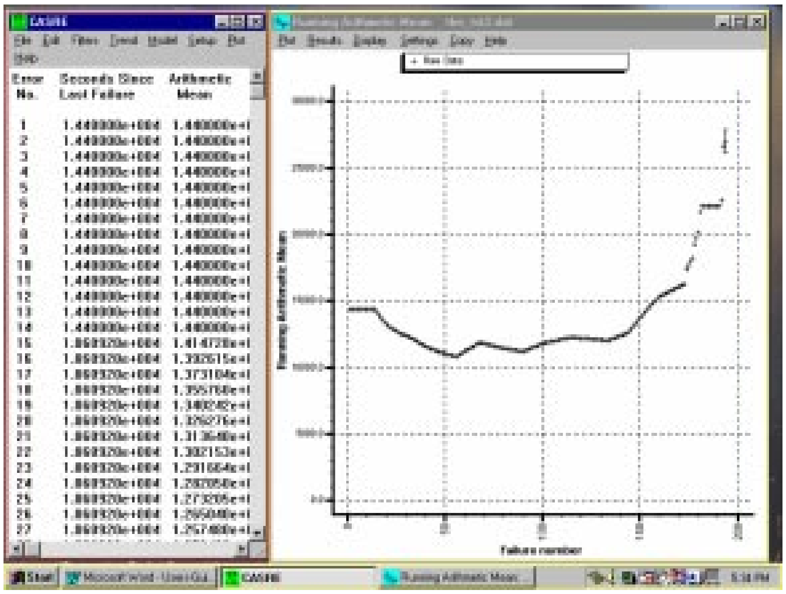

Figure 31 - Running Arithmetic Average - Time Between Failures Data

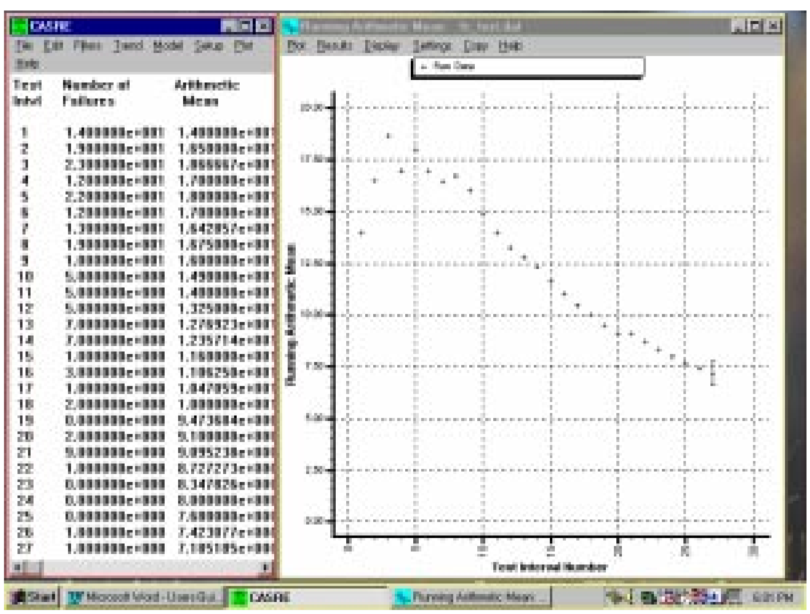

Figure 32 - Running Arithmetic Average - Failure Counts Data

The running arithmetic average in Figure 31 indicates that the average time between failures is decreasing between the start of the testing activity and approximately failure number 50. Between failures 50 and 130, the running arithmetic average time between failures is roughly constant –the reliability of the system does not appear to be either increasing or decreasing. After about the 130th failure, the running arithmetic average time between failures is increasing, indicating that the reliability of the system is increasing. Applying the criteria given above, reliability models may be applied in the interval starting at the 130th failure and continuing through the end of the data set.

For the failure counts data set shown in Figure 32, the running arithmetic average of the number of failures observed per test interval shows a steady decline after the fourth test interval, with the exception of the 8th interval. Reliability models could be applied to this data set starting with the 5th test interval, and continuing through the end of the data set.

4.6.2. Laplace Test

As with the running arithmetic average, the Laplace test may applied to both time between failures data and failure counts data. **For failure counts data, the test may only be applied to data in which the test intervals are of equal length**. To apply the test, simply select the "Laplace test" item in the main window's "Trend" menu. The test is applied to the data between the first data point and last data point, as specified by the dialog box that's brought up by selecting the "Select data range" item in the main window's "Model" menu. The test results are shown in the main display window and the graphic display window. Laplace test results are interpreted as follows:

- Reject the null hypothesis that occurrences of failures follow a Homogeneous Poisson Process (a Poisson process in which the rate remains unchanged over time) in favor of the hypothesis of reliability growth at the α% significance level if the test statistic is less than or equal to the value at which the cumulative distribution function for the normal distribution is $\alpha/100$. For example, if α is set to 5%, the value of the cumulative normal distribution function is approximately –2. If the value of the Laplace test were to be –2 for a set of failure data, then we could reject the null hypothesis of occurrences of failures following a Homogeneous Poisson Process at the 5% significance level.
- Reject the null hypothesis that occurrences of failures follow a Homogeneous Poisson Process in favor of the hypothesis of reliability decrease at the α% significance level if the test statistic is greater than or equal to the value at which the Cumulative Distribution Function (CDF) for the normal distribution is $(1-\alpha)/100$.
- Reject the null hypothesis that there is either reliability growth or reliability decrease in favor of the hypothesis that there is no trend at the α% significance level if the statistic is between the values at which the CDF for the normal distribution is $\alpha/200$ and $(1-\alpha/2)/100$. If α = 5%, the Laplace test statistic must lie between –2 and 2 for this hypothesis to be accepted.

For time between failures data, the value of the Laplace test statistic after the i^{th} failure has been observed, $u(i)$, is given by the following equation,

$$u(i)=\frac{\frac{1}{i-1}\sum_{n=1}^{i-1}\sum_{j=1}^{n}\theta_j-\frac{\sum_{j=1}^{i}\theta_j}{2}}{\sum_{j=1}^{i}\theta_j\sqrt{\frac{1}{12(i-1)}}},$$

where θ_j is the elapsed testing time between the ***j***th and ***(j-1)***st failures. For failure counts data, the Laplace test statistic is given by:

$$u(k)=\frac{\sum_{i=1}^{k}(i-1)n(i)-\frac{(k-1)}{2}\sum_{i=1}^{k}n(i)}{\sqrt{\frac{k^2-1}{12}\sum_{i=1}^{k}n(i)}},$$

where

- The entire testing period, represented by the interval ***[0,T]*** has been divided into ***k*** equal-length time units.
- The number of failures observed during the ***i***th time unit is ***n(i)***.

Figures 33 and 34 show the results of applying the Laplace test to two sets of failure data. The data set shown in Figure 33 is a time between failures data set, while that shown in Figure 34 is a failure counts data set.

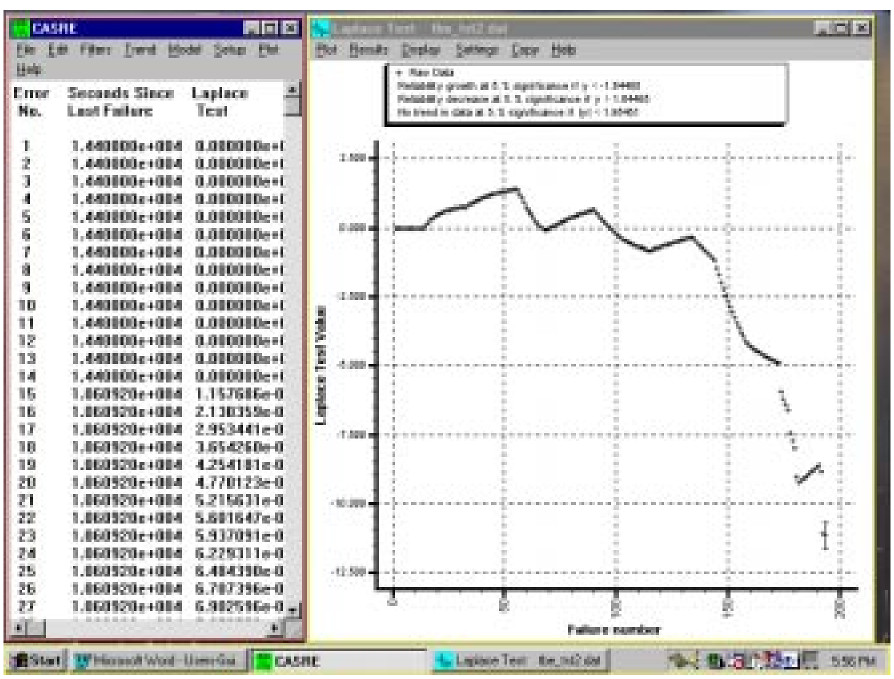

Figure 33 - Laplace Test Applied to Time Between Failures data

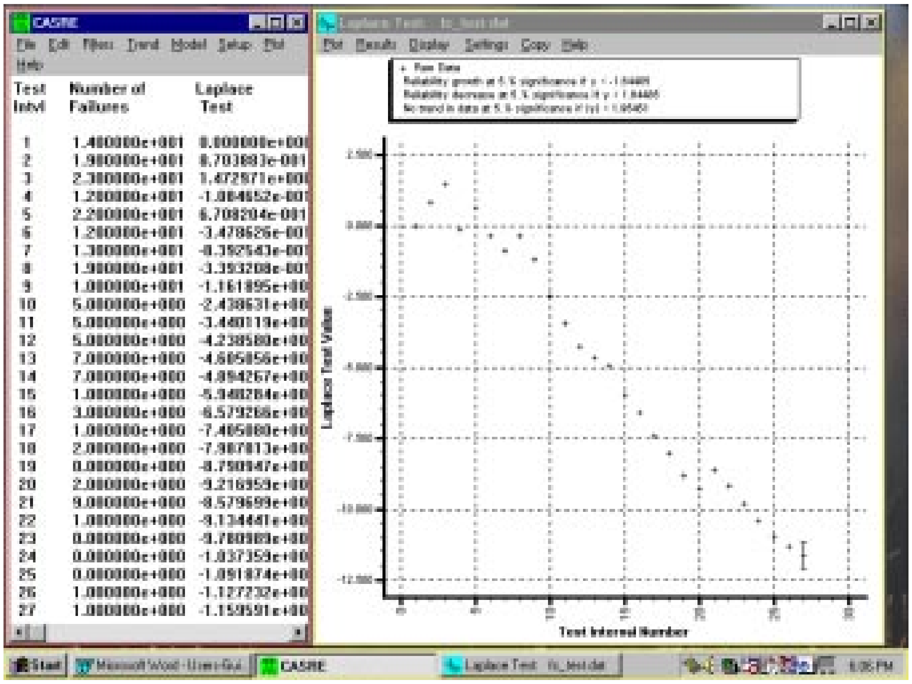

Figure 34 - Laplace Test Applied to Failure Counts Data

We see in Figure 33 that the system does start exhibiting reliability growth at the 5% significance level until about the 145th failure, at which point the Laplace test statistic assumes a value of –2. Reliability models could be applied to a subset of the data, starting at the 145th observed failure and continuing through the end of the data set. For the failure counts data shown in Figure 34, the value of the Laplace test statistic assumes a value of –2 after the 9th test interval, at which point the null hypothesis of occurrences of failures can be rejected at a 5% significance level in favor of the hypothesis that the system's reliability is increasing. For this data set, reliability models could be applied to a subset of the data, starting with the 10th test interval and ending with the 27th (last) observation in the data set.

4.6.3. Restoring the Normal Data Display

After applying trend tests, the normal display of failure data is restored by selecting the "Undo trend test" item in the main display window's "Trend" menu.

4.7. General Information on Models

Before showing you how to select and apply models to the failures data, we'll first discuss software reliability modeling in a fairly general way. In the following paragraphs, you'll see some of the high points of software reliability models. You'll also see the specific form of two particular software reliability models to illustrate some of the points in the general discussion.

4.7.1. Software Reliability Models in General

Software reliability models are statistical models which can be used to make predictions about a software system's failure rate, given the failure history of the system. The models make assumptions about the fault discovery and removal process. These assumptions determine the form of the model and the meaning of the model's parameters. There are two types of models - those that predict times between failures, and those that predict the number of failures that will be found in future test intervals.

Models that predict times between failures can be expressed as a probability density function, $\tilde{f}_i(t)$, whose parameters are estim-ated based on the values of previously observed times between failures t_1, t_2, …, t_{i-1}. This probability density function is used to predict the time to the next failure as well as the reliability of the software system. Suppose that we've observed **i-1** times between failures since the start of testing, and we want to predict the time between failure **i-1** and failure **i**, which we'll represent by the random variable **t**. The expected value of **t** is what we're looking for, and this is given by:

$$E[t] = \int_0^\infty t\, \tilde{f}_i(t)dt$$

where $\tilde{f}_i(t)$ is the probability density function representing the particular model we're using. The parameters of this probability density function can be estimated using either the **Maximum Likelihood** or **Least Squares** methods. See paragraph 4.8.1 and Appendix B for more details.

Since $\tilde{f}_i(t)$ is used to predict the time to the next failure, we can use it to predict the reliability of the system. According to $\tilde{f}(t)$, the probability that the time to the next failure will be less than a certain value, **x**, is simply:

$$P(t \le x) = \int_0^x \tilde{f}_i(t)dt$$

Recall that software reliability is defined as the probability that a software system will run without failure for a specified time in a specified environment. Using this definition, then, the reliability of the software over an interval of time of length **x** is **1 - P(t ≤ x)**, or:

$$Rel(x) = P(t > x) = \int_x^\infty \tilde{f}_i(t)dt$$

For models that predict the number of failures in future test intervals, we also have a probability density function $\tilde{f}_i(t)$. The parameters of $\tilde{f}_i(t)$ are computed based on the failure counts in the previous (i-1) test intervals. Suppose that we've observed failure counts in test intervals f_1, f_2, …, f_i, and we want to predict what the number of failures will be in interval i+1. Representing this quantity by the random variable **x**, we'll get our prediction by finding the expected value of **x**:

$$E[x] = \int_0^\infty x\,\tilde{f}_i(x)dx$$

4.7.2. The Jelinski-Moranda Model

For our time between failures example, we'll use the Jelinski-Moranda Model. This model makes the following assumptions about the fault detection correction process:

a. The rate of fault detection is proportional to the current fault content of the program.
b. All failures are equally likely to occur and are independent of each other.
c. Each failure is of the same order of severity as any other failure.
d. The failure rate remains constant over the interval between failure occurrences.
e. During test, the software is operated in a similar manner as the expected operational usage.
f. The faults are corrected instantaneously without introduction of new faults into the program.

From assumptions a, b, d, and f, we can write the **hazard rate** (the instantaneous failure rate) as:

$$z(t) = \phi(N - (i - 1))$$

where t is any time between the discovery of the i'th and (i-1)th failure. ϕ is the proportionality constant given in assumption (a), and N is the total number of faults initially in the system. This means that if (i-1) faults has been discovered by time t, there would be N-(i-1) faults remaining in the system. If we represent the time between the i'th and the (i-1)th failure by the random variable $\mathbf{X_i}$, from assumption (d) we can see that $\mathbf{X_i}$ has an exponential distribution, $\mathbf{f(X_i)}$, as shown below:

$$f(X_i) = \phi(N - (i - 1))e^{-\phi(N-(i-1))X_i}$$

Using assumption (b), the joint density of all of the $\mathbf{X_i}$'s is:

$$L(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} f(X_i) = \prod_{i=1}^{n}\phi(N - (i - 1))e^{-\phi X_i(N-(i-1))}$$

This is the **Likelihood function**, which we can use to compute estimates for the parameters ϕ and N. To make the computation easier, we can take the natural logarithm of the likelihood function to produce the **log-likelihood function**. After doing this, we then take the partial derivative of the log-likelihood function with respect to each of the two parameters, giving us two equations in two unknowns. Setting these equations to zero and then solving them gives us estimated values, $\hat{\phi}$ and $\hat{N}$, for the model parameters N and Φ:

$$\hat{\phi} = \frac{n}{\hat{N}\left(\sum_{i=1}^{n} X_i\right) - \sum_{i=1}^{n}(i - 1)X_i}$$

We have to find the value of $\hat{N}$ numerically from the following equation, and then use it to solve the previous equation for $\hat{\phi}$:

$$\sum_{i=1}^{n} \frac{1}{\hat{N} - (i-1)} = \frac{n}{\hat{N} - \frac{1}{\sum_{i=1}^{n} X_i} \left(\sum_{i=1}^{n} (i-1) X_i \right)}$$

If we've observed **i** failures, the predicted time to the next failure, $\hat{M}TBF$, is given by:

$$\hat{M}TBF = \frac{1}{z(t)} = \frac{1}{\hat{\phi}(\hat{N} - i)}$$

We can also derive a set of least-squares estimators for the model parameters. In this case, we're trying to minimize the following quantity, which is the sum of the squared differences between the observed $\mathbf{X_i}$ and their mean values, the **MTBF**s:

$$\sum_{i=1}^{n} (X_i - MTBF_i)^2 = \sum_{i=1}^{n} \left(X_i - \frac{1}{\phi(N - (i-1))} \right)$$

As with maximum likelihood estimation, we take the partial derivatives of the equation with respect to each of the model parameters, yielding the two following equations that we then set to 0 and solve:

$$\hat{\phi} = \frac{\sum_{i=1}^{n} \frac{1}{(\hat{N} - i + 1)^2}}{\sum_{i=1}^{n} \frac{X_i}{\hat{N} - i + 1}}$$

and

$$\left(\sum_{i=1}^{n} \frac{X_i}{(\hat{N} - i + 1)^2} \right) \bullet \left(\sum_{i=1}^{n} \frac{1}{(\hat{N} - i + 1)^2} \right) =$$

$$\left(\sum_{i=1}^{n} \frac{X_i}{\hat{N} - i + 1} \right) \bullet \left(\sum_{i=1}^{n} \frac{1}{(\hat{N} - i + 1)^3} \right)$$

The resulting estimate for **MTBF**, $\hat{M}TBF$, has the same form as that obtained from maximum likelihood estimation.

4.7.3. The Non-Homogeneous Poisson Process Model

For our failure counts example, we'll look at the Non-Homogeneous Poisson Process model. This model makes the following assumptions about the fault detection and correction process:

a. During test, the software is operated in a similar manner as the anticipated operational usage.

b. The number of failures, (f_1, f_2, …, f_n) detected in each of the time intervals $[(0,t_1),(t_1,t_2),\ldots,(t_{n-1},t_n)]$ are independent for any finite collection of times $t_1<t_2<\ldots<t_n$.

c. Every fault has the same chance of being detected and is of the same severity as any other fault.

d. The cumulative number of faults detected at any time t, N(t), follows a Poisson distribution with mean m(t). This mean is such that the expected number of failure occurrences for any time (t, t+Δt) is proportional to the expected number of undetected faults at time t.

e. The expected cumulative number of faults function, m(t), is assumed to be a bounded, non-decreasing function of t with:

$$m(t) = 0 \quad t = 0$$
$$m(t) = a \quad t = \infty$$

where **a** is the expected total number of faults to be eventually detected in the testing process.

f. Errors are removed from the software without inserting new errors.

From assumptions (d) and (e), for any time period (t,t+Δt), **m(t, t+Δt) - m(t) = b(a-m(t)) Δt + O(Δt)** where b is the constant of proportionality and **O(Δt)/ Δt** = 0 as **Δt** → 0. As **Δt** → 0, the mean function m(t) satistfies the following differential equation:

$$\frac{dm(t)}{dt} = ab - bm(t)$$

Under the initial condition m(0) = 0, the mean function is:

$$m(t) = a(1 - e^{-bt})$$

From assumption d, the probability that the cumulative number of failures, N(t), is less than n is:

$$P(N(t) \leq n) = \frac{(m(t))^n e^{-m(t)}}{n!}$$

At this point, we can compute estimates for the model parameters, a and b, using the maximum likelihood or least squares method. For $f_i = N(t_i)-N(t_{i-1})$ and the failure counts being independent, the likelihood function is:

$$L(f_1, f_2, \ldots, f_n) = \prod_{i=1}^{n} \frac{(m(t_i) - m(t_{i-1}))^{f_i} e^{m(t_{i-1})-m(t_i)}}{f_i!}$$

Taking the natural log of this equation to form the log-likelihood function, and then taking the partial derivative of the log-likelihood function with respect to each of the model parameters gives us the following two equations which, when solved, will yield estimated values for the parameters:

$$\hat{a} = \frac{\sum_{i=1}^{n} f_i}{1 - e^{-\hat{b}t_n}}$$

The estimated value of the proportionality constant, **b**, must be determined numerically from the following equation:

$$\frac{t_n e^{-\hat{b}t_n} \sum_{i=1}^{n} f_i}{1 - e^{-\hat{b}t_n}} = \sum_{i=1}^{n} \frac{f_i (t_i e^{-\hat{b}t_i} - t_{i-1} e^{-\hat{b}t_{i-1}})}{e^{-\hat{b}t_{i-1}} - e^{-\hat{b}t_i}}$$

The estimated value of **b** is then substituted into the equation used to find the estimated value of **a**.

Once we've found the estimated values for the model parameters, we can compute the number of failures expected to be observed in the (i+1)th interval, given that we've observed failures in intervals 1 through i. This value is given by:

$$\hat{m}(t_{i+1}) - \hat{m}(t_i) = \hat{a}(e^{-\hat{b}t_i} - e^{-\hat{b}t_{i+1}})$$

We can also compute least-squares estimators for the parameters **a** and **b**. In doing so, we want to minimize the following quantity:

$$S = \sum_{i=1}^{n} (f_i - (m(t_i) - m(t_{i-1})))^2$$

Differentiating with respect to the two model parameters and setting the resulting equations to 0 gives us the following two equations which, when solved, provide estimated values for the model parameters:

$$\hat{a} = \frac{\sum_{i=1}^{n} f_i \left(e^{-\hat{b}t_{i-1}} - e^{-\hat{b}t_i} \right)}{\left(e^{-\hat{b}t_{i-1}} - e^{-\hat{b}t_i} \right)^2}$$

and

$$\sum_{i=1}^{n} f_i \left(t_i e^{-\hat{b}t_i} - t_{i-1} e^{-\hat{b}t_{i-1}} \right) =$$

$$\hat{a} \sum_{i=1}^{n} \left[\left(e^{-\hat{b}t_i} - e^{-\hat{b}t_{i-1}} \right) \bullet \left(t_i e^{-\hat{b}t_i} - t_{i-1} e^{-\hat{b}t_{i-1}} \right) \right]$$

4.8. Model Setup and Application

Once a failure data file has been opened, one or more reliability models can be applied to it. If changes have been made to the data using an external application, or if the data has been filtered, the selected software reliability model(s) will use the edited and filtered data as their input. For the examples presented below, the input data will be unfiltered. The data we'll use for the examples is that shown in Figures 4 and 5.

Preparatory to choosing and running models, there are three model controls that can be set. These are:

1. The type of parameter estimation method to use.

2. The range of failure data over which the models will be applied.

3. The number of steps into the future for which the models should make predictions.

Each of these is briefly discussed below. Note that it is not necessary to set these controls prior to selecting and applying one or more models, since each control has a default setting.

4.8.1. Parameter Estimation

The two methods of parameter estimation are maximum-likelihood (ML) and least-squares (LS). The default mechanism is ML. These methods are summarized in the Glossary. If you use ML, you can compute model applicability statistics that are not available if you use LS estimation. See the table on page 98 for a summary. However, ML may not always produce parameter estimates, and takes more computation time than LS. LS usually requires less computation time to estimate parameters, and always finds a parameter estimate. However, if the LS parameter estimates result in meaningless model predictions (e.g. times between failures of less than 0), CASRE will not allow the results to be displayed. For most of the data sets we have seen, ML will produce valid parameter estimates and is sufficiently quick to be the preferred estimation method. Usually, you'll want to apply the least-squares method only after using ML and finding that the ML estimates aren't valid.

To change the parameter estimation method, select the "Model" menu's "Parameter estimation" item, which will bring up a dialog box having a radio button beside the name of the most-recently chosen estimation method. Simply select and click the radio button beside the name of the desired estimation method, then select and click the "OK" button. Until changed again, the chosen parameter estimation method will be used for all subsequent modeling runs.

When you choose ML estimation, every model that you choose to run uses the ML estimation method. If you choose LS estimation, **only those models that can** use the LS method will be run using LS estimation - the remaining models use ML. This is because not all models in the SMERFS modeling library, which form the core of CASRE's modeling capabilities, have implemented the LS estimation method. The following table provides a summary; each cell lists the parameter estimation method actually used regardless of the method selected.

Model Name \ Selected Estimation Method	Maximum Likelihood	Least Squares
Geometric	ML	LS
Jelinski-Moranda	ML	LS
Littlewood-Verrall Linear	ML	LS
Littlewood-Verrall Quadratic	ML	LS
Musa Basic	ML	ML
Musa-Okumoto	ML	ML
Nonhomogeneous Poisson Process (for time between failures)	ML	ML
Generalized Poisson	ML	LS
Generalized Poisson – user selected interval weights	ML	LS
Shick-Wolverton	ML	LS
Nonhomogeneous Poisson process (for failure counts)	ML	LS
Schneidewind	ML	ML
Schneidewind – Ignore first "s" intervals	ML	ML
Schneidewind – Total failures in first "s" intervals	ML	ML
Yamada S-Shaped	ML	ML

Table 2 – Parameter Estimation Methods

In practical terms, this means that if you were to select the ML method and then run the Musa Basic, Musa-Okumoto, and one of the Littlewood-Verrall models, you'd be able to see all of the model evaluation statistics that CASRE is capable of displaying. However, if you were to select the LS estimation method and then run the same three models, you'd only be able to see the model evaluation statistics available for the LS method, regardless of the fact that two of the models were actually run using ML.

4.8.2. Selecting a Modeling Data Range

Reliability models may be applied over some or all of the failure data. In addition, there is also a range at the start of the failure data over which initial parameter estimates are made. After this range, CASRE makes parameter estimates for each observation in the failure data set. The default range over which the models will operate is the smaller of the most recent 1000 observations or the entire data set. We also need to make initial estimates of the model parameters. To do this, we use the first half of the default data range.

Suppose that you have a data set which consists of 2000 observed times between failures. The default range over which models would be applied would be observations 1001-2000, while the initial parameter estimates for the models would be from observations 1001-1500. The default settings can be changed by selecting the "Select data range" item in the "Model" menu. Selecting this item brings up the dialog box shown in Figure 35 below. The data range over which the models operate is defined by the "First data point" and "Last data point" edit windows. Initial parameter estimates are made using the "First data point" through "Parameter estimation end point" observations. For each observation "i" after the parameter estimation end point, parameter estimates are made using observations from the "First data point" through "i". The first data point must be less than the last data point, while the parameter estimation end point must be greater than or equal to the first data point and less than the last data point.

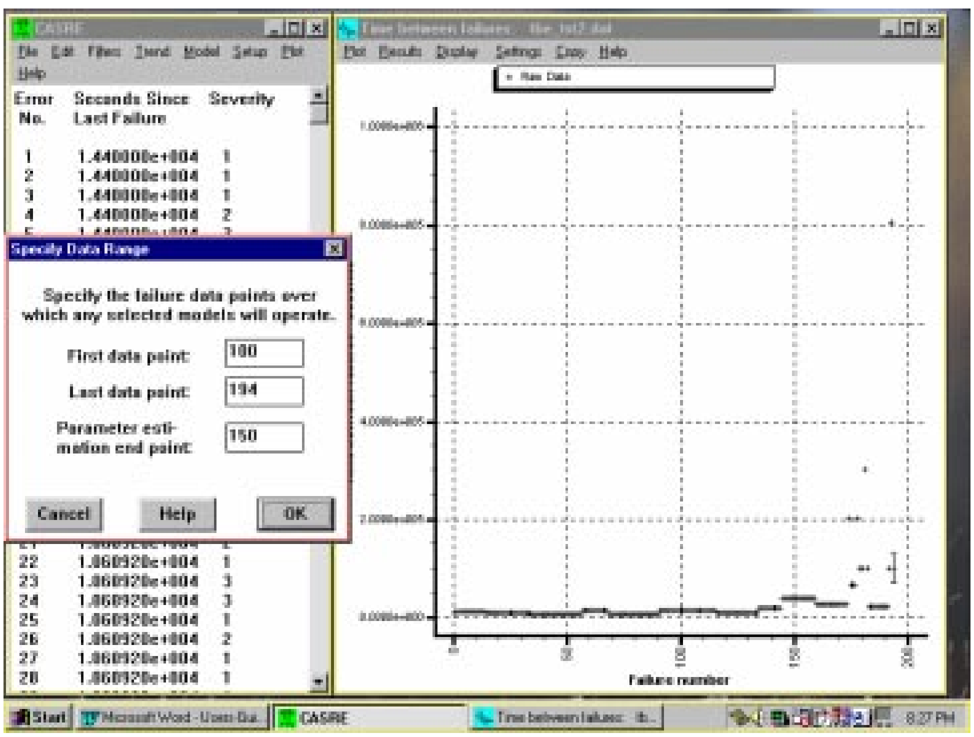

Figure 35 - Selecting a new modeling data range

There may be reasons for which the modeling data range should be changed from the default values. It may be the case that from observations 1-1300 in our 2000-observation data set, white-box testing methods were employed, while from observations 1301-2000, black-box methods were employed. To more accurately predict the reliability for the period of black-box testing, you would choose a modeling range of 1301-2000 and have an initial parameter estimation range of 1301-1500. It could also be the case that the amount of testing resources changed or that different testing equipment became available. It could also be the case that different portions of the system were tested at different times. Suppose, for instance, that our software system was composed of three components. Furthermore, suppose that the first 800 failures reflected the testing of that component, the second 600 failures reflected testing of the second component, and that the last 600 failures reflected testing of the third component. This would show up in the graphics display window as a time between failures plot having a sawtooth shape. You would get more accurate results by modeling the reliability of each component separately rather than just using the default range (1001-2000). This would mean applying models to the following data ranges: 1-800, 801-1400, 1401-2000.

Let's use the failure data shown in Figure 4 as an example. This data set has 194 times between failures. Applying the rules we've given, the default data range is observations 1-194, while the default parameter estimation range is observations 1-97. Suppose we want to change the data range to be from points 100-194, with the initial parameter estimation being from points 100-150. Choose the "Select data range..." item in the main window's "Model" menu. The dialog box shown in Figure 35 will appear. Enter the new start point, the new end point, and the new end point for initial parameter estimation as shown in Figure 35. Complete the operation by selecting and clicking on the "OK" button.

4.8.3. Predicting into the Future

Finally, the number of steps past the last data point in the modeling range for which predictions are to be made can be specified by selecting the "Model" menu's "Predictions" item. Selecting this item brings up a dialog box which prompts you for the following information:

1. The number of failures past the last data point in the modeling range for which times between failures should be predicted, if the failure data type is time between failures,

or

2. The number of test intervals past the last data point in the modeling range for which failure counts should be predicted, if the failure data type is failure counts.

Suppose you have a set of time between failures data which gives the times between failures for 194 errors, as shown in Figure 4. If you want to make estimates of times between failures for errors 195-214, you'll select the "Predictions" menu item, and enter a "20" into the edit window of the dialog box that appears. This is shown in Figure 36 below.

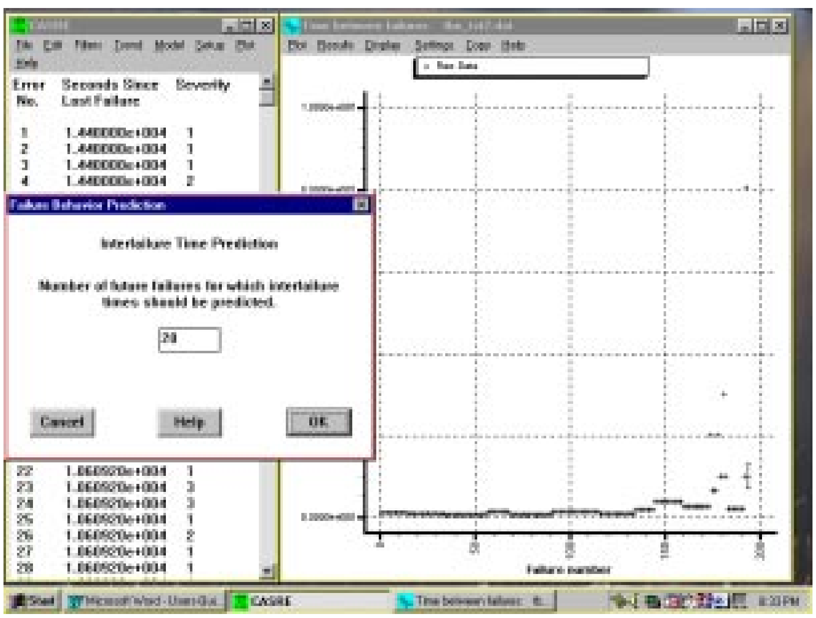

Figure 36 - Predicting future times between failures for time between failures data

The default number of steps ahead is one failure ahead for time between failures data, and one test interval ahead for failure count data. It is generally not a good idea to predict more than 20 failures or test intervals into the future, or 10% of the length of the failure data set, whichever is less, since the character of the testing might change during that time and thereby cause inaccurate long-term predictions. In practice, CASRE allows the selection of as many failures or test intervals ahead as desired, as long as you don't specify more failures or test intervals than can fit into a 64

Kbyte memory segment. If you do specify more future failures or test intervals than will fit into a 64 Kbyte segment, you'll get an error message telling you that there's not enough memory to hold the results, and asking you to specify fewer failures or test intervals.

For failure counts data, we also need to know what we expect the length of future test intervals to be before we can make predictions into the future. In the dialog box shown in Figure 37 below, you can enter this information in the following way:

1. Give a specific length for the test interval by entering a value into the edit window. To do this, you first have to click on the radio button labelled "User-defined length." After you've done this, you can change the value in the edit window labelled "How long do you expect test intervals after the last observation to be?"
2. Specify future test interval lengths as the average test interval length over all of the data. Click on the radio button labelled "Average of all test interval lengths."
3. Specify future test interval lengths as the average test interval length over the subset of the data you selected for modeling. Suppose, for instance, that the modeling data range for this data set was test intervals 5-27. If you think that test intervals 28, 29,..., will have lengths that are the average of the lengths of intervals 5-27, click on the radio button labelled "Average of interval lengths for modeling data range."
4. Specify future test interval lengths as the average test interval length over a specific set of observations. Again, suppose that the range over which models will be applied is intervals 5-27. However, you may think that the length of test intervals 28, 29, ..., will be better represented by the average of the lengths of intervals 15-27. You can specify this by first clicking on the radio button labelled "Average of interval lengths for a particular range:". After clicking this radio button, enter a "15" in the edit window labelled "First interval", and enter a "27" in the edit window labelled "Last interval".

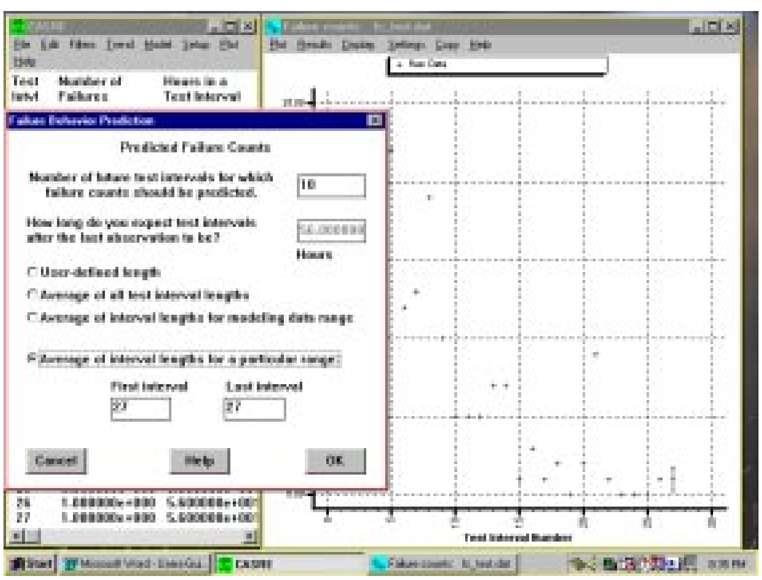

Figure 37 - Predicting future failure counts for failure count data

The default future test interval length is the length of the last test interval in the data set. After choosing the number of intervals past the last observation for which to make predictions, and after choosing the length of future intervals, click on the "OK" button to tell CASRE to use these values in making predictions with failure count models.

4.8.4. Selecting and Running Models

Now that the parameter estimation method, data range, and number steps into the future for which predictions should be made have been selected, we are ready to select and apply one or more models. Use the mouse to choose the "Select models..." item in the "Model" menu. This brings up the dialog box, shown in Figure 38 below, from which the models to be run are chosen.

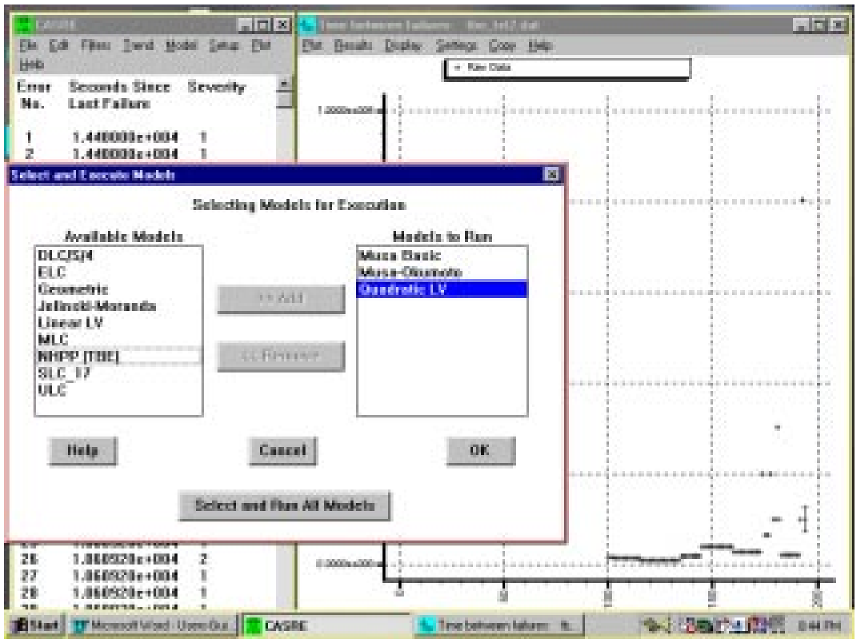

Figure 38 - Choosing models to run for time between failures data

The dialog box shown in Figure 38 has two lists. One of these, "Available Models", shows models that can be run using the failure data shown in the main window as input. The other list, "Models to Run", shows those models that have been selected to run. You can choose a model from the list of available models in one of the two following ways:

1. Use the mouse to highlight the name of a model in the "Available Models" list. Select and click on the "Add" button to place the model name in the list of models to run, and remove it from the list of available models.
2. Double-click on the name of the model in the "Available Models" list.

Similarly, a model name in the list of models to run may be removed by highlighting it and then selecting and clicking the "Remove" button, or by double-clicking on the name of the model in the "Models to Run" list. The model name then moves back into the list of available models. When model selection is complete, select and click on the "OK" button to start execution of the chosen models.

If you want to run all of the models, simply click the “Select and Run All Models” button.

Since CASRE uses two different types of failure data, the contents of the "Available Models" list will depend upon whether the failure data is time between failures or failure counts. The models available for each type of failure data are listed in Table 3 below.

Times Between Failures (TBF) Models	Failure Counts (FC) Models
Geometric*	Generalized Poisson*
Jelinski-Moranda*	Generalized Poisson – user-specified interval weighting*
Littlewood-Verrall Linear*	Nonhomogeneous Poisson(NHPP)
Littlewood-Verrall Quadratic*	Schneidewind*
Musa Basic*	Schneidewind – ignore first "s-1" test intervals*
Musa-Okumoto	Schneidewind – total failures in first "s-1" test intervals*
Nonhomogeneous Poisson(NHPP)*	Shick-Wolverton*
	Yamada S-shaped

Table 3 – Models for Each Type of Failure Data

For failure counts models, all of the failure counts in the data set you're using must be whole numbers. If they're not, CASRE will not let you run any of the failure counts models. If any of the failure counts in your data set are fractional values, you can use the "Round" filter (see paragraph 4.5.1) to make all of the failure counts into whole numbers.

The following models will produce discontinuous curves in the graphics display window:

- Musa Basic - for estimated MTBF
- NHPP (TBF data) - for estimated MTBF
- Generalized Poisson - for estimated failure counts

Detailed mathematical descriptions of the models marked with an asterisk, as well as explanations for the behavior of the Musa Basic and Nonhomogeneous Poisson Process (TBF data) models, may be found in the NavSWC technical report TR 82-171, "A Survey of Software Reliability Modeling and Estimation", prepared by Dr. William H. Farr [NSWC83]. If you want a copy of this report, write to the Naval Surface Warfare Center, Dahlgren, VA, 22448 and ask how you may get one. Mathematical descriptions of the remaining models may be found in Software Reliability - Measurement, Prediction, Application: by Musa, Iannino, and Okumoto, published by McGraw-Hill in 1987 [Musa87] and in Handbook of Software Reliability Engineering, edited by Michael Lyu, published by McGraw-Hill in 1996 [Lyu96].

In addition, there are four permanent combination models built into CASRE – these combinations are for time between failures data only. These combinations are:

- Dynamically-Weighted Linear Combination (DLC/S/4)
- Equally-Weighted Linear Combination (ELC)
- Median-Weighted Linear Combination (MLC)
- Unequally-Weighted Linear Combination (ULC)

Each of these combinations is made up of the following component models:

- Nonhomogeneous Poisson Process (NHPP)
- Musa-Okumoto (MO)
- Littlewood-Verrall Quadratic (LVQ)

The weightings for the combination models are as follows:
- ELC = (1/3)NHPP + (1/3)MO + (1/3)LVQ
- MLC = whichever model's prediction is between the other two.
- ULC = (1/6)Pessimistic + (4/6)Median + (1/6)Optimistic

where Pessimistic denotes the model whose predictions of time to next failure are the lowest, Optimistic denotes the model whose predictions of time to next failure are the largest, and Median denotes whose predictions are between the Optimistic and Pessimistic models.

The permanent combinations are in order of best performance to worst performance, based on our experience. There are no permanent combinations for failure count models because we have not had enough experience with them to warrant their inclusion in the CASRE configuration file.

Any combinations that have been defined by the user are also included in the "Available Models" list. There is one user-defined combination model, SLC_17, shown in the "Available Models" list for the "Select Models" dialog box shown in Figure 38. We will learn how to define our own combinations in paragraph 4.13.3.

Based on our experience, we recommend the following method for selecting models. If you're using time between failures data, select all of the individual time between failure models, as well as the four permanent combinations. Rank the models (details in paragraph 4.9.2), and choose the model with the highest ranking as the basis of your predictions. For failure count data, select all of the individual failure count models and rank them. If none of the failure count models provide a good fit, as discussed in paragraph 4.9.2, then convert the failure count data to times between failures data, apply all of the individual time between failure models (including the permanent combinations), and rank those results. Choose the highest-ranking model as the basis of your predictions.

The first time that the "Select models..." option is chosen, the list of models to run will be blank. After you select one or more models, run them, and then choose "Select models..." again, the list of models to run will contain the last set of models that were run for the type of failure data displayed in the CASRE main window. Suppose, for instance, that you're using time between failures data, and you've chosen the Geometric, Littlewood-Verrall, and Musa-Okumoto models. After running these models, you then open a data file containing failure counts. After selecting and running several failure count type models, a second file of times between failures is opened. When you select models, the Geometric, Littlewood-Verrall, and Musa-Okumoto will appear in the "Models to Run" list for the "Select Models" dialog box.

To start running the models identified in the "Models to Run" list, select and click the "Run Models" button. If no more information is required to run the models, a dialog box will appear in which the name of the currently-executing model and the execution progress for that model are displayed. For a failure data set containing "n" observations, there are six status messages that can be displayed in the dialog box. **For every model you run that's not a combination model, all of the messages are displayed in the following sequence:**

1. The message "XX of n", in which "XX" is either an error number or a test interval number, depending on failure data type. The number of observations is "n". "XX" starts at the point immediately after the initial parameter estimation end point and increases until its value is "n".

2. The message "Computing TBFs" for time between failures models, or "Computing failure counts" for failure count models, is the second message displayed. This message is displayed when the models are making predictions about future failure behavior. On a 486DX-based machine running at 25MHz or faster, this message will appear for only a very short period of time.

If the parameter estimation method is maximum likelihood, and provided that the model produced valid predictions during the time that message 1 was being displayed, the four following messages will appear. Each of these messages appears for roughly the amount of time that message 1 was displayed.

3. "Computing model bias". **This message appears for time between failures models only**, and indicates that the extent of bias in the model is being determined. This bias value can be shown in the bias plot or the bias scatter plot. Model bias and the plots associated with it are discussed in paragraph 4.9.2.

4. "Computing model noise". **This message appears for time between failures models only**, and indicates that the amount of noise introduced into the predictions by the model itself is being determined. Model noise is discussed in paragraph 4.9.2.

5. "Computing bias trend". **This message appears for time between failures models only**, and indicates that a search for trends in the model bias is in progress. Model bias trend is discussed in paragraph 4.9.2.

6. "Computing prequential likelihood". This message appears for both failure count and time between failures models, and is the last model displayed while running a model. Prequential likelihood is a statistic that is used to evaluate a model's applicability to a set of failure data (details in paragraph 4.9.2). This value can be shown in the prequential likelihood and relative accuracy plots, also discussed in paragraph 4.9.2.

These four messages do not appear if LS parameter estimation was chosen (see paragraph 4.8.1), nor do they appear if the model produced invalid estimates at some point while message 1 was displayed. If a model produced invalid estimates at some point while message 1 is displayed, it is not possible to perform the computations that would result in messages 3-6 being displayed.

If you're running one or more combination models, messages 1-6 will appear for each component of the combination that it is not a combination itself. For instance, when you're running the DLC/S/4 combination model, messages 1-6 will appear for the Littlewood-Verrall, Musa-Okumoto, and NHPP components of this model. While forming the combination of model results, the name of the combination (DLC/S/4 for this example) will appear in the modeling progress dialog box, as well as the following statement:

1. The message "XX of n", in which "XX" is either an error number or a test interval number, depending on failure data type. As above, "n" represents the number of observations. "XX" starts at the point immediately after the initial parameter estimation end point and increases until its value is "n" **plus the number of steps ahead for which you want to make predictions.**

This is the only message shown while forming the actual combinations, since the process of forming the combinations is very simple and takes very little time compared to what is required to run the component models.

The dialog box has a "Stop models" button which you can select and click at any time to stop model execution. If you click this button, model execution will stop, and only the results of those models that have been completely run will be displayable in the graphic display window. Note that if the "Stop models" button is clicked while messages 3-6 are being displayed, there may be a long pause (a few seconds to a few minutes, depending on the model, the size of the data set, and whether your computer has a math co-processor) before model execution stops. If this button is not clicked, all of the models you've selected are executed. Once all of the models have been run, the "modeling progress" dialog box disappears from the screen, and you'll be able to display the results of any of the models you've run in the graphic display window. More details on displaying model results are given in the next section.

If one or more of the models you've chosen requires additional control information before it can be run, the message box shown in Figure 39 below appears, telling you how many models require additional control settings. This message box appears after the "Run Models" or “Select and Run all Models” button in the "Select models" dialog box has been clicked. The models requiring additional information are all failure counts models:

- Generalized Poisson – user-specified interval weighting factor
- Schneidewind – ignore first “s” test intervals
- Schneidewind – total failures in first “s” test intervals

The dialog boxes used to gather the required information for these models are shown in Figures 40, 41, and 42 on the following pages. Only the dialog box for the Generalized Poisson model has default settings. In the Generalized Poisson model, you can specify a particular weighting factor "alpha" to apply to the test interval lengths. CASRE supplies a default value of “1.0” for alpha; for the way that CASRE is implemented, this choice of alpha makes the model equivalent to the Jelinski-Moranda model, but for failure count data. We recommend against changing the default value of alpha unless you are familiar with the model and are aware of the consequences that might arise as a result of such a change. Once again, [NSWC83] and [Lyu96] give detailed descriptions of this model, as does [Scha79].

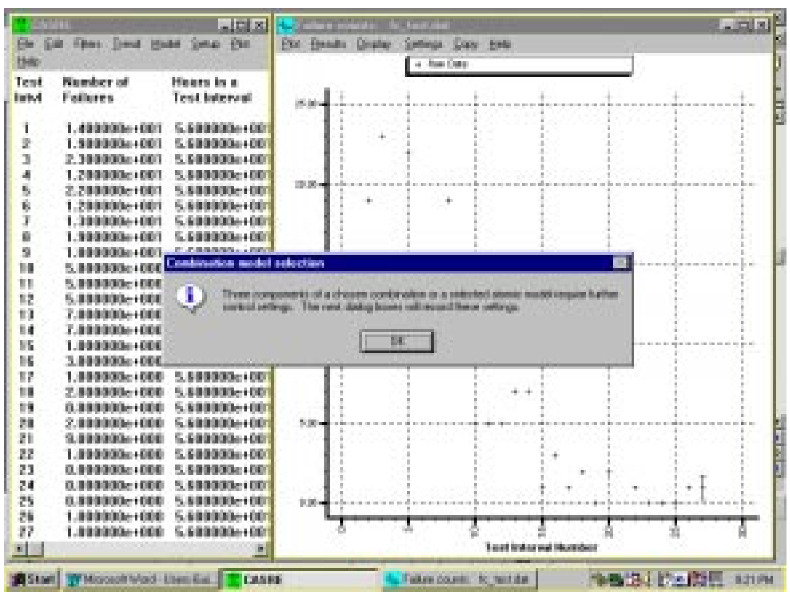

Figure 39 - Additional information required to run models message box

The Schneidewind model is somewhat different from other failure count models in the way it treats data. The model can treat the failure count data in one of three ways:

1. Use all of the failure counts for all the intervals in the modeling range you've selected. This is the default setting. Figure 31 gives an example of selecting a range of data for modeling.

2. For a failure count data set that spans test intervals "x" to "y", ignore completely the failure counts for intervals "x" through "s-1", where "s" is greater than "x" and less than "y+1". You would use this treatment if the test method or test environment has changed somewhere in the data range you've selected, if you expect the future test method and environment to be more closely related to the current method and environment, and you know the point at which it's changed, you may want to choose the second treatment type to get more accurate predictions. In this case, set “s” to be the point at which you believe that the test environment has changed - all the failure counts before the point you choose will be discarded. Figure 41 shows the dialog box for this treatment type. For example, let's suppose that you have a failure count data set that has 27 intervals, and you select a data range that starts at interval 5 and ends at interval 27. If you choose the Schneidewind model and the type of treatment of the data we're discussing now, you can choose a value of "s" between and including 6 and 27. Setting “s” to 10 would cause CASRE to ignore observations 6-9 when running the Schneidewind model; only observations 10-27 would be used in estimating and forecasting reliability. CASRE supplies a default value for “s” – the default value is halfway between the starting point of the data and the observation used as the end point of the initial parameter estimation. For example, if we have a data set consisting of 27 observations (1-27), and the observation used as end point of the initial parameter estimation is point 13, the default value of “s” would be 7.

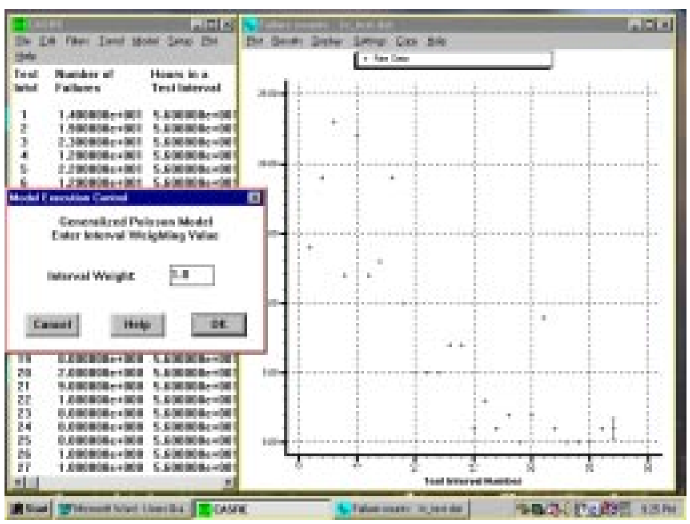

Figure 40 – Generalized Poisson model (user-specified test interval weighting) - dialog box for collecting additional model control information

3. For failure count data spanning test intervals "x" to "y", use the **cumulative failure count** from test interval "x" through "s-1", where "s" has values as described for the second treatment type. Select this treatment, shown in Figure 42, if you feel that all of the data are applicable, but if you're concerned that the data collection process was noisier for intervals "x" through "s-1". This will take into account the data that was collected in intervals "x" through "s-1", but in a cumulative fashion that will tend to remove some of the noise in the data. For intervals "s" through "y", the

individual failure counts for these intervals will be used. For example, suppose that you have a failure count data set that has 27 intervals, and you select a data range that starts at interval 8 and ends at interval 27. If you choose the Schneidewind model and the type of treatment of the data we're discussing now, you can choose a value of "s" between and including 8 and 27. If you choose a value of 12 for "s", the only contribution from intervals 8-11 will be the cumulative number of failures from those intervals. For intervals 12-27, the individual failure counts for each interval will be used in making predictions. Again, CASRE supplies a default value for “s” as described above.

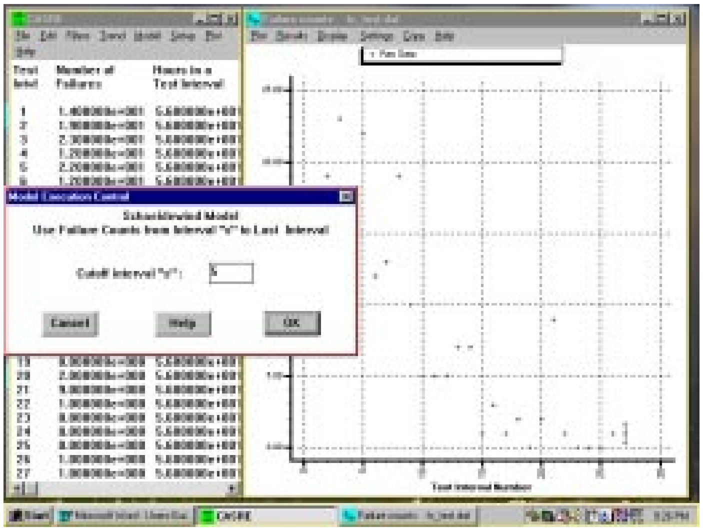

Figure 41 - Schneidewind model – ignore first “s-1” failures - dialog box for collecting additional model control information

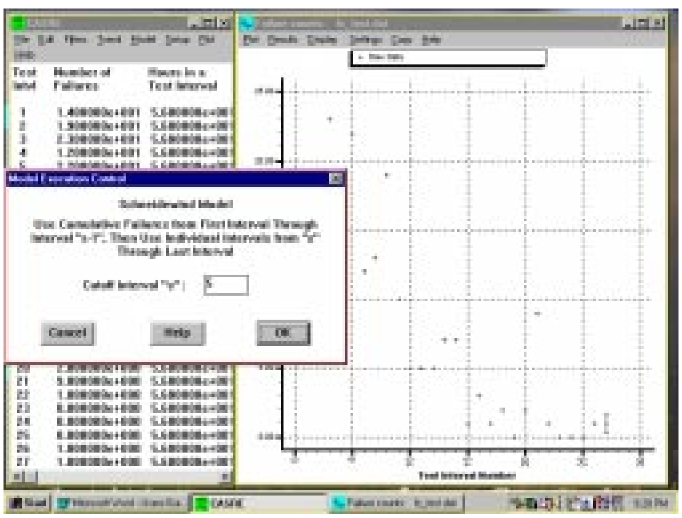

Figure 42 - Schneidewind model – total failures for first “s-1” test intervals - dialog box for collecting additional model control information

If a combination model has been selected, CASRE will execute all components of that combination, even if you haven't explicitly selected some of them. For example, each of the permanent combinations for time between failures data is composed of the Littlewood-Verrall Quadratic, Musa-Okumoto, and Nonhomogeneous Poisson Process models. Suppose you just want to run the ELC combination model. If you just select the ELC combination model, only the name "ELC" will appear in the "Models to Run" list of the "Select Models" dialog box. Since you haven't chosen any of the component models, their names will not appear in the "Models to Run" list. However, the component models will be run as part of running the combination, and the execution status of each component will appear in the modeling progress dialog box previously described. If a combination model contains component models that require additional control settings before they can be run, the dialog box for each of those components would appear as described above.

Since some of the models may take a relatively long time to run (see Appendix E), CASRE tries to minimize the number of times a model must be run during any particular session. For example, suppose you're working with a time between failures data set, and you've run the Littlewood-Verrall Quadratic, Musa Basic, and Musa Okumoto models. You might then decide to also run the Nonhomogeneous Poisson Process (NHPP) model in addition to these three. If you choose the NHPP model without removing the other three models from the "Models Run" list in the "Select Models" dialog box, only the NHPP model will be run, since the results of the other three models will still be available. Once you've chosen one or more non-combination models, CASRE will rerun them only under the following circumstances:

1. You've used one or more filters on the data since the last time you ran the models.

2. You've removed the effects of the last filter of all filters since the last time the models were run.

3. You've changed the number of steps ahead for which you want to make predictions.

4.9. Displaying and Evaluating Model Results

Once one or more models have been run, the results can be displayed in the graphic display window. Selecting the "Select model results..." item in the graphic display window's "Results" menu brings up a dialog box, shown in Figure 43 below, from which one or more sets of model results can be selected.

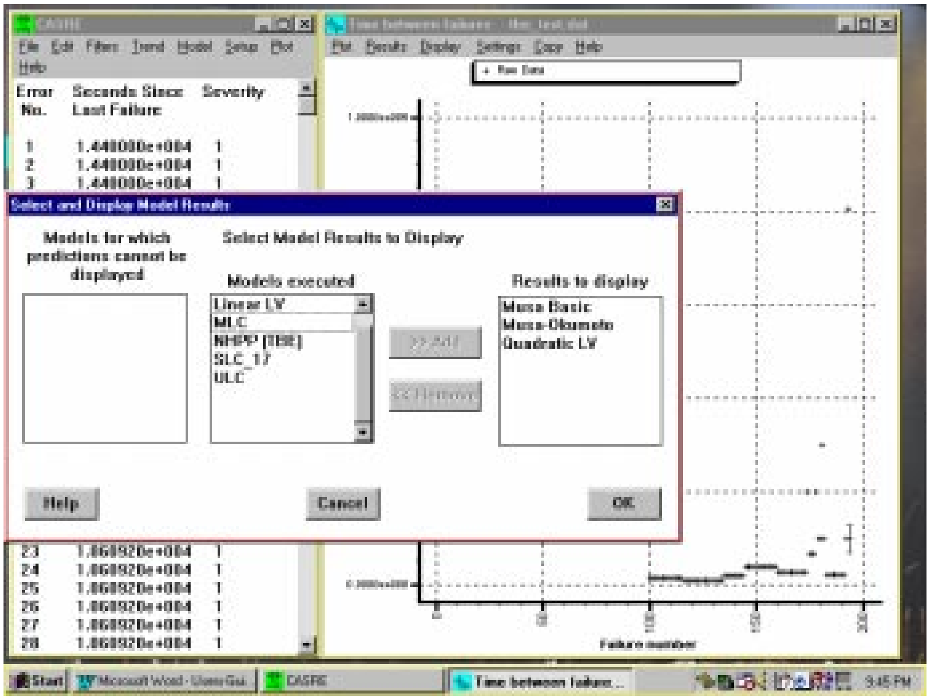

Figure 43 - Choosing model results to display

Selecting model results for display is very much like selecting models to run. Model results are selected by highlighting the name of a model in the "Models Executed" list, then selecting and clicking the "Add" button. You can also double-click on the name of the model in the "Models Executed" list. The selected model name then moves from the "Models Executed" list to the "Results to Display" list. Similarly, model names can be removed from the "Results to Display" list by highlighting the model name, then selecting and clicking the "Remove" button. As with selecting a set of results to display, you can double-click on the name of a model in the "Results to Display" list to remove that model from that list. In order to keep the graphic display window from becoming too cluttered, no more than three sets of model results can be selected for display. If you try to select more than three models, a message box appears to tell you that you can't choose more than three sets of model results to display.

Notice the leftmost list box in Figure 43. This list holds the names of models for which valid predictions could not be made. It may be the case, for instance, that parameter estimates using the maximum likelihood method did not converge for one or more models. If a model appears in this list, it will not appear in the "Models Executed" list, and you won't be able to select its results for display. If a model appears in the list of models for which predictions could not be made, you have two choices for trying to get valid predictions from that model.

1. Choose a different data range on which to apply models. It may be the case that the testing method or the testing environment changed in the middle of the specified data range (see paragraph 4.8.2 for details on specifying a modeling data range), and that choosing a data range for which the testing method and testing environment are constant will produce valid predictions.

2. The failure data may be too noisy. Since it is often more difficult to collect accurate times between failures than failure counts, noise is more of a problem with time between failures data. If the data collection methods allow too much noise to get into the data, a Hann window may be applied to the data to remove some of the noise. The model(s) may then be rerun, using the smoothed data as input. If enough noise has been removed, and if none of the issues raised in point 1 above are a problem, valid predictions may be produced.

   Data can also be grouped to reduce noise. Time between failures data could be converted to failure counts and test interval lengths using the "Change data type..." option of the main window's "Edit" menu. However, this would mean that the time between failures models could not be used on the converted data.

   For failure count data, the size of the test interval could be made larger to reduce noise, but at the cost of losing resolution. For example, if the current interval size is one hour, changing it to one day could remove some of the noise. Similarly, failures per day could be changed to failures per week, and failures per week could be changed to failures per month. This type of regrouping can be done within CASRE by first transforming failure count data to times between failure data, and then re-transforming to failure counts data with a different test interval length than the original data set. Details on changing one type of failure data to another are given in paragraph 4.4.

Upon selecting and clicking on the "OK" button in the "Select model results" dialog box, the model results are shown in the graphic display window. In the next two sections, we'll be looking at model results and model evaluation statistics for the following scenario.

Failure data:	Same as that shown in Figure 4. This is time between failures data.
Modeling interval:	We've selected points 100-194 as the interval to which to apply the data. The initial parameter estimation is from points 100-150. We showed this in Figure 35.
Models selected:	We've run all of the time between failures models, including the permanent combinations. We'll be looking at the results of the Musa Basic, Musa-Okumoto, and Littlewood-Verrall models.

4.9.1. Times Between Failures, Failure Counts, and Other Model Results

Model results are displayed in the graphics display window in the same fashion as the input data. In fact, they are superimposed on the raw data plot as shown in the following series of figures. After one or more sets of model results have been selected for display, as described in paragraph 4.9, the way in which the results are plotted matches the way in which the failure data is currently displayed. Suppose you had been displaying the failure data set in terms of cumulative number of failures. After selecting one or more sets of model results for display, those results would also appear in the form of cumulative number of failures. Likewise, if you had been displaying the raw data in terms of times between failures, selecting model results for display would cause those results to be displayed in terms of times between failures. This is shown in Figures 44 and 45 on the following pages. As stated above, the failure data is the same time between failures data shown in Figure 4, and in Figure 44 below we're displaying the results of the Musa Basic, Musa-Okumoto, and Littlewood-Verrall models in the same plot. Notice the message box in Figure 44. Recall that some models assume that there is an upper bound on the number of failures that you'll observe, and that other models assume that there is no such upper bound. The Musa Basic model is one of the models that assumes an upper bound. In Appendix I, you can find the other models making this assumption. In our running example, our data range for modeling is points 100-194, and we've stated that we want to predict the times between failures for failures 195-214. Given this input data, the Musa Basic model predicts that there are fewer than 20 failures left after point number 194, and the message box shown in Figure 44 appears to tell you this. Anytime you choose to display models assuming there is an upper bound on the number of failures, the message box in Figure 44 appears if the number of failures in the future for which you want to make predictions is more than the upper bound predicted by the model. To clear the message box from the screen, simply click on the "OK" button, and you'll get the plot shown in Figure 45 on the next page.

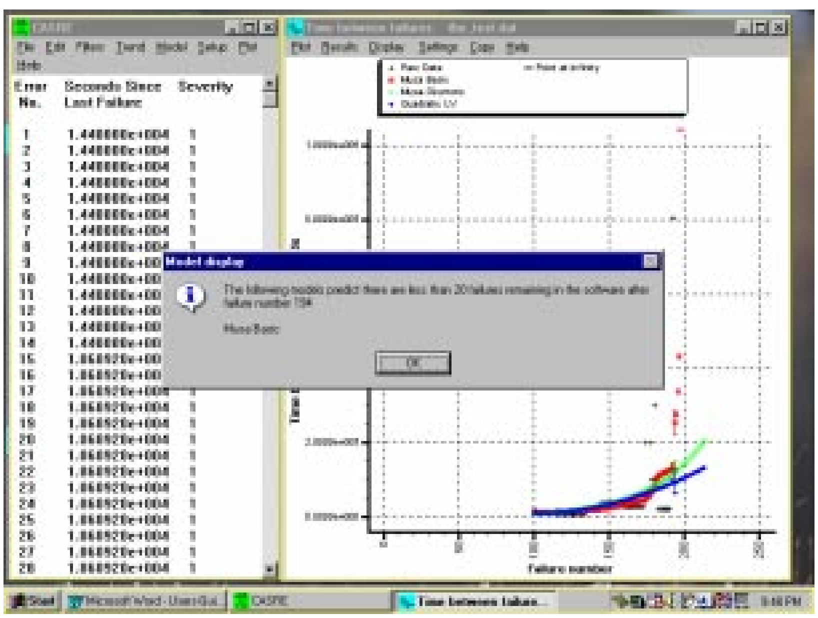

Figure 44 - Model results display for time between failures models - plot type is time between failures

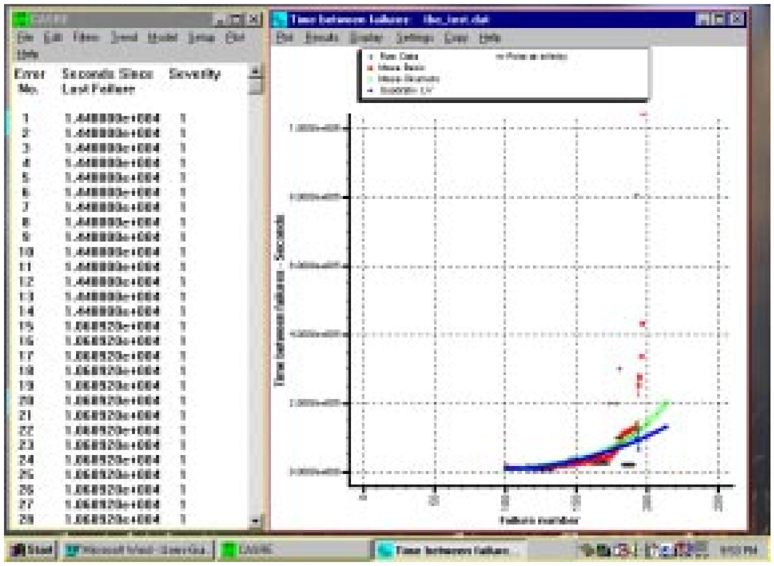

Figure 45 - Display of model results for time between failures models - plot type is time between failures (cont'd)

Let's take a closer look at Figure 45 and see what it tells us. If you look at the legend box near the top of the graphic display window, it names the model results you've selected for display and defines a unique symbol for each model as well as the raw failure data. On the plot itself, look at the results for each of the three models. For each model, you'll see a point that gives the estimated time to the next failure for failure numbers 100-214. For each set of results, points 100-194 represent the model's estimates of the actual times between failures that were observed. This is the information shown in the "Model Estimates" column of the model results table (see paragraph 4.10.1, item 4). Points 195-214 represent the predicted times to failure for the 20 failures after the last observation in the failure data set. This is similar to the information shown in the "Next Step Prediction" column of the model results table (see paragraph 4.10.1, item 3), except that in our example, point 195 is a 1-step ahead prediction, point 196 is a 2-step ahead prediction, and so forth. For each model result, point 194 is identified by a symbol that looks like an I-beam. All points to the right of that symbol represent a model's predictions of future behavior, while all points to the left of that symbol represent a model's estimate of what was actually observed. If you look at the results for the Musa Basic model, you'll only see 3 points to the right of the I-beam, plus one point labelled "Point at infinity." From this, you can see that the Musa Basic model predicts you'll only see three more failures after failure number 194; the point at infinity means that the model predicts that the time between failure number 197 and 198 is infinity.

You'll notice that the plot shown in Figure 45 is cluttered. You may be thinking that it would be nice to be able to zoom in on a portion of that plot to be able to pick out additional detail. We won't go into the details right now, but in paragraph 4.9.3, you'll see how to do this.

You can also display the model results in the form of cumulative number of failures. This is shown in Figures 46 and 47 on the following pages. Figure 46 shows the cumulative number of failures over all of the failure data, while Figure 47 shows the cumulative number of failures from the starting point of the modeling range (in this case, point 100). To display the plot shown in Figure 46, choose the "Cumulative failures" item in the graphic display window's "Display" menu, then choose the "Over all data" item in the submenu that appears. To produce the plot shown in Figure 47, choose the "From model start point" item on the submenu instead of the "Over all data" item. Let's take a look at what Figures 46 and 47 say. In Figure 46, we see the cumulative number of failures as a function of elapsed test time. For the raw failure data, we see all 194 failures, regardless of the modeling range we've specified. For the model results, however, we don't see quite the same thing. Recall that we started the models at failure number 100, so the model estimates don't start at 0 on the y axis, but at 100. In Figure 47, however, we only show the total number of failures, starting at the first point of the modeling range. For our running example, we'll only see 95 points for the raw data. Both the model results and the raw failure data will start at 0 on the y axis. In Figures 46 and 47, the model results that you see **to the right** of the vertical line drawn through the I-beams' centers represent predictions about the future, while the points **to the left and including** that vertical line represent the models' estimates of the failure history that we've already observed. **Note that although the Musa Basic model predicts that there are fewer than twenty failures remaining after failure number 194, there are still 20 predictions for the cumulative number of failures displayed. In the case of the Musa Basic model, the curve approaches an asymptotic limit. In general, when drawing predictions for cumulative number of failures, failure intensity, and reliability for TBF models, CASRE draws curves based on the mean value function. Suppose that we have a data set of "i" points and a total testing time of "T". Point "i+1" will represent the cumulative number of failures, reliability, or failure intensity at time "T+0.05T", point "i+2" represents the prediction at time "T+0.1T", and so forth.**

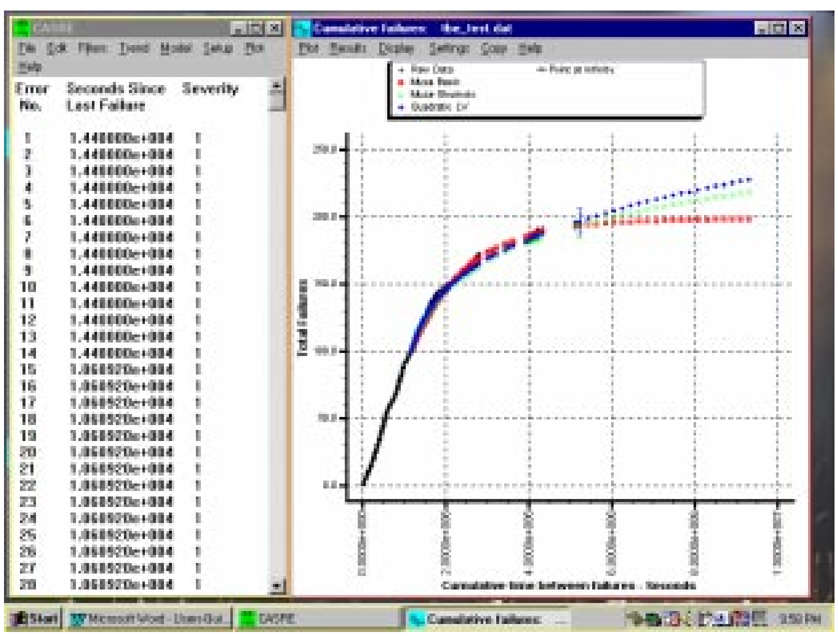

Figure 46 - Display of model results - cumulative number of failures for time between failures models

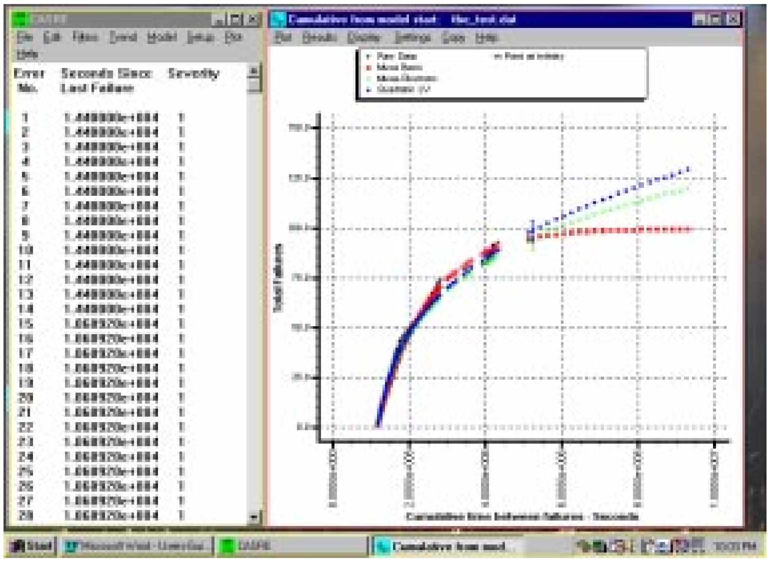

Figure 47 - Model results display - total number of failures from model start point for time between failures models

These curves are drawn this way to make sure that the predictions for different models are shown at the same cumulative testing time. This temporal alignment makes it possible to easily use combination models to predict the cumulative number of fail-ures, failure intensity, and reliability growth for TBF data. It would have been possible to draw these curves so that for each model, predictions of failure intensities, cumulative number of failures, and reliability growth would be displayed at the time at which each subsequent failure was expected to occur. Using our running example, this means that for each model, we could have displayed the predicted failure intensities at the times at which we would expect to see failures 195, 196, ..., 214. These expected times to future failures, however, differ with each model, so the predictions made by one model would be offset in time with respect to other models' predictions. Looking back to Figure 45, for example, we can see that the expected time to failure 197 is considerably larger for the Musa Basic model than it is for the Littlewood-Verrall model. This temporal misalignment would make it considerably more difficult to form combination models. Ease of forming combination models and consistency in the displayed predictions were the primary considerations in choosing the display convention described above.

The way in which the software's reliability changes over time can also be shown in the graphic display window. Briefly, almost all software reliability models assume that reliability will increase as testing progresses and failures are found and removed. The Littlewood-Verrall model does not make this assumption. You can see what other models' assumptions about reliability growth are in Appendix I. A plot of reliability vs. time should start with low values of reliability for small values of elapsed time (close to the start of test), and increase as time progresses and more testing occurs. Reliability always has values between 0 and 1.

To display software reliability as a function of time, you'll have to specify the length of the time interval for which reliability will be calculated (recall the definition of software reliability - the probability of running without failure in a specified environment for a specified time). When the "Reliability..." item in the graphic display window's "Display" menu is chosen, the dialog box previously shown in Figure 11 appears on screen to record the length of the interval. The default values for the interval depend on the type of the failure data. If the failure data is time between failures, the length of the interval is the last observed time to failure in the data range you've chosen. For failure count data, the default value is the length of the last test interval in the data range you've specified. If you're interested in the reliability of the software for a different amount of time than the default value, you can enter a new value into the dialog box's edit window. For our example, we'll enter a value of 20,000, indicating that we want to know the probability of the software running for 20,000 seconds without failure. Selecting and clicking the "OK" button will cause plots of the reliability estimates for each selected model result to appear in the graphic display window. For our running example of time between failures data, the resulting plot is shown in Figure 48 below.

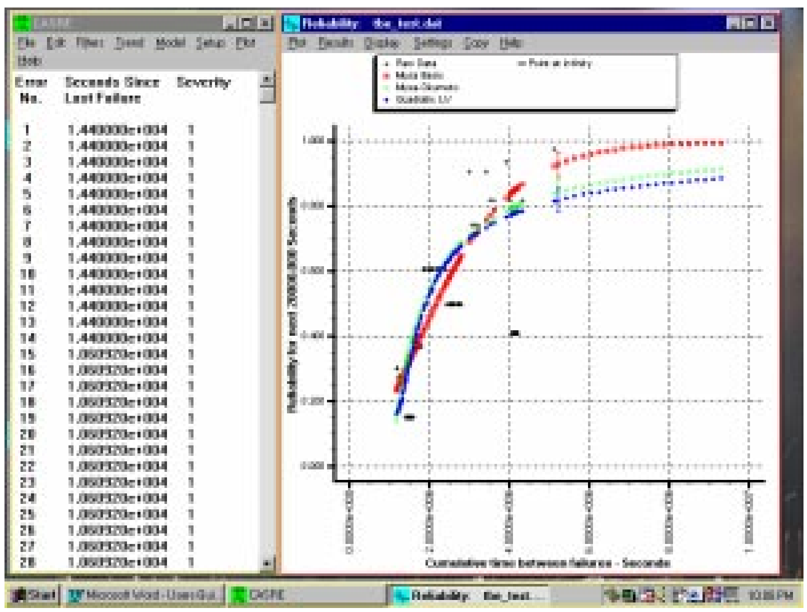

Figure 48 - Display of model results - reliability growth curve for time between failures models

Here's what Figure 48 tells us. By the time we've spent 2,000,000 seconds testing, each model predicts that the reliability will assume a specific value. After 2,000,000 seconds of testing, then, the models predict that the probability of the software running for 20,000 seconds without failure is:

$$\exp\left(\mu(2000000)_{Musa\ Basic} - \mu(2020000)_{Musa\ Basic}\right) \text{ for Musa Basic}$$

$$\exp\left(\mu(2000000)_{Musa\ Okumoto} - \mu(2020000)_{Musa\ Okumoto}\right) \text{ for Musa-Okumoto}$$

$$\exp\left(-\int_0^{20000} h(x)_{Quadratic\ LV}\, dx\right) \text{ for the Quadratic Littlewood-Verrall model}$$

where $\mu(t)$ denotes the mean value function (expected total number of failures by time t). In Figure 48, this shows up as about 0.4 for the Musa Basic model, and about 0.5 for the other two models. If we continue testing, we'll see more failures, and since we're removing faults while we're observing failures, we'll tend to have lower hazard rates the more we test. In fact, if we look at the plot in Figure 48 after we've spent 4,000,000 seconds testing, we see the Musa Basic model predicts that the probability of running for 20,000 seconds without failure is about 0.85, while for the other two models it's about 0.75. For each set of model results shown in Figure 48, the points to the right of the I-beam symbol are predictions of what the reliability will be for failures 195-214, while the symbols to the left and including the I-beam are the model's estimates of the actual reliability that we've observed.

For failure count data, there are additional displays that we can see. These are failure counts and test interval lengths. We'll briefly leave our running example of times between failures data, and use the failure count data shown in Figure 5 to show these additional plots. For this failure counts example, we'll select a modeling range starting at test interval 5 and ending with interval 27. We'll make predictions of failure counts for test intervals 28-37. To illustrate the point that test intervals do not have to be of equal length, we'll specify a length of 60 hours for test intervals 28-37. As with the time between failures data, we'll run all of the models available, choosing to display the results of the Schick-Wolverton, the Nonhomogeneous Poisson Process (for failure count data), and the Yamada S-Shaped models. The failure counts plot is shown in Figure 49 below, while the plot of test interval lengths is shown in Figure 50 on the next page.

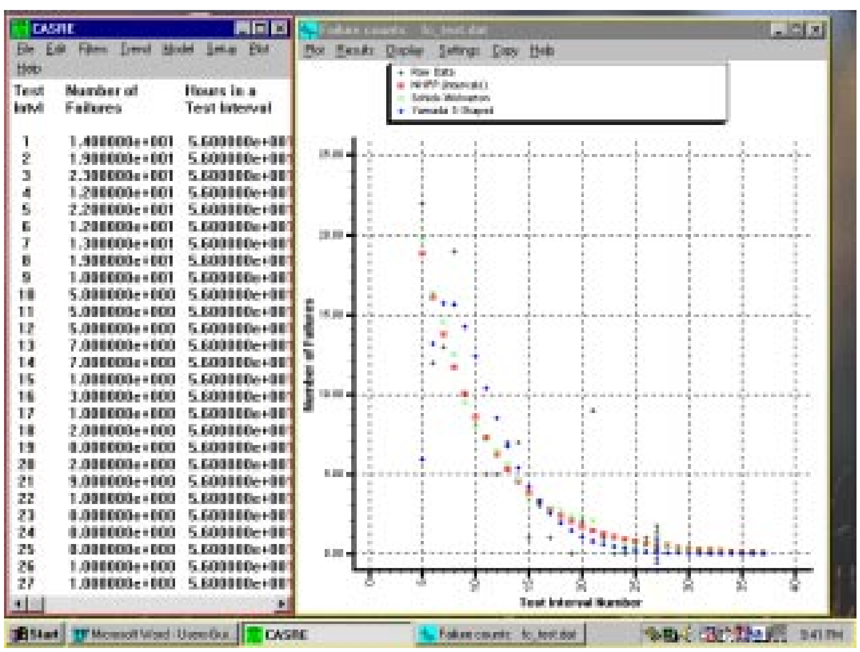

Figure 49 - Model results display for failure count models - failure counts plot type

In Figure 50, we see that the lengths of the test intervals for the failure data are all of equal length (56 hours in this case). Since we've specified a test interval length of 60 hours for test intervals 28-37, the symbols to the right of the I-beam symbol at test interval 27 all indicate test interval lengths of 60 hours. Predictions for failure counts, cumulative number of failures, failure intensity, and reliability for intervals 28-37 will be based on a test interval length of 60 hours.

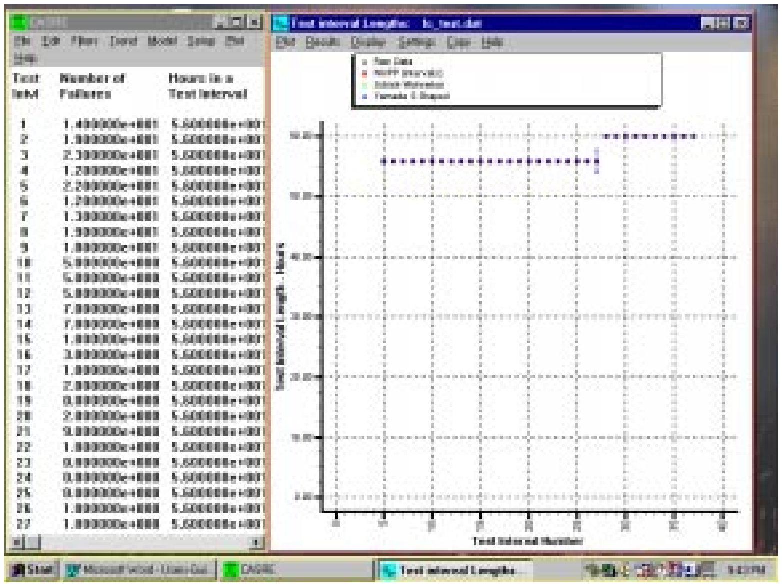

Figure 50 - Model results display for failure count models - test interval lengths plot

Finally, we can display the reliability estimated by failure count models. For our example, we'll show the estimated reliability for a 28-hour period. Using the dialog box first shown in Figure 11, we enter a value of 28 into the edit window. The reliability predicted by the failure count models is shown in Figure 51 below. Again, we assume that the failures are evenly distributed within each test interval.

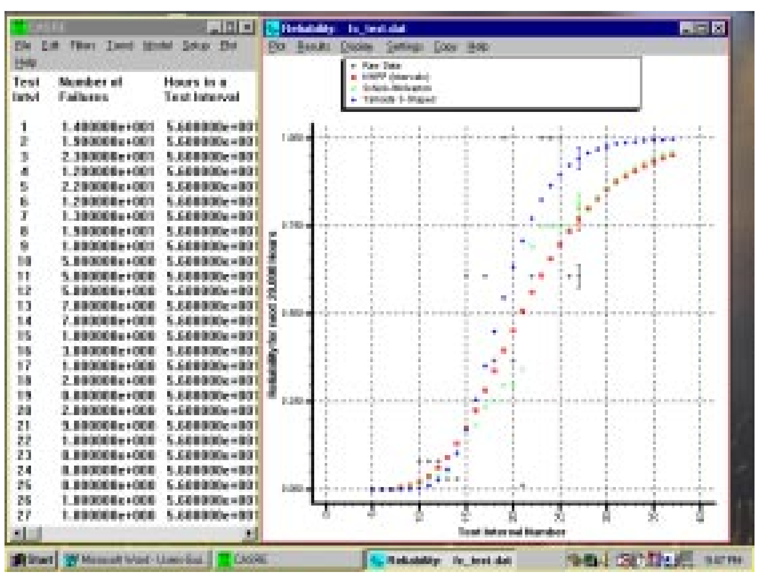

Figure 51 - Display of model results - reliability growth curve for failure count models

4.9.2. Model Evaluation Statistics Display

In addition to modeling results, statistics relating to the predictive accuracy of each model can be displayed in the graphic display window. These statistics have to do with the relative accuracy of one model with respect to another, model biases, model prediction noise, and goodness-of-fit. A brief description of the first four items is given in this section. These items are thoroughly discussed in a paper that appeared on pages 950-967 of the September, 1986, edition of the IEEE Transactions on Software Engineering. The title of the paper is "Evaluation of Competing Software Reliability Models", written by A. A. Abdel-Ghaly, P. Y. Chan, and Bev Littlewood [Abde86].

Prequential likelihood is a measure that can tell you how much more appropriate one model is than another. The **prequential likelihood ratio** can be used to discredit one model in favor of another for a particular set of failure data. Consider a random variable T_j, which represents the time to the next failure. Recall from paragraph 4.7.1 that software reliability models assign a value to T_i based on the form of the model and the values of the previous (j-1) observed times between failures t_1, t_2, …, t_{j-1}. The models have the form of a predictive probability density function (pdf), $\tilde{f}_j(t)$, whose parameters have been estimated from the previously observed times between failures. Using this predictive pdf, we can find the value of the next time to failure, represented by the expected value of the random variable T_j. This expected value is given by:

$$E[T_j] = \int_0^\infty t \tilde{f}_j(t)dt$$

Now we will use the predictive pdf to compute a model evaluation statistic called the *prequential likelihood*. Assuming that we have observed j failures, for one-step ahead predictions of T_{j+1}, T_{j+2},…, T_{j+n}, the prequential likelihood is given by:

$$PL_n = \prod_{i=j+1}^{j+n} \tilde{f}_i(t_i)$$

A comparison of two models, A and B, may be made by forming the prequential likelihood ratio $PLR_n = PL_n^A/PL_n^B$. The reliability practitioner believes that either model A is true with p(A) or that model B is true with probability p(B) = 1 - p(A). The practitioner observes the failure behavior of the system, makes predictions using the two models A and B, and compares the predictions to the actual behavior via the prequential likelihood ratio. When predictions have been made for T_{j+1}, T_{j+2},…, T_{j+n}, the PLR is given by:

$$PLR_n = \frac{p(t_{j+n},\ldots,t_{j+1} \mid t_j,\ldots,t_1, A)}{p(t_{j+n},\ldots,t_{j+1} \mid t_j,\ldots,t_1, B)} = \frac{PL_n^A}{PL_n^B}$$

Using Bayes' Rule, PLR_n is rewritten as:

$$PLR_n = \frac{\dfrac{p(A|t_{j+n},\ldots,t_1)p(t_{j+n},\ldots,t_{j+1}|t_j,\ldots,t_1)}{p(A|t_j,\ldots,t_1)}}{\dfrac{p(B|t_{j+n},\ldots,t_1)p(t_{j+n},\ldots,t_{j+1}|t_j,\ldots,t_1)}{p(B|t_j,\ldots,t_1)}}$$

$$= \frac{p(A|t_{j+n},\ldots,t_1)}{p(B|t_{j+n},\ldots,t_1)} \cdot \frac{p(B|t_1,\ldots,t_j)}{p(A|t_1,\ldots,t_j)}$$

If the initial predictions were based only on prior belief, the second factor of the last equation is the prior odds ratio. If the user is indifferent between models A and B at this point, this ratio has a value of 1, since p(A) = p(B) for the first j failures. The last equation is then rewritten as:

$$PLR_n = \frac{p(A|t_{j+n},\ldots,t_1)}{p(B|t_{j+n},\ldots,t_1)} = \frac{w_A}{1 - w_A}$$

This is the posterior odds ratio, where w_A is the posterior belief that A is true after making predictions with both A and B and comparing them with actual behavior. If $PLR_n \to \infty$ as $n \to \infty$, model B is discarded in favor of model A.

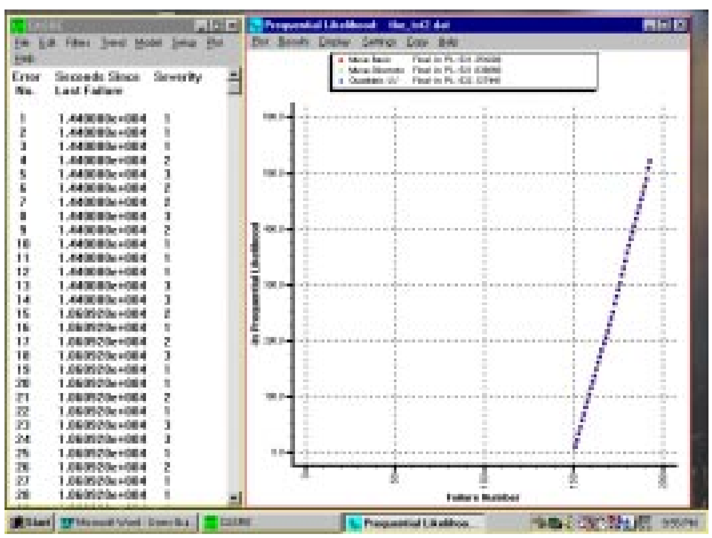

Figure 52 - Model evaluation statistics - plot of -ln(PL_i) for time between failures models

Prequential likelihood is displayed as shown in Figure 52 above. This plot is shown by selecting the "Prequential likelihood" item in the graphic display's "Display" menu. Since prequential likelihood values can get very small, we use the negative of the natural log of the prequential likelihood. This value is plotted as a function of error number (or test interval number for failure counts data), starting with the first observation past the end of the initial parameter estimation range, and continuing until the end of the failure data range, up to and including the next to last observation in the data set. This plot, which shows the way in which each model's prequential likelihood measure changes with time, indicates the predictive accuracy of one model with respect to the others that were run. Given the prequential likelihood measures for two models, we can use the prequential likelihood ratio to see how much more likely it is that one model will produce more accurate predictions than the other. For any two models, A and B, the ratio PL_A/PL_B tells you how much more likely it is that model A will give more accurate predictions that model B. The legend box gives the final value of the negative of the natural log of each model's prequential likelihood measure.

In Figure 52, we see that the prequential likelihood statistics for the three models are very close to one another. The relative accuracy plot, related to the prequential likelihood plot in Figure 52, shows you directly how much more likely one model is to produce more accurate predictions than another. You can show this plot by selecting the "Relative accuracy" item in the graphic display's "Display" menu. Suppose the Musa-Okumoto, Musa Basic, NHPP, and Littlewood-Verrall models have been run. You can then select the results of the Musa-Okumoto, Musa Basic, and Littlewood-Verrall models to be displayed. If you then select the "Relative accuracy" display, you will see the plot shown in Figure 53 on the next page, which shows how much more likely it is that one model will produce more accurate predictions than the others. For every point in the failure data set, we compute the relative accuracy in our example as follows:

1. At each point **i**, find the model having the smallest prequential likelihood value (the largest value of $-\ln(PL_i)$). Call this model "PLMin".
2. Assign a value of 1 to the relative accuracy of model "PLMin" at that point.
3. Suppose that at some point in the data set, the Musa Basic model has been identified as "PLMin". Compute the prequential likelihood ratios PL(Musa-Okumoto)/PL(Musa Basic) and PL(Littlewood-Verrall)/PL(Musa Basic). These will be the relative accuracy values assigned to the Musa-Okumoto and Littlewood-Verrall models, respectively. This will tell you how much more likely it is at that point that the Musa-Okumoto and Littlewood-Verrall models will produce more accurate estimates than the Musa Basic model.

In Figure 53, notice that while the models are processing failure numbers 100 through about 185, no model is ever more than 2.5 times as likely as any other to produce more accurate predictions. During the time when the models are analyzing failures in the range of 185-190, the Littlewood-Verrall model is clearly the preferred model, becoming nearly 15 times more likely to produce more accurate predictions than the Musa Basic model and twice as likely as the Musa-Okumoto model at point 190. At the end of the data set, however, we see that the Musa Basic model is 2.16 times as likely to produce more accurate predictions than the Littlewood-Verrall model, while the Musa-Okumoto model is about 1.33 times more likely to produce more accurate predictions than the Littlewood-Verrall model. This illustrates how the preferred model can change as the testing effort progresses. You can see the final values for the relative accuracy of each model in the plot's legend box.

We can use the predictive probability density function described in paragraph 4.7.1 to determine whether a model is **biased**. A model is biased if its predictions show a systematic deviation from the failure behavior that is actually observed. For instance, a model may consistently predict times between failures greater than those that are actually observed. In this case, we would say that the model has an optimistic bias. A model might also make predictions of times between failures that are less than those actually observed. We would say that the model exhibits a pessimistic bias. We can also use the predictive probability density function to determine whether there are trends in a model's bias. It may be the case that a model makes optimistic predictions during the first half of the testing phase, and makes pessimistic predictions during the last half of the testing phase. Although the model's predictions are definitely biased, this fact might not show up in an analysis of model bias. To see this behavior, it would be necessary to do an analysis of model bias trend.

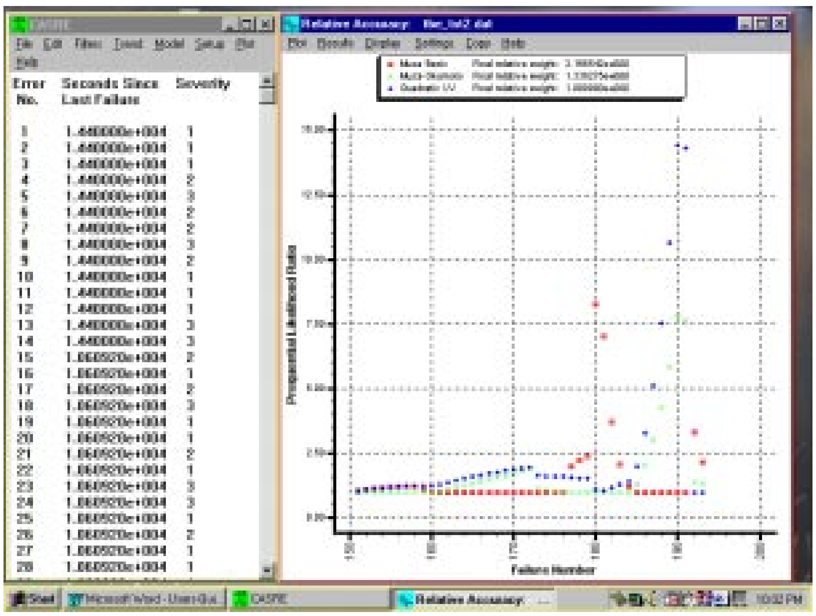

Figure 53 - Model evaluation statistics - relative accuracy plot for time between failures models

Recall from paragraph 4.7.1 that a model's predictive probability density can be used to determine the probability that a software system will failure before a certain time, **t**. Suppose that we've observed i times between failures, $t_1, t_2, \ldots, t_i$, and we want to see what the model has to say about the probability of failing before t_i, given that the (i-1)th failure has already been observed. As in paragraph 4.7.1, we can compute a predictive probability density, $\tilde{f}_i(t)$, based on the first (i-1) observed times between failures. We can let the random variable $\mathbf{u_i}$ represent the model's estimation of the probability of failure that we're looking for, which is given by the following equation:

$$u_i = \int_0^{t_i} \tilde{f}_i(t)dt$$

This probability of failure before time t_i is also expressed as $(1-R_i(t_i))$, where $R_i(t_i)$ is the reliability for an interval of length t_i predicted by the time the i'th failure has been observed.

The first way in which way we can display a model's bias is the plot shown in Figure 54 below. This is the "Bias scatter plot" item in the graphic display window's "Display" menu. This is simply a scatter plot of the u_i, defined above, versus the failure number, **i**. This plot was produced by applying the Jelinski-Moranda and Musa-Okumoto models to one of the data sets on your distribution disk, "s1.dat". An unbiased model should produce values for u_i that are evenly distributed throughout the plot. Biases in the model will be revealed by clusters in the plot. Clusters close to the x-axis, for instance, would indicate intervals during which the model's predicted probability of failure is low, meaning the optimistic predictions are being made. Clusters near the top of the plot would indicate intervals during which the model's predictions of failure probability are high, meaning that the predictions are pessimistic. In Figure 54, we're looking at values of u_i for $68 < i < 136$. For the interval $68 < i < 101$, we see that for the Jelinski-Moranda model, there are 12 instances for which u_i is greater than 0.5. This leaves 20 instances for which $u_i \leq 0.5$, indicating that the model produces optimistic predictions during this interval. During this same interval, there are 21 out of 32 instances for which the Musa-Okumoto model produces values of $u_i > 0.5$, indicating that this model tends to produce pessimistic predictions during this interval. If we now look at the interval $100 < i < 136$, we see that for the Jelinski-Moranda model, there are only 6 out of 36 instances for which $u_i > 0.5$. This indicates that the model exhibits an even greater tendency to make optimistic predictions than during the previous interval. For this same interval, we see that there are 14 instances during which the Musa-Okumoto model produces values of u_i that are greater than 0.5, indicating that the model may have a small optimistic bias during this interval.

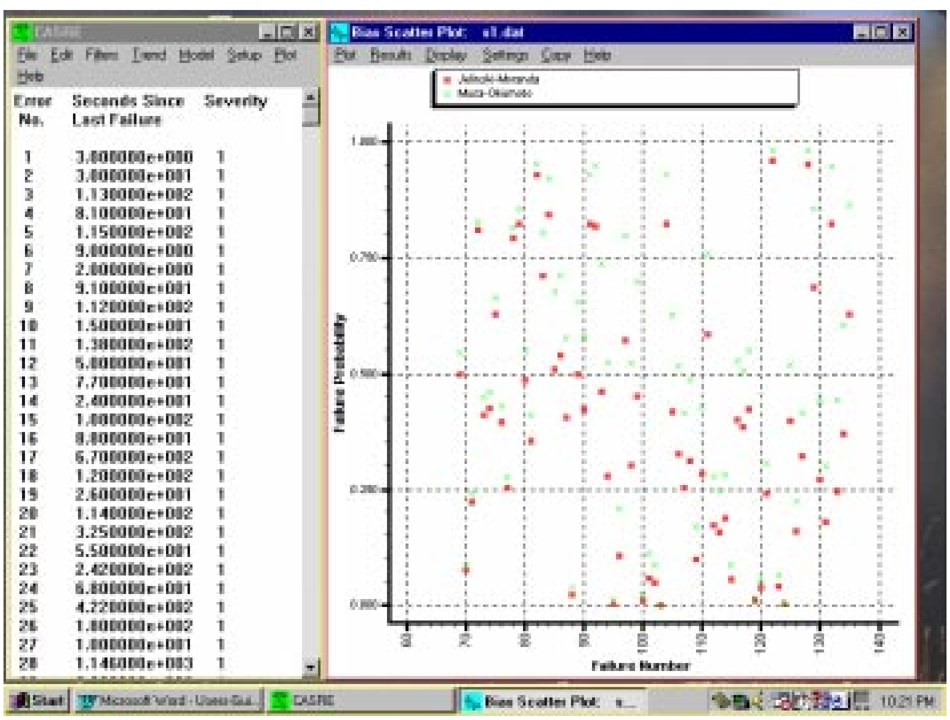

Figure 54 - Model evaluation statistics - scatter plot of model bias

Now consider how the failures that we observe are actually produced. The time between two successive failures depends on the environment in which the software is being run, the purpose for which it is being used (the operational profile), the clock speed of the computer, and many other factors. We can think of the process that produces the failures as a random process, in which the time between the (i-1)'th and the i'th failures is determined by the "true" probability density function $f_i(t_i)$. Unlike the probability density function that we get from the model equations, we don't know the form of $f_i(t_i)$. This is because we don't have enough knowledge about the usage patterns and execution environment of the software to determine its form. Even though we don't know $f_i(t_i)$, we can still determine the pdf that we'd get by transforming the random variables t_i into new random variables r_i using the same sort of probability integral transformation that we used to produce the u_i. The equation for the pdf of r_i is:

$$pdf(r_i) = \frac{f_i(t)}{\left.\dfrac{d\int_0^{t_i} f_i(t)dt}{dt}\right|_{t_i}} = \frac{f_i(t)}{f_i(t)} = 1$$

In short, u_i represents the model's viewpoint of the probability of failure, while r_i represents the "true", but unknowable, probability of failure.

Like u_i, r_i is a probability, so its values lie between 0 and 1. No matter what the value of r_i is, the value of the pdf of r_i is always 1. We determine a model's tendency to being biased by examining the closeness of the cumulative distribution functions (cdf) for u_i and r_i. To do this, we draw the cdfs for u_i and r_i and determine the maximum vertical distance (Kolmogorov distance) between them. Since the u_i are discrete random variables, the cdf for u_i is a step function, drawn according to the following rules:

1. Sort the u_i in ascending order.

2. Suppose that there are **n** values of u_i. The cdf of the u_i is a step function that increases by 1/(n+1) for every u_i. This means that the cdf for the smallest u_i has a value of 1/(n+1), the cdf for the next smallest u_i is 2/(n+1), and so forth until the largest u_i is reached. The cdf for the largest u_i is n/(n+1).

The cdf for r_i has the form of a straight line between the origin and the point (1, 1). The value of the Kolmogorov distance measures the extent to which the model is biased. You can display this type of plot, called a **u-plot**, by choosing the "Model bias" item in the graphic display window's "Display" menu. An example of this type of plot is shown in Figure 55 on the next page. The same failure data set and models were used to produce Figure 55 as were used for Figure 54.

If the cdf for u_i is above that for r_i, the model will yield optimistic (too large) estimates for the time to failure. Otherwise, if the cdf for u_i is below that for r_i, the model's estimates of time to failure will tend to be pessimistic (smaller than the observed times to failure). We can illustrate these points with examples. First, suppose we have a situation in which the u_i all have values between 0 and 0.25. This means that when we draw the u-plot according to the rules above, the step function representing the cdf for the u_i will rise very rapidly. By the time $u_i = 0.25$ is reached, the value of the cdf will be nearly 1. The curve of this cdf will be above the straight line representing the cdf of r_i. Now let's look at what it means when the u_i have low values. Recall that the u_i are the probabilities of failing at or before time t_i according to the model. If the values of u_i tend to be low, this means that according to the model, the probability of failing at or before time t_i will tend to be low. If the probability of failing before time t_i is low, this means that the model expects the time to the next failure to be larger than t_i. If the model expects the time to the next failure to be larger than the actual observed time to failure, the model's prediction is optimistic. Now we can look at the situation in which the u_i are clustered at the right end of the x-axis. Suppose that the values of u_i lie between 0.75 and 0.99. When we draw the step function representing the cdf of u_i, we can see that this step function will be below the straight line representing the cdf of r_i. If the values of u_i tend to be high, this means that the model expects the probability of failure at or before t_i to be high. This, in turn, means that the model expects the time to the next failure to be lower than the actual observed time to failure, which means that the model is making pessimistic predictions. In Figure 55 below, the bias plot shows that the Jelinski-Moranda is biased in the direction of making optimistic predictions about the time between failures. The Musa-Okumoto model closely follows the cdf for r_i, indicating that it has no overall bias.

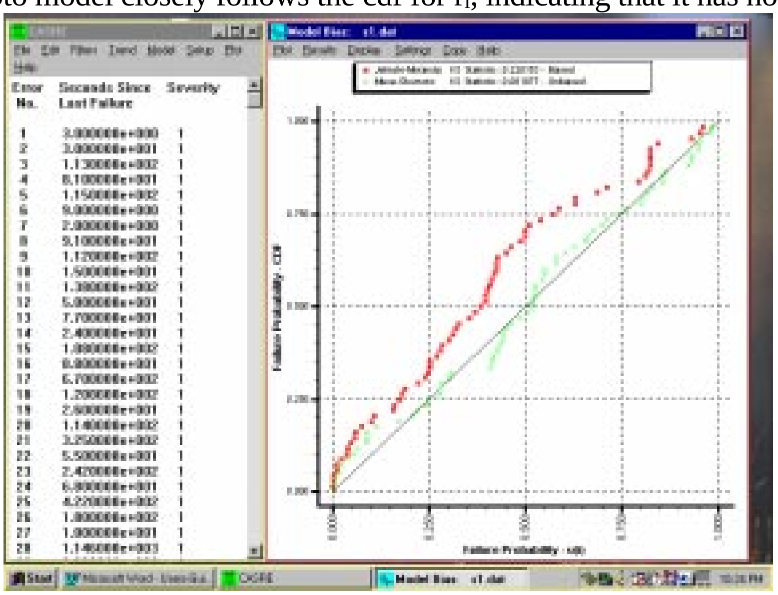

Figure 55 - Model evaluation statistics - bias plot for time between failures models

In Figure 51 above, the legend box gives the Kolmogorov distance for each of the models. It also tells you whether the Kolmogorov distance indicates that a model is biased at the 5% significance level. You can use the Kolmogorov distance values, the number of points in the plot for each model, and a table of critical values for the Kolmogorov-Smirnov two-sample statistic to see if the model is biased at other significance levels. Note how this differs from the bias scatter plot - for the bias scatter plot, CASRE implements no statistical techniques to help determine whether a model is biased at a particular significance level.

The bias plot does not show the way in which a model's bias may change over time. It is possible, for example, that the same model will tend to make pessimistic predictions during the early portion of a testing effort, and optimistic predictions during the later testing stages. This would show up as an unbiased model on the bias plot. Selecting the "Bias trend" option of the graphic display window's "Display" menu will generate the plot shown in Figure 56 below. This figure was produced using the same models and failure data set as for Figure 52. The interpretation is similar to that for the bias plot, except that the vertical distance between the straight line of unit slope and a model's plot represents the model's bias at a particular observation. This plot is known as a **y-plot**.

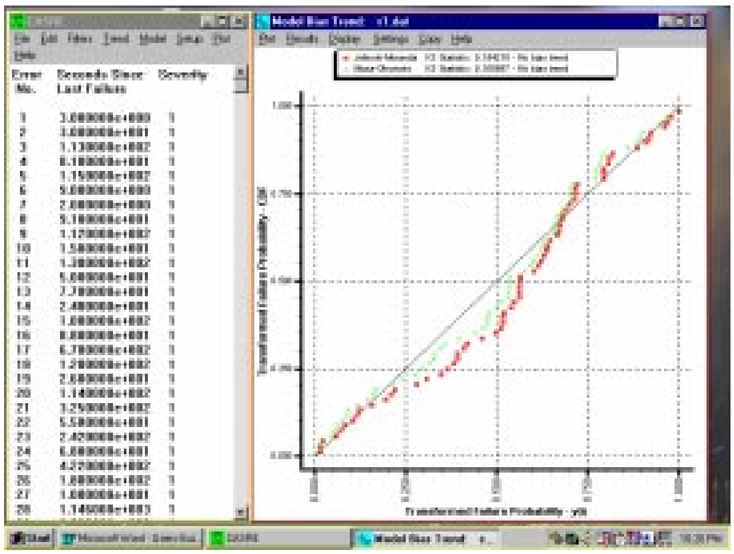

Figure 56 - Model evaluation statistics - model bias trend for time between failures models

Recall that in drawing the u-plot, we had to sort the u_i in ascending order to plot the cdf. This may cause temporal information about the model bias to be lost. Suppose we have a set, U, of u_i. In producing a u-plot, we create a new set, U', which contains the values of u_i sorted in ascending order. If the largest values of u_i were at the start of the set U, they would be moved to the end of the set U'. Since the positions of u_i can change when creating U', information relating to the way model bias changes with time can be lost. To preserve the temporal information lost in producing the bias plot, we have to perform two additional transformations after producing the random variable u_i. The transformation sequence is:

$$x_i = -\ln(1 - u_i),\ y_i = \frac{\sum_{j=1}^{i} x_j}{\sum_{j=1}^{n} x_j}$$

where n is the total number of failures observed. The random variable r_i is transformed in the same way. The cdf of the y_i and the cdf for the transformed r_i are then drawn, as was done for the u-plot. As with r_i, the cdf for the transformed r_i turns out to be a straight line of unit slope passing through the origin. This y-plot reveals temporal trends in the u_i. The point at which the cdf of the y_i departs from the cdf for r_i indicates the time at which the estimates made by the model are biased. In Figure 56 on the previous page, you can see that the models might be biased toward making pessimistic predictions for about the first two-thirds of the testing effort, while they may be biased toward making optimistic predictions for the last third of the testing interval. However, the legend box in the plot gives you the Kolmogorov distance for each model, and tells you whether or not the model exhibits any bias trend at the 5% significance level. In this case, neither one of the models being displayed exhibits bias trend at this significance level.

The noise introduced into a model's predictions by the model itself can be displayed by selecting the "Noise" item of the "Display" menu. Model noise is defined by the following equation:

$$\textit{Model Noise} = \sum_i \left| \frac{t_i - t_{i-1}}{t_{i-1}} \right|$$

The quantity $\mathbf{t_i}$ is the prediction for the i'th time between failure made by the model. This display, shown in Figure 57 on the next page, shows the amount of noise introduced into the predictions by the model itself. As the value for noise increases, the predictive accuracy of the model tends to decrease. Unlike the plots shown so far, this plot includes every model that was run. Note that model noise is not an absolute measure of model applicability in the sense of traditional goodness-of-fit tests or the bias measures we've discussed. Neither does it yield the likelihood that one model will provide more accurate predictions than another, as do the prequential likelihood and relative accuracy statistics. Of all of the model evaluation criteria described so far, this will probably be the least used. The prequential likelihood statistic includes aspects of the model bias and noise measures. Together with the goodness-of-fit tests, prequential likelihood will probably be the most frequently used criterion in determining a model's applicability to the failure data being analyzed.

Finally, traditional goodness-of-fit tests may be applied. You can use these tests to determine how closely the model results fit the actual failure data. As does SMERFS, CASRE employs two goodness-of-fit tests: the Chi-Square test is applied when failure counts models have been run, and the Kolmogorov-Smirnov (KS) test is applied when TBF type models have been run. For time between failures models, selecting the "Goodness of fit" item in the "Display" menu brings up the display shown in Figure 58 on the next page. The goodness-of-fit test that was applied is identified. For each model that was run, the following items appear in the scrolling text display:

1. The name of the model.
2. The goodness-of-fit value (Kolmogorov Smirnov test statistic for time between failures models).
3. For time between failure models, "Yes" indicates that the model is a good fit at the 5% significance level, while "No" means that the model does not fit at that level.

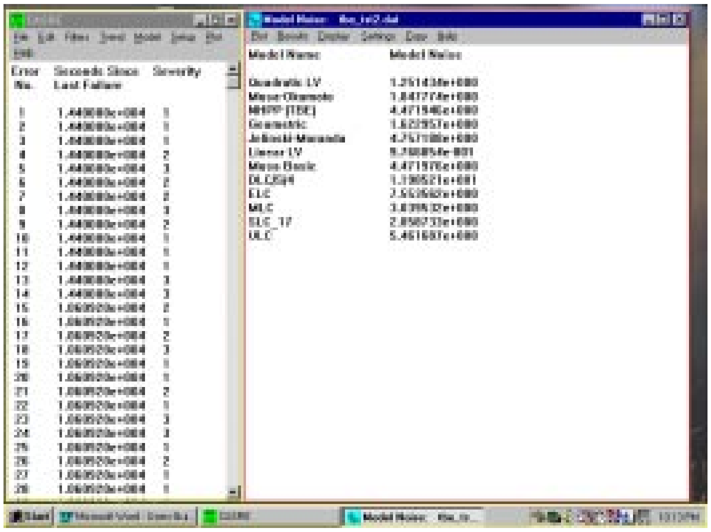

Figure 57 - Model evaluation statistics - Model noise for time between failures models

If you look at Figure 58, you can see that for the models we've run, none of them provide a fit to the data at the 5% significance level. If you're using failure count models, the display you get when you choose the "Goodness of fit" menu item is shown in Figure 59 on the next page. The differences between this display and the one shown in Figure 58 are:

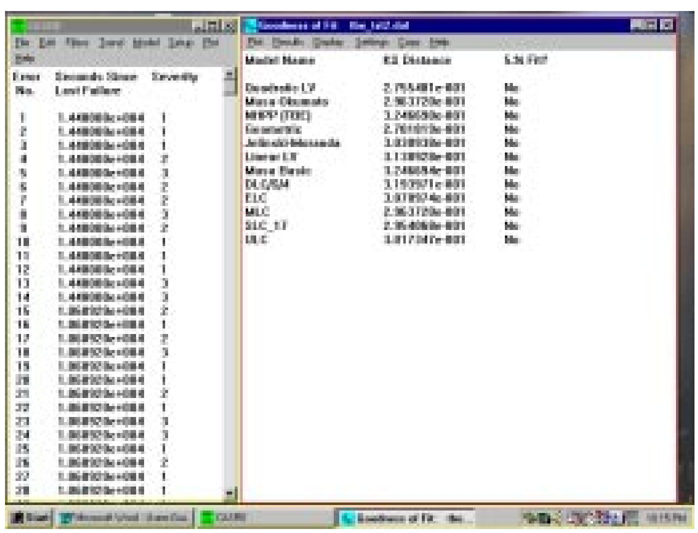

Figure 58 - Model evaluation statistics - goodness of fit display for time between failures models

1. The second column gives the value for the Chi-Square test statistic instead of that for the Kolmogorov-Smirnov test.

2. Instead of indicating whether the model fits the data at the 5% significance level, the third column gives you the number of degrees of freedom. The number of degrees of freedom is related to the number of data points and the number of parameters in the model. For ungrouped data, the number of degrees of freedom is the number of data points minus the number of model parameters. CASRE groups the failure count data into cells such that the number of failures in each cell is at least 5. The number of degrees of freedom is then the number of cells minus the number of parameters in the model. You can use the value of Chi-Square statistic, the number of degrees of freedom, and a table of percentage points for the Chi-Square distribution to determine the significance level, if any, at which the model estimates fit the data.

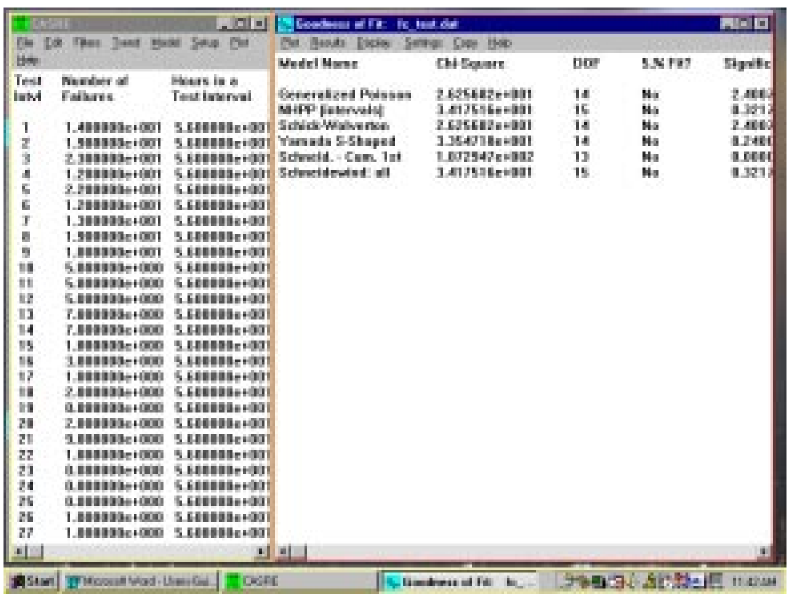

Figure 59 - Model evaluation statistics - goodness of fit display for failure counts models

.

3. The fourth column indicates whether the model results fit the data at a significance level specified by the user (see section 4.14 for details on specifying a significance level).

4. The fifth column gives the percentage point for the Chi-Square statistic given in the first column. For instance, the percentage value for the Generalized Poisson model is approximately 2.4%, meaning that the model does not fit at the 5% significance level.

The ability to see goodness-of-fit, prequential likelihood, relative accuracy, model bias, model bias trend, bias scatter plots, and model noise displays depends on the type of failure data (failure counts or time between failures) and the parameter estimation used. Table 4 below lists the conditions under which these statistics can be seen.

Data type / **Parameter estimation**	**Failure counts**	**Time between failures**
Maximum likelihood	Prequential likelihood, Relative accuracy, Chi-Square test statistic	Prequential likelihood, Relative accuracy, Model bias, Model bias trend, Bias scatter plot, Model noise, Kolmogorov-Smirnov test statistic
Least squares	Chi-Square test statistic	Kolmogorov-Smirnov test statistic

Table 4 – Model Applicability Criteria for TBF and FC Models

Finally, the models can be ranked with respect to the following criteria:

1. Prequential likelihood
2. Model bias
3. Bias trend
4. Model noise
5. Goodness-of-fit

To rank the models, first choose the "Model rankings" item in the graphic display window's "Display" menu. This brings up a submenu giving you two choices of how to display model rankings. The first item in the submenu, "Rank summary" will show the overall rank for each model as well as the current estimate for the software's reliability. The second submenu item, "Ranking details", displays the overall rank of each model as well as the rank of each model with respect to each of the ranking criteria. Regardless of which display you choose, you'll first see a dialog box which prompts you to specify the order in which the ranking criteria will be applied, as well as weights for each of the ranking criteria. This dialog box is shown in Figure 60 below. In this dialog box, you can assign floating point values between 0 and 1 to each criterion. The higher the value of the weight, the more importance placed on that criterion. If there are criteria that you don't want to consider in the ranking, give them a weight of 0. You can also assign the order in which the criteria are applied in ranking the models. This order is specified by the integer-valued entries you type into the “Ranking Priority” edit windows of the dialog box. Assigning a ranking priority of “1” to a criterion means that it will be the first one used in ranking the models. If a tie occurs using that criterion, the tie will be broken using the criterion to which you've a ranking priority of "2", and so forth. You can assign the same ranking priority to more than one criterion. Once you've assigned ranking order and weights to each criterion, selecting the "OK" button on the dialog box will produce either the rank summary or ranking detail display, depending upon the menu item that was initially selected.

By default, the goodness-of-fit criterion is used as a first stage screening. This means that before any other of the ranking criteria are applied, a test is done to determine whether the model results fit the data at a specified significance level (remember that for failure counts data, you can specify the significance level. For time between failures data, the goodness of fit test determines whether the results fit the data at the 5% significance level). If the model results fit at the specified significance level, the remaining criteria are used to rank the models. No further ranking is done for models where the results do not fit at the specified significance level. In Figure 60, the ranking criteria are set up so that prequential likelihood would be used to rank the models after application of the initial goodness-of-fit screening. If any models would have the same prequential likelihood value, that tie would be broken by comparing the model bias values. Model bias ties would be broken by looking at bias trend values, and ties in bias trend values would be broken by looking at model noise. This ranking method is described in [Niko95].

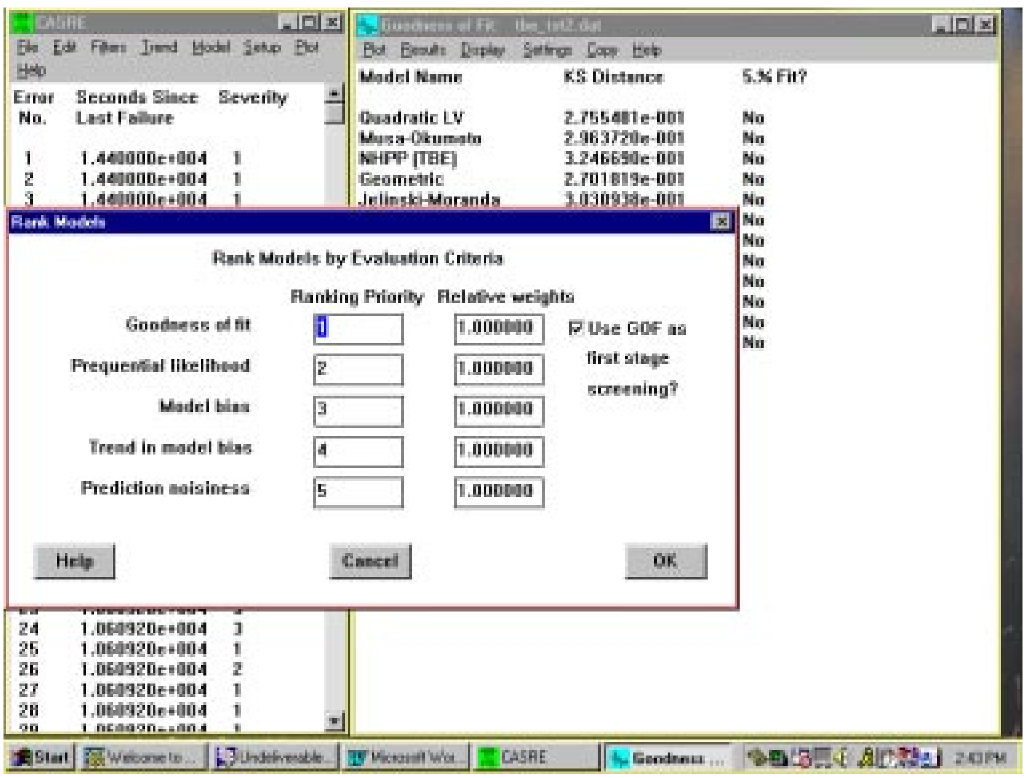

Figure 60 - Model evaluation statistics - setting ranking priority and weights for model ranking criteria

The summary of the model rankings, shown in Figure 61 below, shows the following for each model that was run:

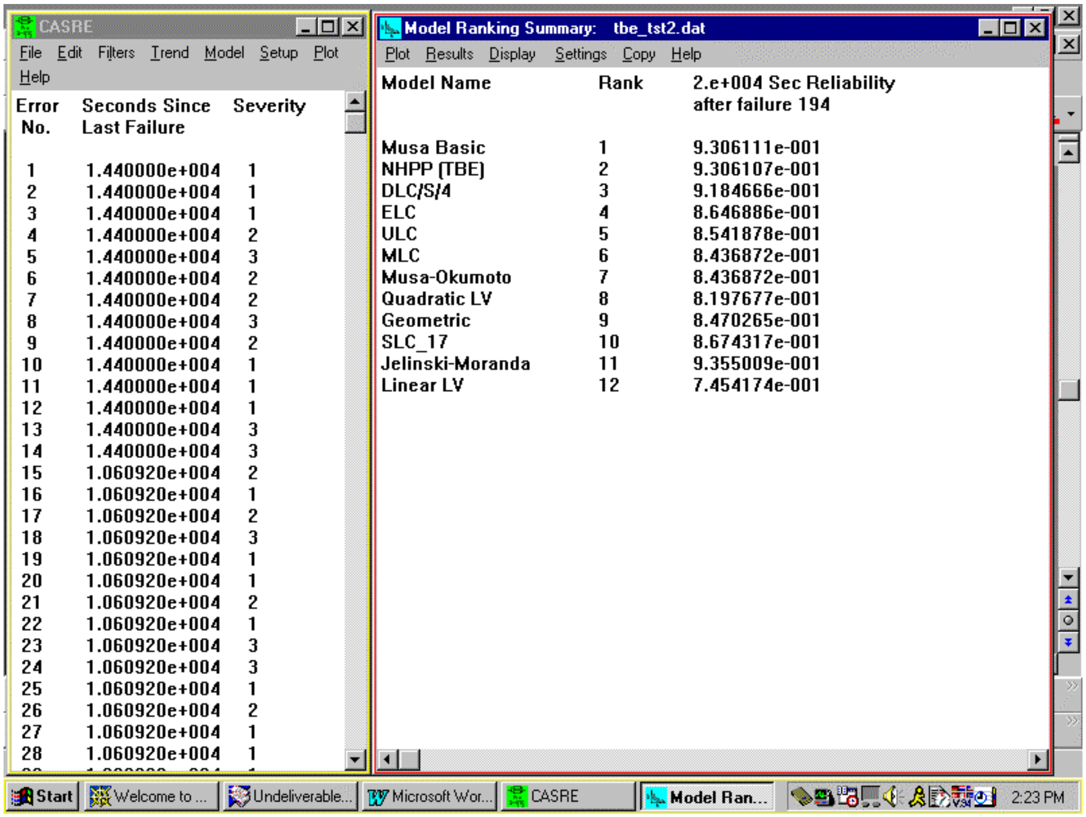

Figure 61 - Model evaluation statistics - model ranking summary display

1. The name of the model.
2. The overall rank of the model.
3. The current reliability of the software, as predicted by the model. For our time between failures example, recall that we're looking at failure numbers 150-194, and that we make reliability predictions for a time interval 20,000 seconds long. After having analyzed all of the data points, a model then makes an estimate of the current hazard rate. If we call this estimated hazard rate ***h(x)***, where ***x*** represents the cumulative elapsed testing time, the predicted reliability for a 20,000 unit time period is:

$$\exp\left(-\int_0^{20000} h(x)dx\right).$$

This is the value found in the third column of the rank summary display.

The second ranking display, shown in Figures 62 and 63 below, gives the model ranking details, and displays the following items for each model that was run:

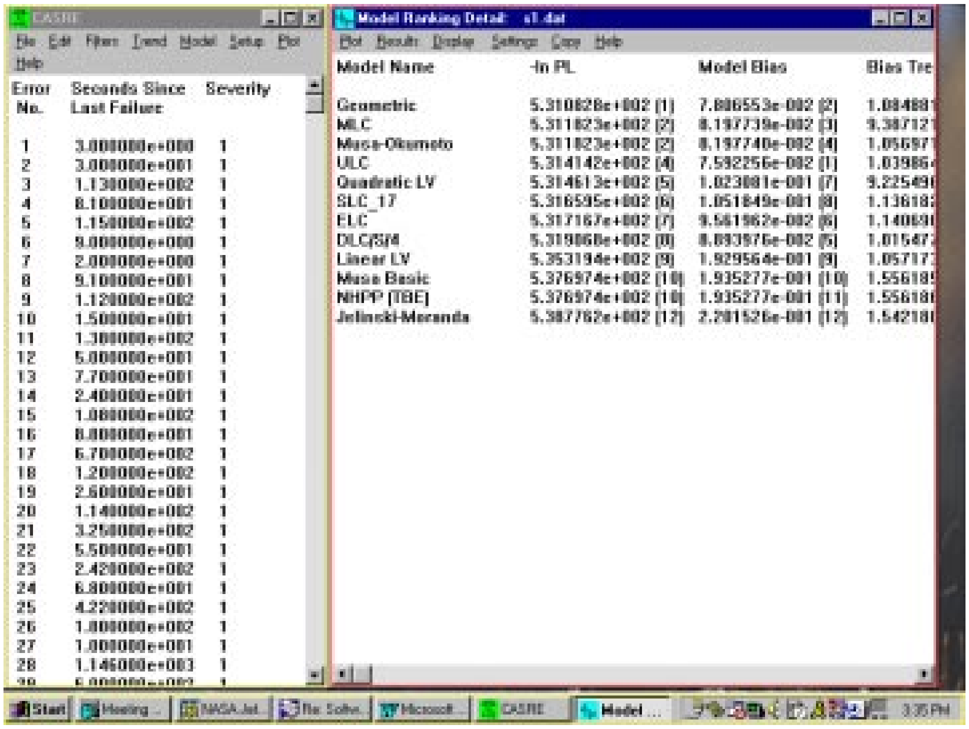

Figure 62 - Model evaluation statistics - model ranking details (leftmost columns of table)

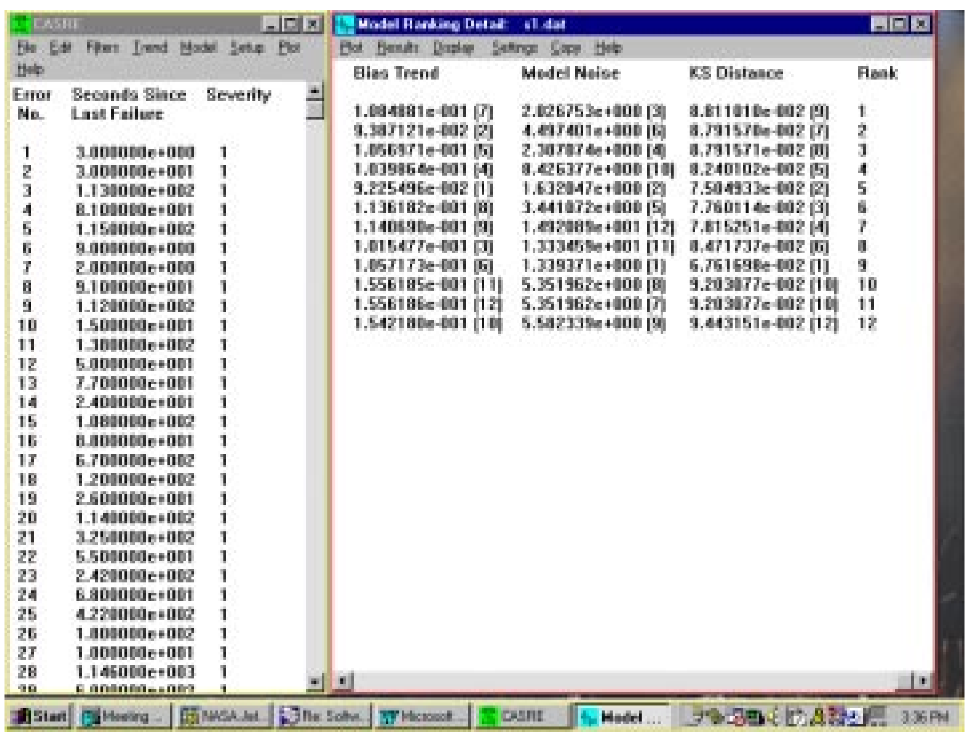

Figure 63 - Model evaluation statistics - model ranking details (rightmost columns of table)

1. The name of the model.

2. Prequential likelihood and rank for PL criterion (seen for maximum likelihood parameter estimation only).

3. Model bias value and rank for the bias criterion (seen for maximum likelihood parameter estimation and time between failures data only).

4. Model bias trend and rank (seen for maximum likelihood parameter estimation and time between failures data only).

5. Model noise and rank (seen for maximum likelihood parameter estimation and time between failures data only).

6. Goodness-of-fit value and rank (Chi-Square statistic for failure counts data, and Kolmogorov-Smirnov statistic for time between failures data).

7. Overall model rank.

Note that the time between failures set shown in Figures 62 and 63 is different from that shown in Figure 61. This is because the model results for the data set shown in Figure 61 do not fit the real data at the 5% significance level. Figure 64 below shows the “Ranking details” display for the data set used in Figure 61 if the ranking scheme is the same as shown in Figure 60. Since none of the models fit, the message “Model did not fit data at specified significance level.” is displayed for each model.

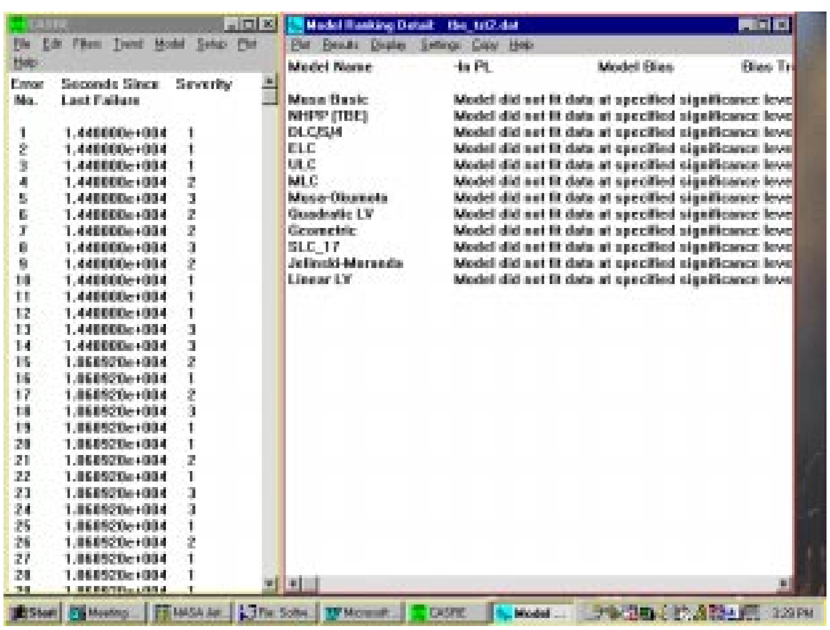

Figure 64 - Ranking detail display when model results do not fit failure data

If you look at other failure data sets, you may find that some model results fit the actual data at the specified significance level, while others do not. In this case, you'd get model rankings for those models whose results fit the data at the specified significance level. For those models whose results don't fit the data, you'd get the message shown in Figure 64.

The dialog box shown in Figure 60 above shows the default ranking priority and criteria weighting values for ranking models. Also, the "Use GOF as first stage screening?" checkbox in the upper right-hand corner of the dialog box is checked by default. If you decide not to use the goodness of fit values as a screening filter, uncheck this checkbox, and goodness of fit will be treated the same as the other ranking criteria.

Ordinarily, you won't be changing the default values for model ranking criteria priorities and weights. The only change you might make to the dialog box is to uncheck the "Use GOF as first stage screening?" checkbox if you find that the model results don't fit the data at the specified significance level. This way, you'll get numerical values for each model criterion and an overall model ranking instead of the "Model did not fit as specified significance level." message. Bear in mind, however, that if the results do not fit the actual failure data very well, the reliability estimates and predictions you make may not be a very good match with what you actually observe. If you decide to change the default values shown in Figure 60, here are some guidelines to help you assign weights to the model ranking criteria:

1. For time between failures data using maximum likelihood parameter estimation, assign weights of 1 to all of the criteria except for model noise, to which you can assign a weight of 0. Recall that model noise may not be as good an indicator of the most appropriate model as the other criteria.

2. For failure counts data using maximum likelihood parameter estimation, assign weights of 1 to the goodness of fit and prequential likelihood criteria. These are the only ranking criteria you'll be able to use in this situation. The edit windows for the other three criteria will be disabled and shown in gray for this situation.

3. For time between failures data and failure counts data using least squares parameter estimation, you can only rank the models with respect to goodness of fit. The edit windows for the other four ranking criteria will be disabled and shown in gray.

4.9.3. Scaling the Plot and Other Drawing Controls

CASRE lets you zoom in on a specific portion of the graphic display window. For instance, it is difficult to distinguish individual points in the model result displays shown in some of the earlier figures of this user's guide. Moreover, if the results of several models are displayed simultaneously, the predictions made by those models may be difficult to tell apart. In this case, it is desirable to be able to view a specific area of the plot in greater detail. The "Scale axes" item of the graphic display window's "Scale" menu let you take a more detailed look at a specific portion of the plotted model results.

To illustrate how to scale a plot, return for a moment to Figure 45, which shows a somewhat cluttered plot of model results. To start scaling, choose the "Scale axes" in the graphic display window's "Settings" menu. Doing this brings up the dialog box shown in Figure 65 below. The rectangular area specified by completing the dialog box's four editing fields will be plotted to cover the entire display area after you select and click the "OK" button. All values for x and y must be greater than or equal to 0. The edit windows in Figure 60 specify the portion of the plot in Figure 45 that will be enlarged. Figure 66 on the next page shows the resulting enlargement of Figure 45. In Figure 61, seen on the next page, notice that some of the symbols representing different model results appear above the horizontal line drawn at y=200,000. This means that those results are beyond the scaling limits that we've set in Figure 65.

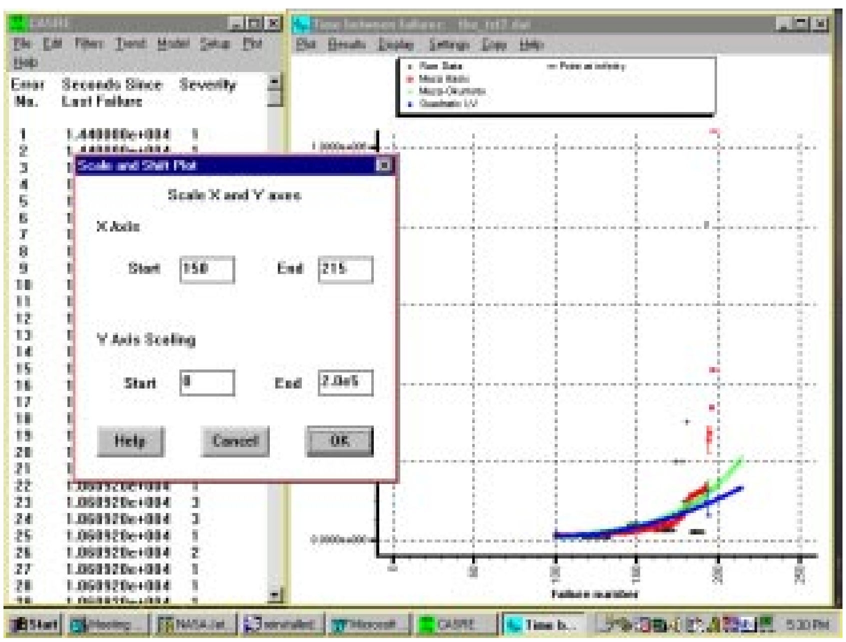

Figure 65 - Scaling a plot - specifying area of plot to be enlarged

You can also scale a plot by clicking and dragging on the plot to form a rectangle. The area within the rectangle will then be scaled to occupy the entire graphic display window. Any fine tuning can be done by using the dialog box described above. To scale the plot this way, do the following:

1. Click the mouse on the plot. The cursor will change to a crosshair.
2. Move the crosshair to one of the corners of the rectangle you want to expand. Keep the left mouse button down while dragging the crosshair to the diagonally opposite corner of the rectangle. While you're dragging the crosshair, the area inside the rectangle will alternate between normal video and inverse video to help you keep track of the rectangle you're defining.
3. Release the left mouse button when you've finished defining the area of the plot you want to expand.

Whichever way you choose to rescale a plot, one it has been rescaled, scroll bars will appear as shown in Figure 66. You can use these controls to scroll back and forth, up and down, through the entire plot. The scroll bars remain with the plot until the "Show entire plot" option in the "Settings" menu is selected.

Once a plot has been rescaled, the scaling settings remain with that particular plot until one of the following occurs:

1. The "Scale axes" option is used on the same plot to change the scale settings.
2. The "Show entire plot" option in the "Settings" menu is used to remove the scale settings for the plot.
3. A new file is opened, at which time all scaling options are reset.

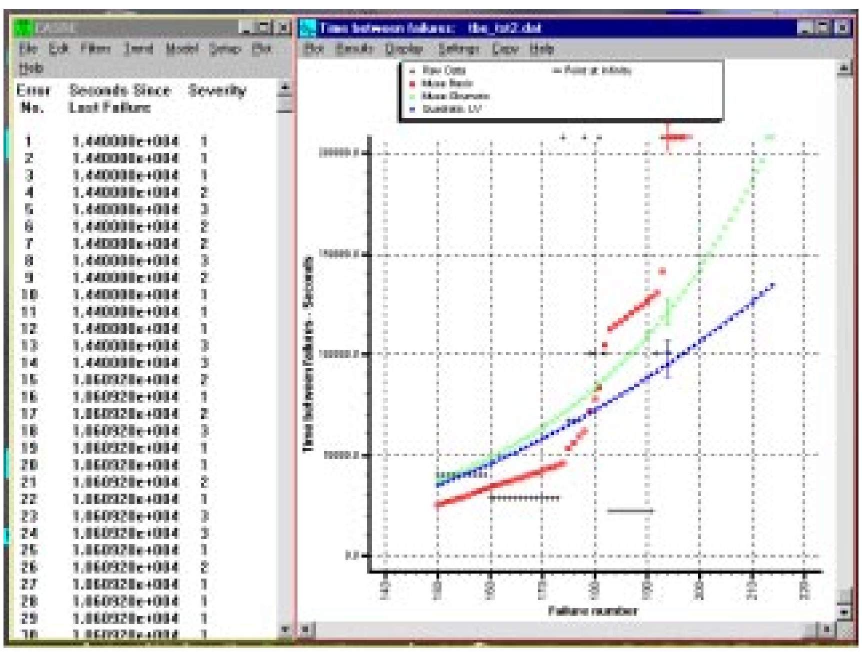

Figure 66 - Scaled plot display

The "Show entire plot" option gives you two choices. You can reset the scale settings for the plot that is currently displayed, or you can reset the scale settings for all plots. Suppose, for instance, that the time between failures plot shown in Figure 45 is currently being displayed. Since the plot is rather cluttered, you rescale the plot to bring out detail in a particular area shown in Figure 66. You then change the display to look at relative accuracy, where you also rescale the plot to zoom in on a particular location. If you then return to the time between failures plot, the scale settings you made before switching to the relative accuracy plot will still be in place, and you'll be looking at the portion of the plot which you'd previously highlighted, as shown in Figure 66.

If you now choose the "Current plot" option of the "Show entire plot" submenu, you'll lose the scale settings for the cumulative number of failures plot. Instead of seeing a magnified portion of the plot, you'll see the plot in its entirety, just as it was before you did any scaling on it. For our example, this will return the plot shown in Figure 45 to the graphic display window. If you now switch to the relative accuracy plot, you'll still be zoomed-in on the particular area of that plot that you highlighted by using the "Scale axes" option.

Suppose you had chosen the "All plots" option of the "Show entire plot" submenu instead of the "Current plot" option. No matter which plot you showed, you'd see the entire plot, regardless of whether any of them had been rescaled prior to selecting the "All plots" option.

Using other options in the graphic display window's "Settings" menu, you have additional control over the way in which failure data and model results are displayed. For instance, you can choose whether to draw raw failure data only, model results only, or both failure data and model results in the graphic display window. To make this choice, select the "Draw data only", "Draw results only", or "Draw data and results" option in the "Settings" menu. After you make your choice, a check mark will appear in the menu next to the choice you've made. This setting will remain in effect until the next time you change it. Figures 67 and 68 below show how Figure 66 would change if we were to choose the "Draw data only" and the "Draw results only" menu items. For Figure 67, we've chosen the "Draw data only" menu item, and we see that only the raw failure data is shown. For Figure 68, we've chosen to display only the model results, and we see that the raw failure data no longer appears on the plot. This capability can be useful if you want to concentrate only on the model results, for instance, or if the display is getting somewhat cluttered.

When using failure counts data, you can make a choice of how to label the x-axis. When you first open a set of failure counts data, the initial display shows failure counts on the y-axis and the test interval number on the x-axis. You might want to change this display so that instead of test intervals, the x-axis shows the total amount of time elapsed from the start of test. This might produce more useful failure counts, cumulative number of failures, and reliability growth displays with which to predict future behavior. You can change this by choosing the "Elapsed time" option in the "Settings" menu. Once you've made this choice, a check mark will appear next to the "Elapsed time" menu item. To change the display back to showing test intervals on the x-axis, choose the "Test interval number" item in the "Settings" menu. As an example, look at the failure counts data originally shown in Figure 5. The x axis in this figure is labelled in terms of test intervals. Figure 69 below shows what happens to Figure 5 if you choose the "Elapsed time" item in the "Settings" menu. Since all test intervals are the same length, the shape of the failure counts curve remains the same, but the x axis is relabelled to show the elapsed time since the start of testing. This option is available only for failure counts data.

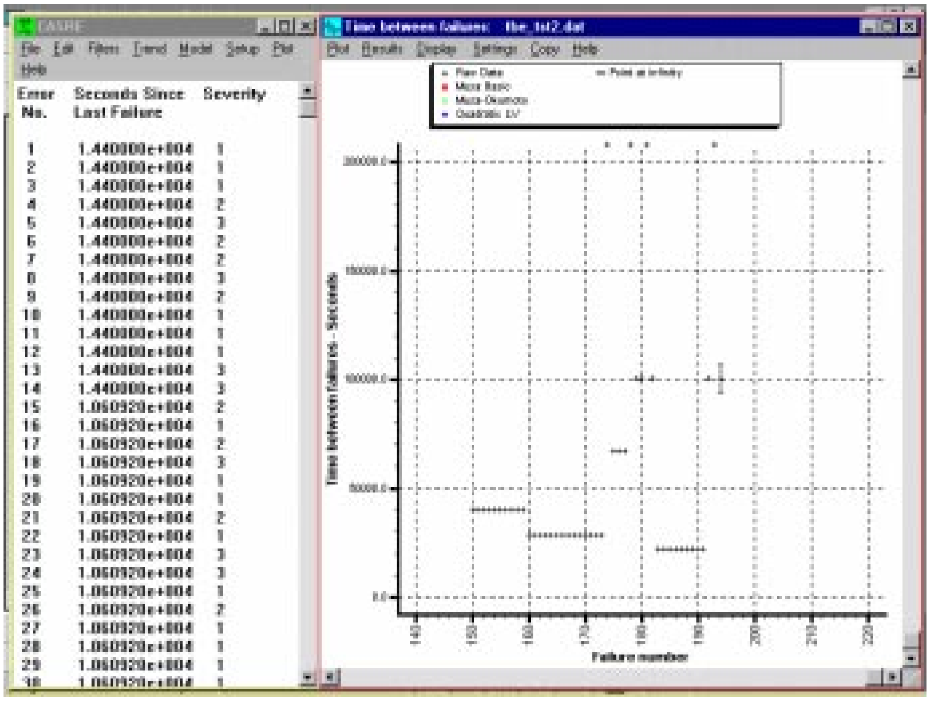

Figure 67 - Model results display - setting the plot to display only the raw failure data

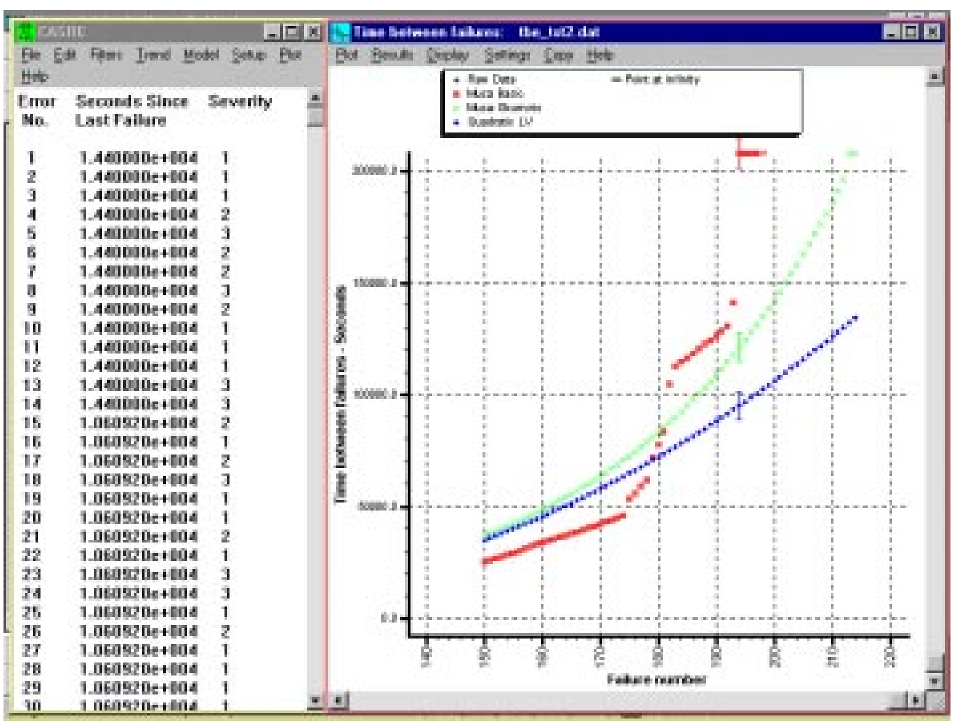

Figure 68 - Model results display - setting the plot to show only model results

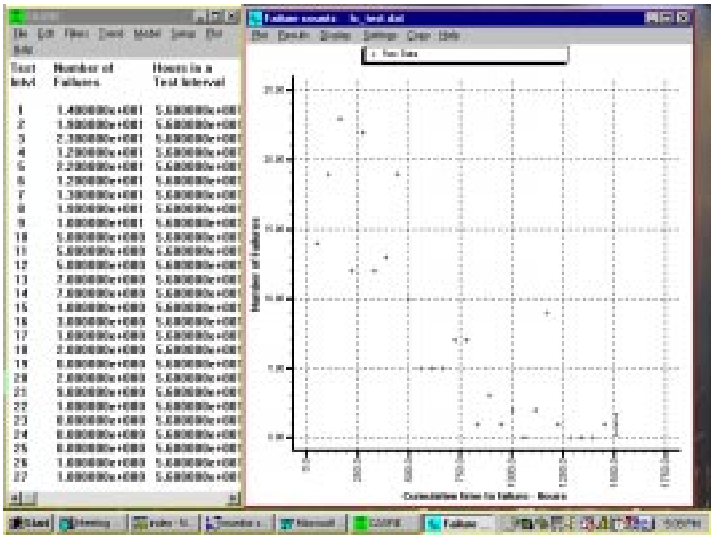

Figure 69 - Changing the labelling of the x axis for failure counts data

You can also choose the way in which model predictions are shown. After you choose one or more sets of model results, they are drawn in the graphic display window as scatter points. Instead of drawing scatter points, you may want to draw lines through the points that predict how the software will behave in the future. To do this, choose the "Line plot" item in the "Settings" menu. Once you've made this choice, a check mark will appear next to that menu item. Figure 70, shown below, shows how this setting works. Figure 70 is identical to Figure 66 except that this particular setting has been changed, In Figure 70, you can see that the predictions for failures 195-214 are indicated by a line drawn through the model predictions rather than by scatter points. You can change back to the default way of drawing model results by selecting the "Scatter point" option in the "Settings" menu.

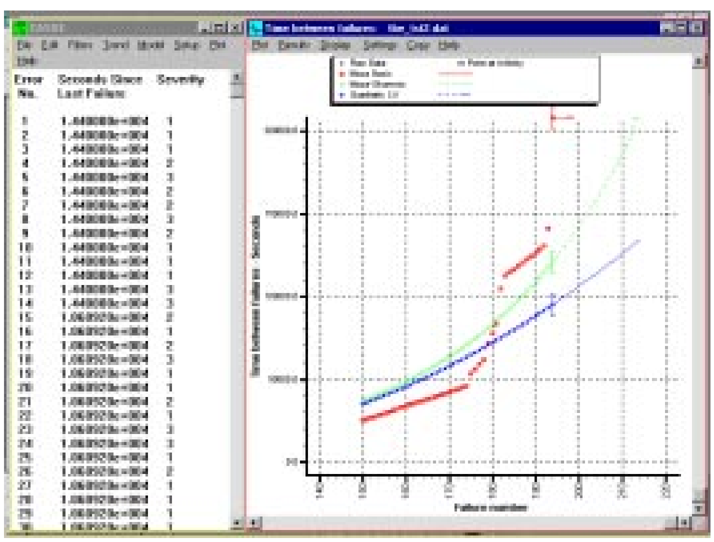

Figure 70 - Displaying model predictions as a line plot

You can also specify the colors in which the raw data and each set of model results is drawn. In all of the figures we have seen so far, the raw data and model results have been drawn in the default colors: raw data is drawn in black, the first set of model results in red, the second set of results in green, and the third set in blue. However, you can change these settings by selecting the "Choose colors" item in the "Settings" menu. This will bring up the "Set Up Colors" dialog box shown in Figure 71 below.

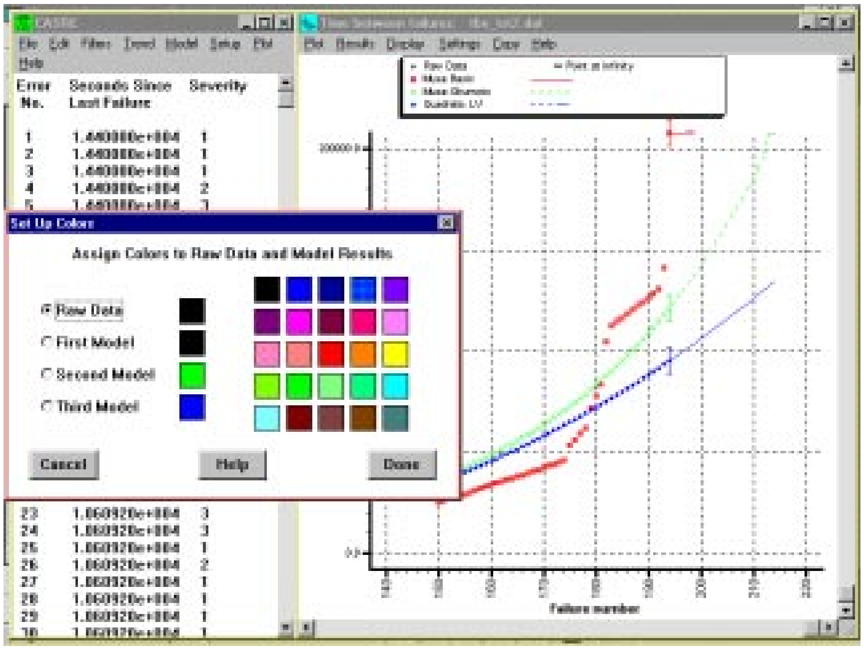

Figure 71 - Specifying display colors

On the right side of the dialog box are 25 buttons representing colors that you can use to draw the raw data and model results. On the left side of the dialog box are four radio buttons that you use to specify the data or model results whose color you're selecting. For instance, to change the color in which the raw data is drawn, you would select the "Raw data" radio button with the mouse, and then click the mouse on one of the 25 colors on the right side of the dialog box. The square just to the right of the "Raw data" radio button will change color to match the one you've just selected. After clicking on the "Done" button, the raw data will then be drawn in the selected color. The colors in which model results are drawn are changed in the same way. Let's use Figure 70 as an example. Suppose you want to change the color in which the Quadratic Littlewood-Verrall model results are drawn. You'd just bring up the "Set Up Colors" dialog box, click on the "First Model" radio button, then select the desired color from the 25 available colors. After pressing the "Done" button, the Quadratic Littlewood-Verrall model results will be drawn in the specified color.

The "Settings" menu contains two other options: "Draw in color" and "Draw in B and W". If you select the "Draw in color" item, the contents of the graphic display window will be drawn in the colors you've selected using the "Choose colors" item. This is the default setting. If you select "Draw in B and W", the contents of the graphic display window will be drawn in black and white. This option is useful when printing on a black and white printer. Since CASRE tries to print exactly what appears in the graphic display window, it will try to print color on any printer you've selected if you've chosen to draw in color. Depending on the printer you've chosen, colors may not show up as well as black and white.

Finally, you can make multiple copies of the graphic display window by selecting the "Copy plot" item in the "Copy" menu. This is useful if you want to see multiple views of the same data set or set of model results. For instance, in one of the windows, you can show the data as time between failures, while in another window, you can show the data in the form of cumulative failures. If you've run models on the data, you can display one set of model results in one window, and another set in a second window. This is shown in Figure 72 below. The colors for each window can also be separately specified, and each window can be separately scaled.

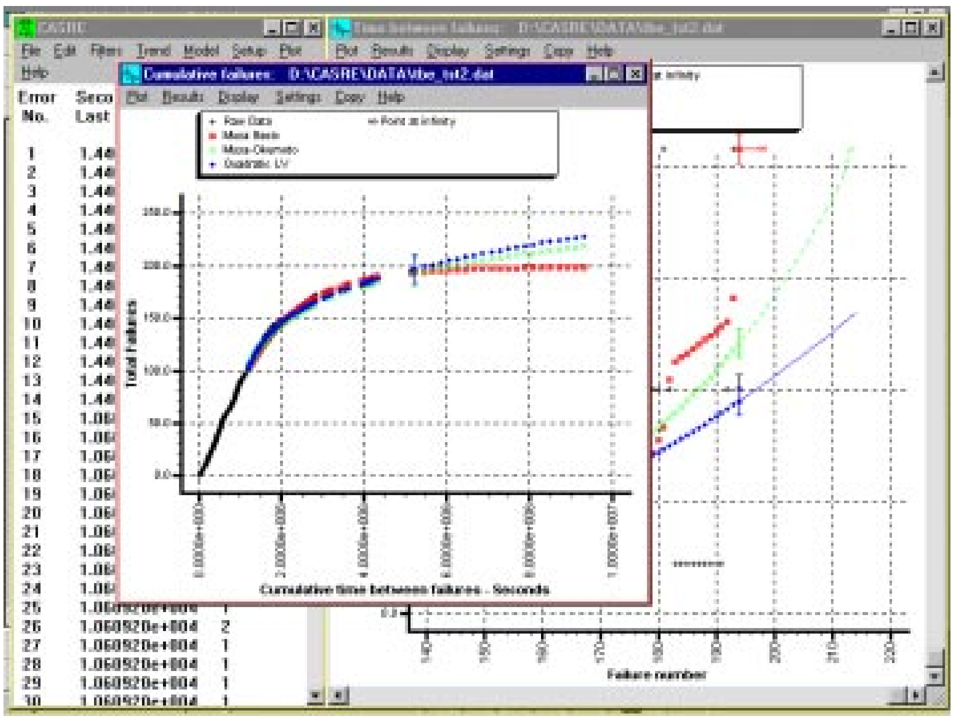

Figure 72 - Making copies of the graphic display window

When you make a copy of the graphic display window, the copy is offset from the original plot to let you see how many copies you've made. The number of copies of the graphic display window that you can make varies depending on the type of failure data you're using and whether you've run any models. The maximum number can range from 8 to 14. If you try making more than the maximum number of copies, an error message will appear to tell you that you can't make any more copies.

4.10. The Model Results Table

There are times at which you may want to see model results displayed in tabular form rather than high resolution plots. CASRE lets you do this with the "Model results table" in the graphic display window's "Results" menu. In this table, you can scroll across the table's columns, or switch back and forth between different tables of model results. These operations are discussed in the following paragraphs.

4.10.1. Viewing Model Results Tables

Once you've run one or more sets of model results, you can bring up a window in which the results are shown in tabular form. You'll use the "Model results table" item in the graphic display window's "Results" menu. The resulting display is shown in Figures 73-81 on the following pages. Figures 73-77 show the table for time between failures data, while Figures 78-81 show the table for failure counts data.

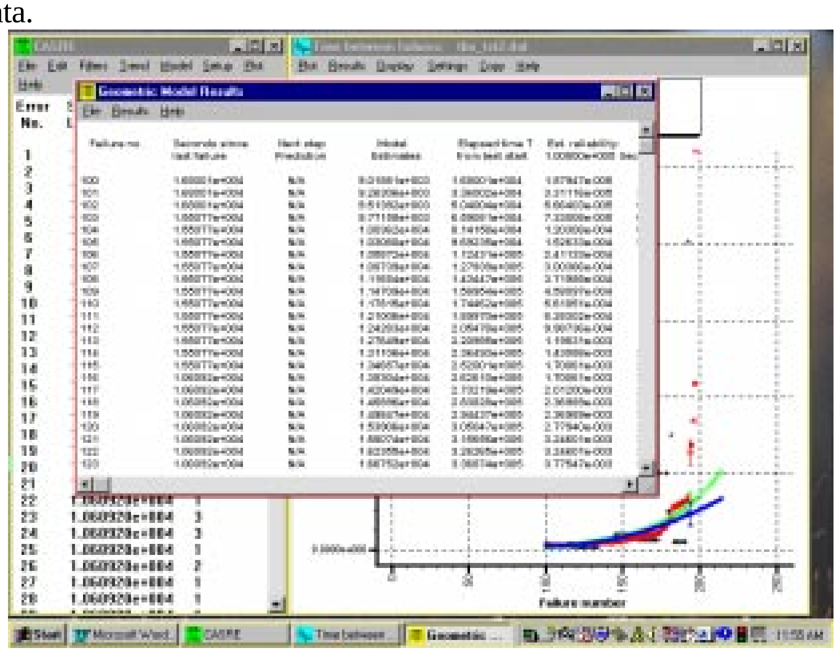

Figure 73 - Model results table for time between failures models - leftmost columns

In Figures 73-76, we continue the example we've been using in previous figures. Recall that we've selected as our data range failures 100-194, with the initial parameter estimation range from failures 100-150. We've also specified that we'll be making predictions of times between failures for the 20 failures that will be seen after failure 194. The specific model results displayed in Figures 73-76 are for the Geometric model. The values reported in the columns of the tabular display for time between failures data are as follows:

1. **Failure number** - this gives the actual times between subsequent failures as well as the predicted times between failures. In our example, the first failure number will

be 100, since the selected data range starts at 100, and the last will be 214 (the end of the selected data range, 194, plus the 20 failures in the future for which we predict time to failure).

2. **Time since last failure** - for the failures that belong to the actual data, this column reports the actual time elapsed since the last failure. In our example, there will be numerical values for failures 100-194. For failures 195-214, this column will have values of "N/A", since there is no actual time between failures data beyond failure number 194.

3. **Next step prediction** - this column reports the model's estimate of the time to the next failure, based on the model's parameters computed using the data between the model start point and the current failure number. If the current failure number is less than or equal to the observation selected as the end of the initial parameter estimation range, this column will have a value of "N/A". In our example, the initial parameter estimation range was specified to be failures 100-150. This means that for every failure between and including 100 and 150, this column will have the value "N/A". For failure 151, this column will give the model's estimate of the time between failures 151 and 152, based on the parameter estimates computed with the times between failures for failures 100-151. For failure 152, this column will give the model's estimate of the time between failures 152 and 153, based on the parameter estimates computed using failures 100-152. Since the end of the data range is at failure 194, the values of this column for failures 195 and beyond will be "N/A".

   A value of "UNAVAILABLE" means that the prediction could not be made, most likely because the model's parameter estimates did not converge.

4. **Estimates and Predictions** - for failure **i**, where **i** is less than or equal to the number of the last failure observed, this column gives the model's estimate of the amount of time elapsed since failure **i-1**. This value is estimated using all of the failure data in the data range (failures 100-194 in our time between failures example). For failures between the model start and model end points (failures 100-194 in our example), you can think of this as a curve fitting exercise in which the model produces the estimates that best fit the actual observations. In our example, for instance, the value for failure 100 is the estimated time between failure 99 and failure 100, and is computed using the actual observations between and including 100 and 194.

   This column also contains predictions for time between future failures. In our running example, recall that we've specified that predictions of time between failures will be made for the next 20 failures after the end of the data range; i.e., time between failure predictions will be made for failures 195-214. Each of these predictions is based on model parameter estimates made using the entire data range (failures 100-194). Note that these predictions are similar to the next step predictions described above. The prediction for failure 195 is a one step ahead (next step) prediction based on the parameters calculated using observations 100-194, the prediction for failure 196 is a two step ahead prediction based on the same parameters, and so forth.

There are models that assume a finite number of failures are in the software. In our example, for instance, the Jelinski-Moranda model predicts that there are a total of 199 failures in the software, based on computations using failures 100-194. In this case, the model predicts that there is an infinite amount of time required to detect the last failure; this would be recorded as "INFINITY" for failure 199 (see Figure 77). The model estimates for failures 200-214 would also be INFINITY.

A value of "UNAVAILABLE" means that the prediction could not be made, most likely because the model's parameter estimates did not converge.

5. **Elapsed time T from test start** - this is the cumulative time elapsed since the first failure in the selected data range. In our example, the value for failure 100 would be the total elapsed time from failure 99. The value for failure 194 would be the total elapsed time between failures 99 and 194.

   For predictions into the future (i.e. failures after failure 194), the elapsed time is computed as follow:

   a. Find the total elapsed time for the actual data - in this case, the total elapsed time between failures 99 and 194.
   b. Take 5% of the value computed in (a) and call it **T**.
   c. For each point after the last failure in the actual data set, increment the total elapsed time by **T**. For instance, the elapsed time for point 195 would be the elapsed time for failure 194 plus an additional 5%. The elapsed time for point 196 would be the elapsed time for failure 194 plus an additional 10%, and so forth.

   Recall from our discussion in paragraph 4.9.1 that elapsed time is computed in this manner to produce consistent displays and make the computation of combination model results as easy as possible.

6. **Estimated reliability** - this column gives the estimated reliability based on the model parameter estimates at the end of the data set and the elapsed time **T** shown in column 5. In our example, the reliability for each failure would be computed using the model's parameter estimates using failures 100-194. The interval for which the reliability is computed is specified as described in section 4.3.

   A value of "UNAVAILABLE" means that the prediction could not be made, most likely because the model's parameter estimates did not converge.

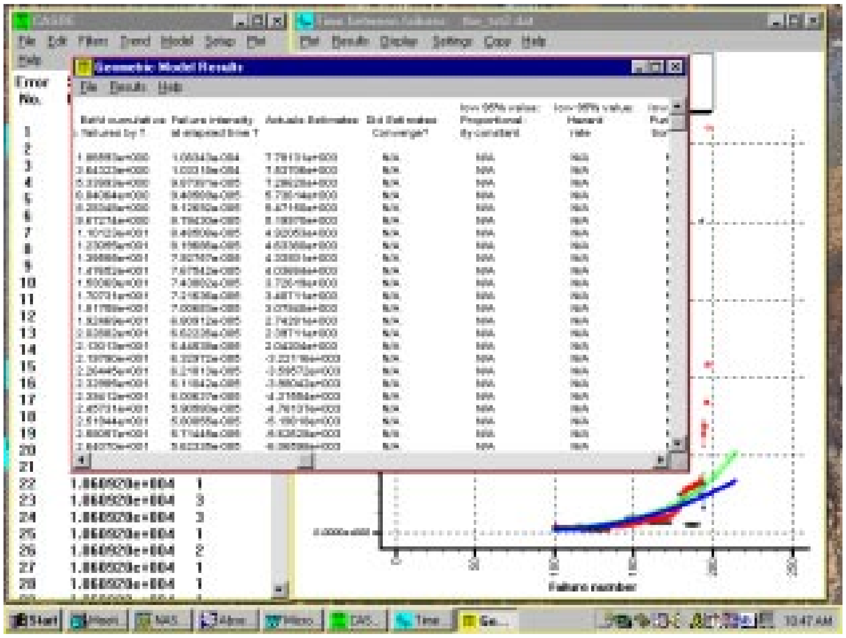

Figure 74 - Model results table for time between failures models - columns 7-12

7. **Estimated cumulative number of failures at elapsed time T** - this column gives the model's estimates and predictions of the cumulative number of failures at elapsed time **T**, shown in column 5. As with the entry for column 6, this value is computed based on the model parameter estimates at the end of the data set. This is known as the mean value function.

   A value of "UNAVAILABLE" means that the prediction could not be made, most likely because the model's parameter estimates did not converge.

8. **Failure intensity at elapsed time T** - gives the model's estimates and predictions of the failure intensity (time derivative of the estimated cumulative number of failures) at elapsed time **T**, shown in column 5. As with the value in column 6, this value is computed based on the model parameter estimates at the end of the data set.

   A value of "UNAVAILABLE" means that the prediction could not be made, most likely because the model's parameter estimates did not converge.

9. **Actuals - Estimates** - this column gives the difference between the actual observed times between failures (column 2) and the model's estimates (column 4). If the model has made an estimate of "INFINITY" as the time since the last failure, this value is also set to "INFINITY".

   A value of "UNAVAILABLE" means that the prediction could not be made, most likely because the model's parameter estimates did not converge.

10. **Did Estimates Converge** - for each failure **i**, this column specifies whether the model parameter estimates based on the failure data between the start of the selected data range and failure **i** converged. A "YES" means that the model parameter estimates are valid, while a "NO" means that valid model parameter estimates could not be obtained.

11. **Parameter 1 - low 95% confidence value**
12. **Parameter 2 - low 95% confidence value**
13. **Parameter 3 - low 95% confidence value**

For some of the models in CASRE, low and high 95% confidence values of the model parameters are available. For these models, these columns give the low 95% confidence value of the first, second, and third parameters if the selected parameter estimation method was maximum likelihood. The time between failures models for which low 95% confidence values of the parameters are available are:

a. Geometric model
b. Musa Basic model - note that 95% values are available only for the estimated total number of faults and the estimated initial mean time to failure.

A value of "UNAVAILABLE" means that this value could not be computed

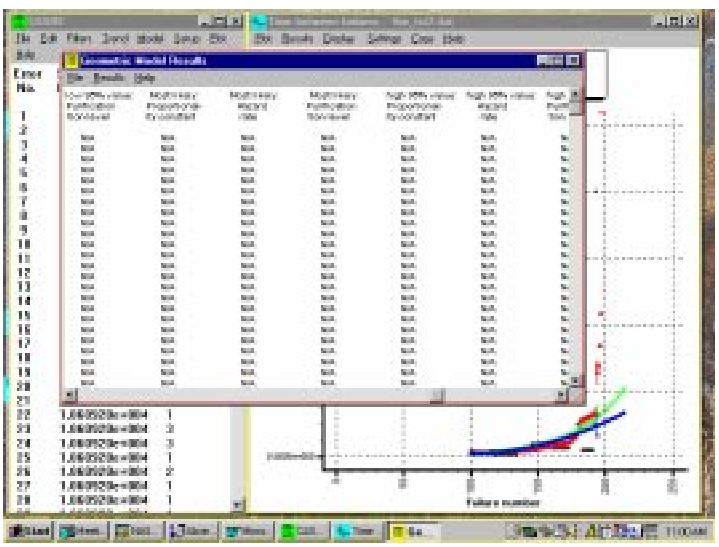

Figure 75 - Model results table for time between failures models - columns 13-18

A value of "N/A" after the end of the initial parameter estimation interval and up to the last failure in the selected data range means that this is a model for which low and high 95% confidence values of the parameters are not available, or that the least squares parameter estimation method was chosen. In our example, values of "N/A" for failures 151-194 would indicate least squares parameter estimation or a model for which high and low 95% confidence values of the parameters were not available.

All of these columns will have value of "N/A" between the first failure and the end of the initial parameter estimation interval, and after the last failure in the selected data range.

You should also note that the column headings will be specific to the models. For instance, the three parameters for the Geometric model are the proportionality constant, the hazard rate, and the purification level. The parameters for other models will have different names.

Some of the CASRE models have two parameters, while others have three. A value of "N/A" in the third parameter column after the end of the initial parameter estimation interval and up to the last failure in the selected data range means that this is a model that has only two parameters. In our example, values of "N/A" in the third parameter column for failures 151-194 would indicate a model that has only two parameters.

It should be noted that if the results of a combination model are being displayed, all model parameter estimates have the value of "N/A". This is because combinations are formed from model results; there is no additional parameter estimation required to form a combination.

14. **Parameter 1 - most likely value**
15. **Parameter 2 - most likely value**
16. **Parameter 3 - most likely value**

Depending on the parameter estimation method, these columns give either the most likely values for the model parameters (maximum likelihood estimation), or the least squares parameter estimates.

A value of "UNAVAILABLE" means that this value could not be computed.

Some of the CASRE models have two parameters, while others have three. A value of "N/A" in the third parameter column after the end of the initial parameter estimation interval and up to the last failure in the selected data range means that this is a model that has only two parameters. In our example, values of "N/A" in the third parameter column for failures 151-194 would indicate a model that has only two parameters.

All of these columns will have value of "N/A" between the first failure and the end of the initial parameter estimation interval, and after the last failure in the selected data range.

It should be noted that if the results of a combination model are being displayed, all model parameter estimates have the value of "N/A". This is because combinations are formed from model results; there is no additional parameter estimation required to form a combination.

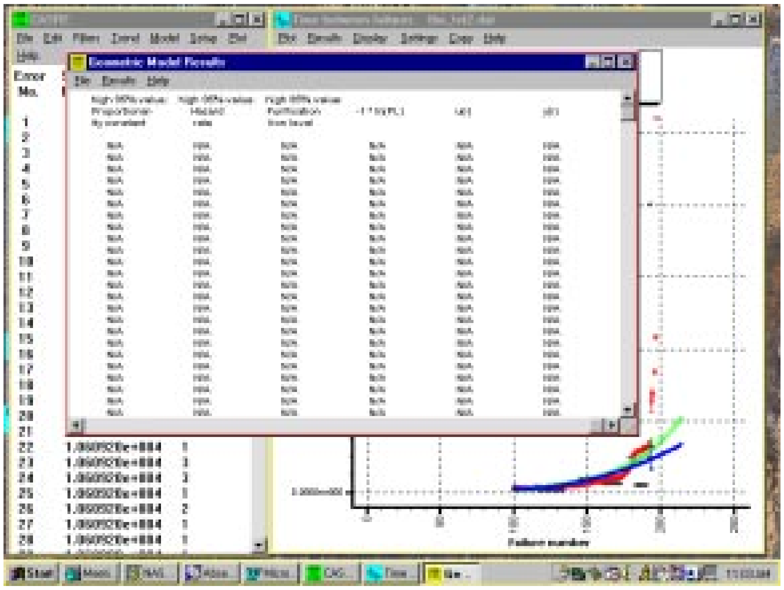

Figure 76 - Model results table for time between failures models - columns 17-22

17. **Parameter 1 - high 95% confidence value**
18. **Parameter 2 - high 95% confidence value**
19. **Parameter 3 - high 95% confidence value**

For some of the models in CASRE, low and high 95% confidence values of the model parameters are available. For these models, these columns give the high 95% confidence value of the first, second, and third parameters if the selected parameter estimation method was maximum likelihood. The time between failures models for which high 95% confidence values of the parameters are available are:

a. Geometric model
b. Musa Basic model - note that 95% values are available only for the estimated total number of faults and the estimated initial mean time to failure.

A value of "UNAVAILABLE" means that this value could not be computed.

A value of "N/A" after the end of the initial parameter estimation interval and up to the last failure in the selected data range means that this is a model for which low and high 95% confidence values of the parameters are not available, or that the least squares parameter estimation method was chosen. In our example, values of "N/A" for failures 151-194 would indicate least squares parameter estimation or a model for which high and low 95% confidence values of the parameters were not available.

All of these columns will have value of "N/A" between the first failure and the end of the initial parameter estimation interval, and after the last failure in the selected data range.

It should be noted that if the results of a combination model are being displayed, all model parameter estimates have the value of "N/A". This is because combinations are formed from model results; there is no additional parameter estimation required to form a combination.

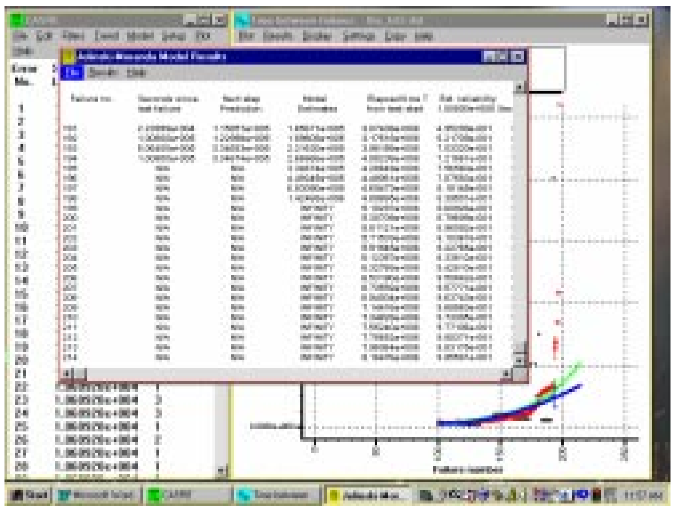

Figure 77 - Model results table - estimated remaining number of failures less than number of prediction points

20. **-ln(Prequential Likelihood)** - for each failure number **i**, where **i** is greater than the end of the initial parameter estimation range and less than the last failure in the selected data range, this column shows the value of -(natural log of the prequential likelihood) (see paragraph 4.9.2). For our example, values for the prequential likelihood would appear for failures 151-193. For all other failures, the value of this column would be "N/A".

    A value of "UNAVAILABLE" means that the calculation could not be made, most likely because the model's parameter estimates did not converge.

21. **Model Bias** - for each failure number **i**, where **i** is greater than the end of the initial parameter estimation range and less than the last failure in the selected data range, this column shows the value of the model bias (see paragraph 4.9.2). For our example, values for the model bias would appear for failures 151-193. For all other failures, the value of this column would be "N/A".

A value of "UNAVAILABLE" means that the calculation could not be made, most likely because the model's parameter estimates did not converge.

22. **Model Bias Trend** - for each failure number **i**, where **i** is greater than the end of the initial parameter estimation range and less than the last failure in the selected data range, this column shows the value of model bias trend (see paragraph 4.9.2). For our example, values for the model bias trend would appear for failures 151-193. For all other failures, the value of this column would be "N/A".

    A value of "UNAVAILABLE" means that the calculation could not be made, most likely because the model's parameter estimates did not converge.

The model results table for failure count models is shown in Figures 78-81 below. We continue with the failure counts example in which the initial parameter estimation range is test intervals 1-13, the selected data range is test intervals 1-27, and we've specified that we want to predict the number of failures that will be observed in test intervals 28-37. It is similar to that for time between failures, except that next step predictions and model estimates are expressed in terms of the number of failures estimated for this interval, or the number of failures predicted to be seen in the next interval. "Model Estimates" and "Est. reliability" for these failures will be based on the model parameters computed for failure 27 and the value computed for "Elapsed time". In addition, computation of model bias and bias trend for failure count models is not supported for this version of CASRE. The specific model results displayed in Figures 78-81 are for the NHPP model. The values reported in the columns of the tabular display for time between failures data are as follows:

1. **Test interval** - this identifies the test intervals during which failures were actually observed as well as the intervals for which predictions are made. In our example, the first test interval number will be 1, since the selected data range starts at 1, and the last will be 37 (the end of the selected data range, 27, plus the 10 intervals in the future for which we predict failure counts).

2. **Failures per test interval** - the number of failures actually observed in a test interval. In our example, there will be numerical values for test intervals 1-27. For intervals 28-37, this column will have values of "N/A", since there are no actual failure counts beyond test interval 27.

3. **Test interval length** - this column gives the length of each test interval, specifying the units of time in which this length is measured. The test interval lengths for the actual data are given, as are the lengths of the intervals for which we're predicting failure counts. In our example, actual lengths would be given for test intervals 1-27. The entries for intervals 28-37 would contain the lengths that we've specified as discussed in paragraph 4.8.3.

4. **Next step prediction** - this column reports the model's estimate of the number of failures that will be seen in the next interval, based on the model's parameters

computed using the data between the model start point and the current test interval. If the current test interval is less than or equal to the observation selected as the end of the initial parameter estimation range, this column will have a value of "N/A". In our example, the initial parameter estimation range was specified to be intervals 1-13. This means that for every interval between and including 1 and 13, this column will have the value "N/A". For interval 14, this column will give the model's estimate of the number of failures that will be seen in interval 15, based on the parameter estimates computed with the failure counts for intervals 1-14. For interval 15, this column will give the model's estimate of the number of failures for interval 16, based on the parameter estimates computed using intervals 1-15. Since the end of the data range is at interval 27, the values of this column for intervals 28 and beyond will be "N/A".

A value of "UNAVAILABLE" means that the prediction could not be made, most likely because the model's parameter estimates did not converge.

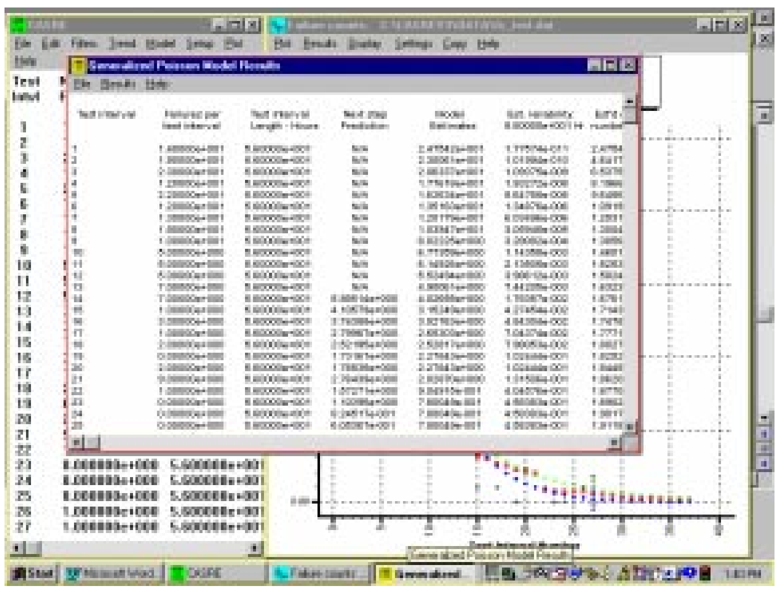

Figure 78 - Model results table for failure count models - columns 1-6 and part of column 7

5. **Estimates and Predictions** - for interval **i**, where **i** is less than or equal to the number of the last interval in which failures were observed, this column gives the model's estimate of the number of failures that will be seen in that interval. This value is estimated using all of the failure data in the data range (intervals 1-27 in our time between failures example). For failures between the model start and model end point (intervals 1-27 in our example), you can think of this as a curve fitting exercise in which the model produces the estimates that best fit the actual observations. In our example, for instance, the value for interval 1 is the estimated number of failures in that interval, and is computed using the actual observations between and including 1 and 27.

This column also contains predictions for failure counts in future test intervals. In our running example, recall that we've specified that predictions of failure counts will be made for the next 10 test intervals after the end of the data range; i.e., predictions will be made for intervals 28-37. Each of these predictions is based on model parameter estimates made using the entire data range (intervals 1-27). Note that these predictions are similar to the next step predictions described above. The prediction for interval 28 is a one step ahead (next step) prediction based on the parameters calculated using intervals 1-27, the prediction for interval 29 is a two step ahead prediction based on the same parameters, and so forth.

A value of "UNAVAILABLE" means that the prediction could not be made, most likely because the model's parameter estimates did not converge.

6. **Estimated reliability** - this column gives the estimated reliability based on the model parameter estimates at the end of the data set and the total elapsed time **T**, computed by adding the lengths of the all test intervals in the selected data range. In our example, the reliability for each interval would be computed using the model's parameter estimates using intervals 1-27. The interval for which the reliability is computed is specified as described in section 4.3.

   A value of "UNAVAILABLE" means that the prediction could not be made, most likely because the model's parameter estimates did not converge.

7. **Estimated cumulative number of failures** - for interval **i**, this column gives the model's estimates and predictions of the cumulative number of failures at interval **i**. As with the entry for column 6, this value is computed based on the model parameter estimates at the end of the data set. This is known as the mean value function.

   A value of "UNAVAILABLE" means that the prediction could not be made, most likely because the model's parameter estimates did not converge.

8. **Failure intensity at elapsed time T** - for interval **i**, this column gives the model's estimates and predictions of the failure intensity (time derivative of the estimated cumulative number of failures, item 7 above) at interval **i**. As with the value in column 6, this value is computed based on the model parameter estimates at the end of the data set.

   A value of "UNAVAILABLE" means that the prediction could not be made, most likely because the model's parameter estimates did not converge.

9. **Actuals - Estimates** - this column gives the difference between the actual observed failure counts (column 2) and the model's estimates (column 5).

A value of "UNAVAILABLE" means that the prediction could not be made, most likely because the model's parameter estimates did not converge.

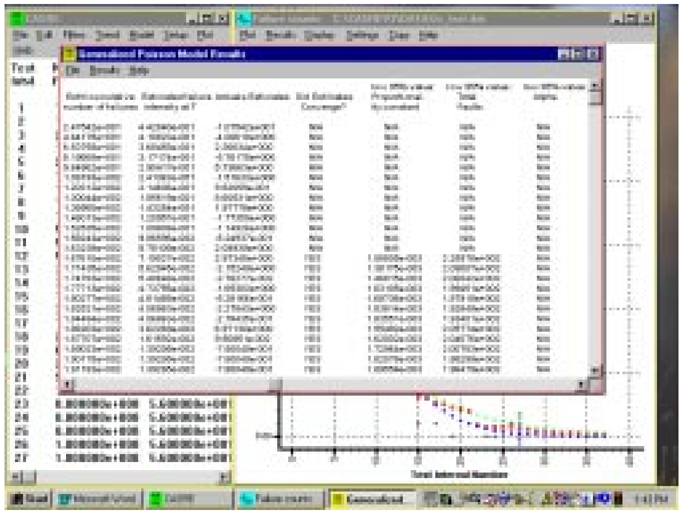

Figure 79 - Model results table for failure count models - columns 7-13

10. **Did Estimates Converge** - for each interval **i**, this column specifies whether the model parameter estimates based on the failure data between the start of the selected data range and interval **i** converged. A "YES" means that the model parameter estimates are valid, while a "NO" means that valid model parameter estimates could not be obtained.

11. **Parameter 1 - low 95% confidence value**
12. **Parameter 2 - low 95% confidence value**
13. **Parameter 3 - low 95% confidence value**

For some of the models in CASRE, low and high 95% confidence values of the model parameters are available. For these models, these columns give the low 95% confidence value of the first, second, and third parameters if the selected parameter estimation method was maximum likelihood. The failure counts models for which low 95% confidence values of the parameters are available are:

a. Generalized Poisson model
b. Nonhomogeneous Poisson Process model
c. Yamada S-shaped model

A value of "UNAVAILABLE" means that this value could not be computed.

A value of "N/A" after the end of the initial parameter estimation interval and up to the last interval in the selected data range means that this is a model for which low and high 95% confidence values of the parameters are not available, or that the least squares parameter estimation method was chosen. In our example, values of "N/A" for intervals 14-27 would indicate least squares parameter estimation or a model for which high and low 95% confidence values of the parameters were not available.

All of these columns will have value of "N/A" between the first interval and the end of the initial parameter estimation range, and after the last interval in the selected data range.

As with the time between failures models, you should also note that the column headings will be specific to the models, since each model has a different set of parameters.

Some of the CASRE models have two parameters, while others have three. A value of "N/A" in the third parameter column after the end of the initial parameter estimation range and up to the last interval in the selected data range means that this is a model that has only two parameters. In our example, values of "N/A" in the third parameter column for intervals 14-27 would indicate a model that has only two parameters.

It should be noted that if the results of a combination model are being displayed, all model parameter estimates have the value of "N/A". This is because combinations are formed from model results; there is no additional parameter estimation required to form a combination.

14. **Parameter 1 - most likely value**
15. **Parameter 2 - most likely value**
16. **Parameter 3 - most likely value**

Depending on the parameter estimation method, these columns give either the most likely values for the model parameters (maximum likelihood estimation), or the least squares parameter estimates.

A value of "UNAVAILABLE" means that this value could not be computed.

Some of the CASRE models have two parameters, while others have three. A value of "N/A" in the third parameter column after the end of the initial parameter estimation range and up to the last interval in the selected data range means that this is a model that has only two parameters. In our example, values of "N/A" in the third parameter column for intervals 14-27 would indicate a model that has only two parameters.

All of these columns will have value of "N/A" between the first interval and the end of the initial parameter estimation range and after the last interval in the selected data range.

It should be noted that if the results of a combination model are being displayed, all model parameter estimates have the value of "N/A". This is because combinations are formed from model results; there is no additional parameter estimation required to form a combination.

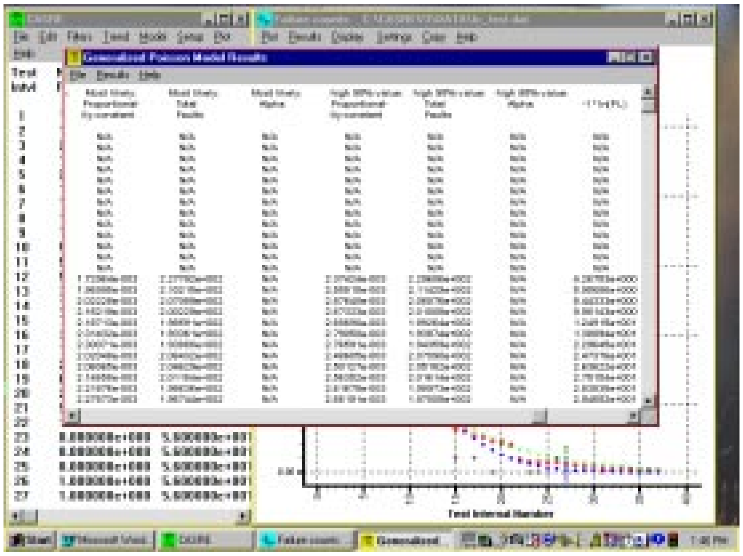

Figure 80 - Model results table for failure count models - columns 14-20

17. **Parameter 1 - high 95% confidence value**
18. **Parameter 2 - high 95% confidence value**
19. **Parameter 3 - high 95% confidence value**

For some of the models in CASRE, low and high 95% confidence values of the model parameters are available. For these models, these columns give the high 95% confidence value of the first, second, and third parameters if the selected parameter estimation method was maximum likelihood. The failure counts models for which high 95% confidence values of the parameters are available are:

a. Generalized Poisson model
b. Nonhomogeneous Poisson Process model
c. Yamada S-shaped model

A value of "UNAVAILABLE" means that this value could not be computed.

A value of "N/A" after the end of the initial parameter estimation range and up to the last interval in the selected data range means that this is a model for which low and high 95% confidence values of the parameters are not available, or that the least squares parameter estimation method was chosen. In our example, values of "N/A" for intervals 14-27 would indicate least squares parameter estimation or a model for which high and low 95% confidence values of the parameters were not available.

All of these columns will have value of "N/A" between the first interval and the end of the initial parameter estimation range, and after the last interval in the selected data range.

It should be noted that if the results of a combination model are being displayed, all model parameter estimates have the value of "N/A". This is because combinations are formed from model results; there is no additional parameter estimation required to form a combination.

20. **-ln(Prequential Likelihood)** - for each interval **i**, where **i** is greater than the end of the initial parameter estimation range and less than the last interval in the selected data range, this column shows the value of -(natural log of the prequential likelihood) (see paragraph 4.9.2). For our example, values for the prequential likelihood would appear for intervals 14-26. For all other intervals, the value of this column would be "N/A".

    A value of "UNAVAILABLE" means that the calculation could not be made, most likely because the model's parameter estimates did not converge.

21. **Model Bias**
22. **Model Bias Trend**

    These values are not available in version 3.0 of CASRE for failure counts data because of numerical representation issues, so the value of these columns is always "N/A". These values may become available in subsequent versions.

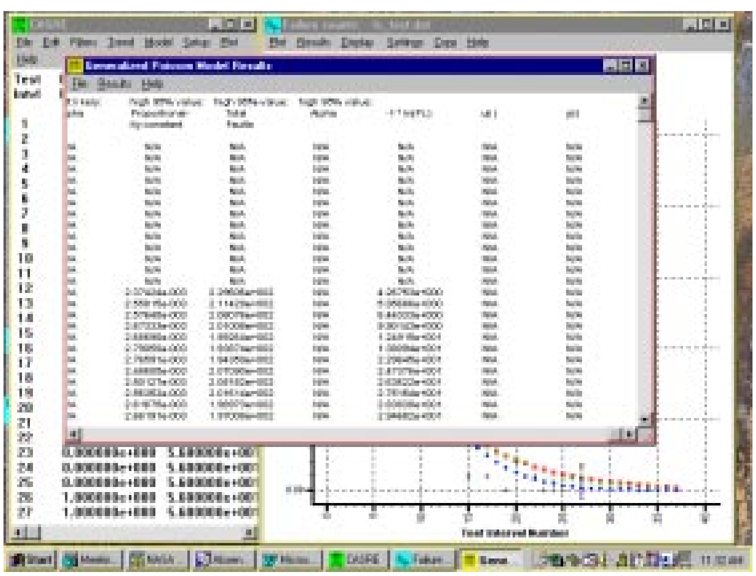

Figure 81 - Model results table for failure count models - columns 17-22

4.10.1.1. Notes on the Schneidewind Model

As noted in section 4.8.4, the Schneidewind model has three ways of treating failure data:

1. Use the failure counts from all intervals.
2. Ignore the failure counts for the first "s-1" test intervals, using only the failure counts from test interval "s" onward.
3. Use the **cumulative failure count** from the start of the data set through test interval "s-1", and then use the individual failure counts from test intervals "s" forward.

If the Schneidewind model is run with the second or third treatment types, the appearance of the results table will be somewhat different from what is described above.

Treatment Type 2 – Ignore first "s-1" Test Intervals

If the Schneidewind model is run with this treatment type, the model results table will be affected as follows:

Column 2 – The failure counts shown in this column will have a value of "0" from the start of the data set through test interval "s-1". The failure counts from test interval "s" forward will show the actual number of failures observed in each interval. For example, suppose that we have a data set in which the failure counts for intervals 1-8 are 9, 8, 9, 6, 7, 7, 5, and 6. If we choose "s" to have a value of 5, then the failure count values that will appear in the model result table are 0, 0, 0, 0, 7, 7, 5, and 6.

Column 3 - The test interval lengths shown in this column will have a value of "0" from the start of the data set through test interval "s-1". The test interval lengths from test interval "s" forward will show the actual length of each test interval.

Cols 4-22 - The values of these data fields will be "N/A" (not applicable) from the first test interval through test interval "s-1". From test interval "s" forward, the values will be as shown above.

Treatment Type 3 – Use Cumulative Failure Count for Initial Test Intervals

If the Schneidewind model is run with this treatment type, the model results table will be affected as follows:

Column 2 – The failure counts shown in this column will have a value of "0" from the start of the data set through test interval "s-2". The value appearing for test interval "s-1" will be the sum of the failure counts from the first observation through interval "s-1". The failure counts from test interval "s" forward will show the actual number of failures observed in each interval. For example, suppose that we have a data set in which the failure counts for intervals 1-8 are 9, 8, 9, 6, 7, 7, 5, and 6. If we choose "s" to have a value of 5, then the failure count values that will appear in the model result table are 0, 0, 0, 32, 7, 7, 5, and 6.

Column 3 - The test interval lengths shown in this column will have a value of "0" from the start of the data set through test interval "s-2". The value appearing for test interval "s-1" will be the sum of the test interval lengths from the first observation through interval "s-1". The test interval lengths from test interval "s" forward will show the actual length of each test interval.

Cols 4-22 - The values of these data fields will be "N/A" (not applicable) from the first test interval through test interval "s-2". From test interval "s-1" forward, valid values for these fields, which include the model's estimates of number of failures per test interval, parameter values, and model applicability criteria, will be given.

4.10.2. Selecting a Set of Results

In our time between failures example, we've run several models. We can choose which set of model results we'll display by using one of two mechanisms:

1. Choose the "Select results" item in the model result table's "Results" menu.

2. Use the "Previous model" or "Next model" items in the model result table's "Results" menu to go to the previous model in an alphabetized list of models that were run, or to go to the next model in that list. The F9 and F10 function keys are shortcut ways of choosing these menu items.

If you use the "Select results" menu item, you'll get a dialog box as shown in Figure 82 below. This dialog box contains a scrolling, alphabetized list of all the models that were run. To choose the model results that will be displayed, you can either

1. Click the mouse on the name of the model whose results you want to see, and then click on the "OK" button, or

2. Double-click the mouse on the name of the model whose results you want to see.

The selected model results will then appear in the model results table.

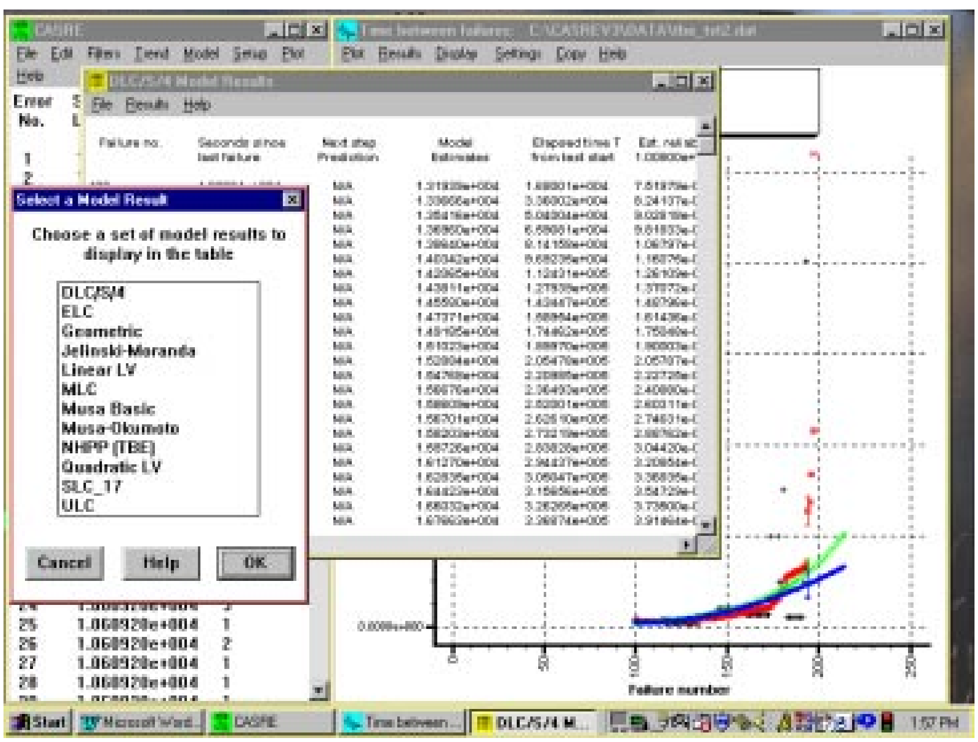

Figure 82 - Model results table - selecting a set of model results to be displayed

You can use the "Previous model" and "Next model" items in the "Results" menu to go to the previous or next set of results in an alphabetized list of models that were run. In Figure 82, for instance, the model results table contains the results of the Geometric model. Selecting the "Previous model" menu item would display the ELC model results, while selecting the "Next model" menu item would display the Jelinski-Moranda model results. You can use the F9 and F10 function keys as a shortcut way of choosing these menu items.

4.11. Printing

CASRE allows you to print the failure data, displayed in the main window, the contents of the graphic display window, and the contents of the model results table. The procedures for printing these items are very similar to each other. Note that it is not necessary to print these items on the same printer. For example, if your computer has one printer connected to the LPT1 port, another to the LPT2 port, and the third to the LPT3 port (entirely possible in a network environment), it is possible to print the failure data out on one port, the model results plot through the second port, and the model results table through the third port.

4.11.1. Printing Data Files

To print the contents of the main window, you must first choose and configure a printer. This is done by choosing the "Setup printer" menu item in the main window's "File" menu. Selecting this menu item brings up the dialog box shown in Figure 83 on the following page. The available printers are listed in the scrolling list that appears in the dialog box. To choose one of the printers in the list, use the mouse, click on the name of the printer, and then select the "OK" button. For example, Figure 83 shows the HP LaserJet Series II printer being chosen as the output device. You can use the dialog box shown in Figure 83 to select the range of data to print as well as specify how many copies of the data will be printed. By default, CASRE prints the currently selected data range used for modeling. In our running example of time between failures data, this is failures 100-194. You can choose to print the entire set of failure data, or you can specify the range of data to be printed by entering the first and last failure numbers (test interval numbers for failure counts data) in the edit windows at the bottom of the dialog box. In the edit window labelled "Number of copies", you can enter the number of copies of the failure data you'd like to have printed. The default value for the number of copies is 1. Finally, some printer drivers allow you to specify the number of copies to print. For instance, the driver for the Apple LaserWriter Pro 600 printer (whose set-up dialog is shown in Figure 84 on the following page) allows you to specify the number of copies. The two radio buttons under the label "Number of copies set in:" in Figure 83 let you choose where you'll be specifying the number of copies to print. If you choose the radio button marked "Driver", only the printer driver controls the number of copies to be printed. If, on the other hand, you choose the "CASRE" radio button, both CASRE and the printer driver control the number of copies to be printed. Suppose you set the number of copies field in Figure 83's dialog box to 2, and the number of copies field in Figure 84's dialog box to 3. Then, if you select the "CASRE" radio button in the dialog box shown in Figure 83, you'll produce a total of 6 copies.

The printer configuration can be changed by selecting a printer from the list of printers, and then choosing the "Setup" button in the dialog box. For instance, you may want to change the print orientation from portrait to landscape, or you may want to change the print resolution from 600 dots per inch (dpi) to 300 dpi. These and other settings can be changed using dialog boxes specific to each printer supported by Windows. For example, Figure 84 shows the dialog box that appears for the Apple LaserWriter Pro 600 printer.

After selecting and configuring a printer, the contents of the main window can be printed by selecting the "Print" item in the "File" menu. Once you've selected this menu item, the range of data you've chosen in the dialog box shown in Figure 83 will be printed on the device you've chosen. While the failure data is printing, a dialog box will appear to tell you that printing is in progress. This dialog box has a "Cancel" pushbutton, which will stop the printing operation if you click it. Once the failure data has been printed, the dialog box will disappear from the screen, regardless of whether or not you've clicked on the "Cancel" pushbutton.

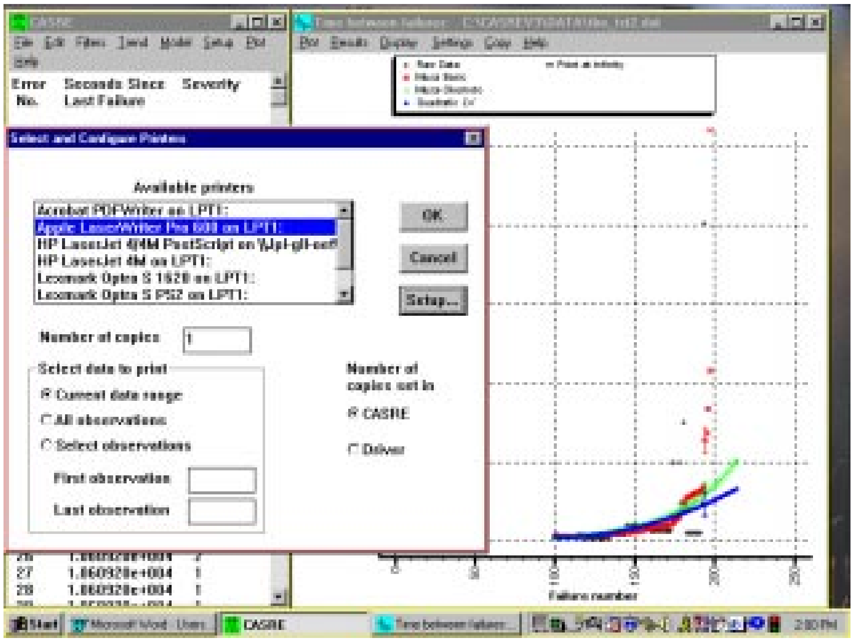

Figure 83 - Choosing a printer on which to print the failure data

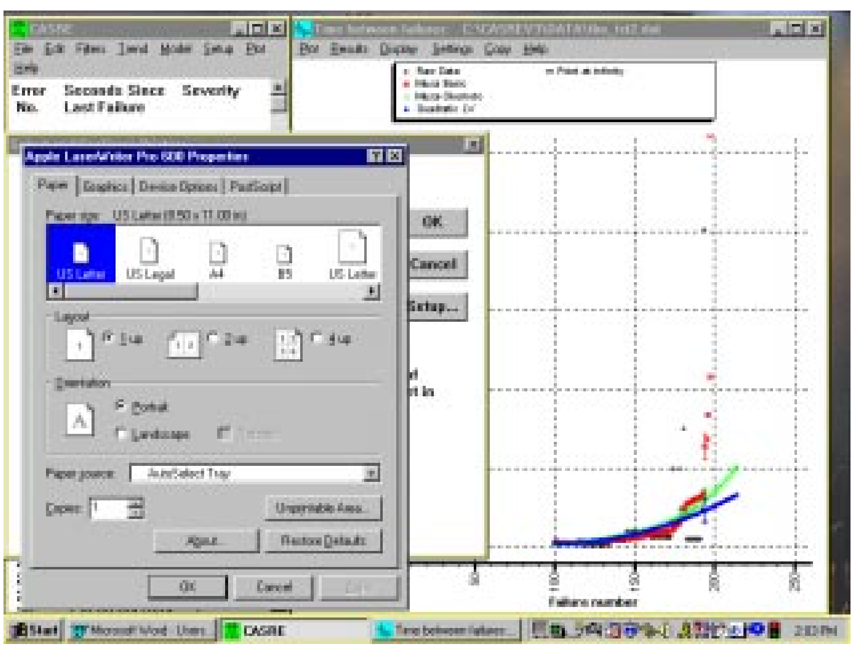

Figure 84 - Changing the configuration for the selected printer

4.11.2. Printing Model Result Plots

The raw data and model results plots shown in the graphic display window can be printed in a manner very similar to that described for the main window. Note that a printer can be chosen for the graphic display window, separate from the one chosen from the main window. Printers for the graphic display window are selected and configured in almost exactly the same way as shown in Figures 83 and 84, using the "Setup printer" item in the graphic display window's "Plot" menu. For the graphic display window, however, there are no radio buttons with which you can select a subset of the plot shown in the graphic display window. To make sure that the plot fits properly on the paper, a dialog box will appear in which you can enter a scale factor to control the size of the plot on the printout.

If you're currently displaying a plot that you've previously saved to file (see paragraph 4.12.2 in the next section), you can also print that plot. As above, a dialog box will appear in which you can enter a scale factor to control the size of the plot on the printout. This feature is included because a plot file produced on one system may have a different size when you try to print it on another system, and may not fit well on the printer. Changing the value of the scale factor will let you fit the plot file to your printer. Reducing the value of the scale factor produces a smaller printed plot; increasing it will increase the size of the printed plot.

4.11.3. Printing the Model Results Table

The model results table can be printed in a manner very similar to that described for the main window. Note that a printer can be chosen for the model results table, separate from the one chosen from the main window and the graphic display window. Printers for the graphic display window are selected and configured in almost exactly the same way as shown in Figures 83 and 84, using the "Setup printer" item in the graphic display window's "Plot" menu. For the graphic display window, however, there are no radio buttons with which you can select a subset of the plot shown in the graphic display window. Instead, there are two radio buttons which let you decide whether to print only the currently-displayed model results, or the results for all of the models that were run. The default selection is to print all of the model results.

4.12. Saving Data and Model Results

The failure data shown in the main window and the model results shown in the graphic display window can be saved as disk files. To save the failure data shown in the main window, you'll use the "Save" or "Save as..." items in the "File" menu. To save plots shown in the graphic display window, there is a "Save as..." option in the graphic display window's "Plot" menu that works nearly the same as the "Save as..." menu item in the main window's "File" menu. We've included these capabilities because you may want to display model results from an earlier time during the test period without having to run the models again, or it may be useful to look at modeling results for a similar project in the past to get some insight into what might happen as testing progresses. CASRE lets you save the contents of the graphic display window as a disk file that can be opened and drawn in the graphic display window during subsequent sessions.

4.12.1. Saving Data Files

The failure data displayed in the main window can be saved in one of two ways:

1. Choose the "Save" item in the "File" menu to save the data in the main window to the disk, using the name of the file. This has the effect of updating the disk file to reflect changes that have been made (e.g. applying filters). Note that the "Save" item is available only after you've made changes made to the data (e.g. applying one or more filters).

2. Choose the "Save as..." item in the "File" menu to save the data under a new name. The original disk file will remain unchanged, and a new file will be created. If this option is used, the original data file will be closed, and the main window will now display the contents of the newly-created file. The net effect is that the original file has been renamed, and that a backup version of the original file is retained on disk under the original name.

4.12.2. Saving, Re-Displaying, and Printing Model Result Plots

The contents of the graphic display window can be saved to disk for later re-display. For the graphics display window, there is only a "Save plot as" item in the "Plot" menu. This option behaves nearly identically to the "Save as..." option in the main window. When the graphic display window shows information in a graphical form (e.g. plots of times between failures, cumulative failure plots, reliability growth plots), saving the contents of the window produces a Windows metafile (see the Windows user's guide for details) that can be later be redisplayed in the graphic display window. If the contents of the graphic display window are text tables, saving the contents of the window produces an ASCII text file that can be displayed by invoking an external application (e.g. the "Write" editor mentioned earlier). However, ASCII text files CANNOT be redisplayed in the graphics display window. You can use an external application, such as a text editor or word processor, to display these files. Details on external applications are given in paragraph 4.4.

The following types of displays are saved as ASCII text files rather than as Windows metafiles when using the "Save plot as..." menu item to save the plot as a disk file:

1. **Goodness-of-fit**
2. **Model noise**
3. **Rank summary**
4. **Ranking detail**

These four types of displays, then, CANNOT be redisplayed in the graphic display window using the graphic display window's "Draw from file" menu item. However, the files to which they are saved can be brought into a word processor, DBMS, or spreadsheet for inclusion in documentation (e.g. test status reports) or further analysis.

With the exception of these four types of display, a previously-saved plot file can be redisplayed any time after it has been written to disk by choosing the "Plot" menu's "Draw from file" option. This brings up a dialog box identical to that shown when the main window's "Open" menu item is chosen. After selecting a plot file from the scrolling lists, and clicking on the "OK" button, the contents of the plot file appear in the graphic display window. The contents of the main window are unaffected. Note that when displaying a plot file, the "Results", "Display", and "Settings" menus are all disabled. Redisplaying the current set of failure data and model results will re-enable these menus.

To redisplay the current set of data, choose the "Draw from file" item in the "Plot" menu. Select and click on the "Plot current data and results" pushbutton. This will close the plot file, erase the contents of the graphic display window, and redisplay the currently open failure data and current model results. While you're displaying a plot file, you can print it in the same way that you print the contents of the graphic display window (see paragraph 4.11.2).

4.12.3. Saving the Model Results Table

The failure data displayed in the main window can be saved by choosing the "Save as..." item in the model result table's "File" menu to save the data under a new name. This will save the results for all of the models as an ASCII text file. The name of the file will be the name you've supplied in the dialog box that is displayed as a result of choosing the "Save as..." menu item. The model results are formatted as described in paragraph 4.10.1. The purpose of this capability is let you import the model results table into a spreadsheet, database, or statistical package for further analysis.

4.13. External Applications and Model Combination Definitions

The CASREV3.INI file, first mentioned in paragraph 2.2, is the configuration file in which external applications are listed, the conversion factors used in the "Change time units" filter (see paragraph 4.5) are kept, and the descriptions of user-defined combinations are kept. You can change this file in the following ways:

1. External applications can be added to or removed from the main window's "External applications" submenu. Only the "Notepad" editor, which is included with Windows, cannot be removed from CASRE's configuration.

2. Model combination definitions can be added to the configuration.

3. Model combination definitions **that have been defined by the user** can be removed. As mentioned earlier, there are four permanent combination definitions - these are the DLC/4/S, ELC, MLC, and ULC definitions that appear in the lists of TBF type models.

4. Use a text editor to change the portion of the file containing the conversion units used in the "Change time units" filter. You can't make these changes from the CASRE menus. You'd only want to change these factors if you find that they are different from your organization's experience. For example, we assume that one week is equal to 40 hours. You may find that for your organization, one week is equal to 56 hours. **If you do have to change these factors, make sure that they remain consistent with each other. For instance, don't set a week to 56 hours, and then forget to make sure that the factors for converting days, months, and years to hours are consistent with one week being 56 hours.**

The following paragraphs explain how to change this file. The configuration file format itself is given in Appendix D.

4.13.1. Adding External Applications

Up to 67 external applications can appear in the main window's "External application" submenu. One of these is the "Notepad" editor; the remaining applications are those that have been added by the user. These applications are not limited to editors, but can also include database management systems, spreadsheets, and any other application that runs under Windows95, Windows 3.1, or DOS 5.0.

To add an external application to CASRE's configuration, choose the "Add application" item of the main window's "Setup" menu. The dialog box shown in Figure 85 on the next page is displayed. Complete the edit windows in the dialog box as follows:

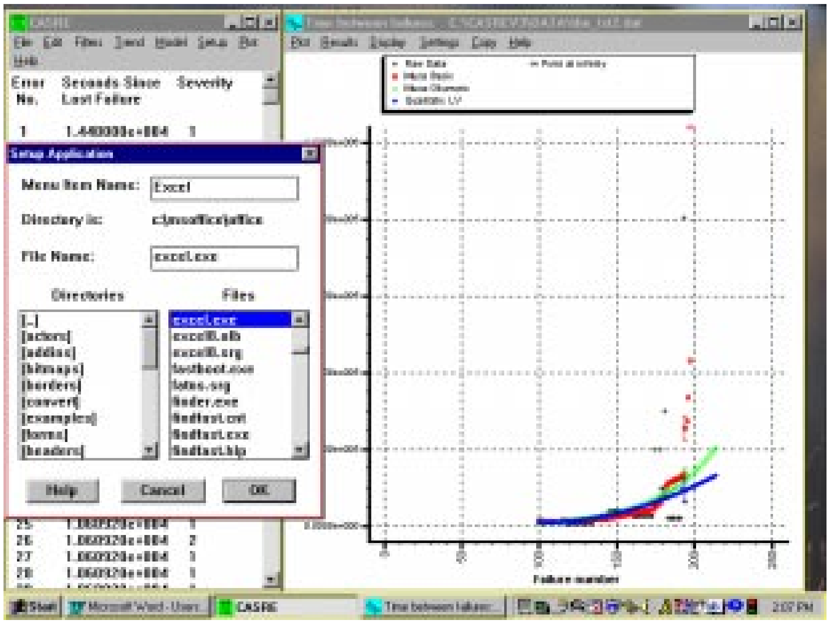

Figure 85 - Adding an external application to CASRE's configuration

1. In the edit window labeled "Menu Item Name", enter the name of the application as it should appear in the "External application" submenu.

2. Select the application. You can do this in the same way that you open a file. First, double-click in the scrolling list on the left to get to the subdirectory in which the application resides. The “Directory is:” window in the dialog box tells you which subdirectory you’re currently in. When you’ve reached the subdirectory in which the application file resides, do one of the following:

   - Click on the name of the application file in the scrolling list on the right. The application file name is displayed in the “File Name” window. Click “OK” when you’re done.
   - Alternatively, you can double-click on the name of the application file in the scrolling list on the right. This is equivalent to clicking once on the application file name and then clicking “OK”.

   You can also enter the fully-qualified name of the application file by just typing into the “File Name” window, but it’s probably easier to find it by scrolling and clicking in the scrolling list boxes.

   The file may be any executable file, including .BAT, .COM, .EXE, or .PIF files. **See the WindowsNT, Windows95, or Windows 3.1 user's guides for more details on executable files, especially .PIF files. Do not include the subdirectory in which the file appears as part of the file name.**

For instance, suppose you want to add Excel as an external application. Complete the “Setup Application” dialog box as shown in Figure 85. The system on which these examples are done is configured so that the application file, “excel.exe”, is located in “c:\msoffice\office”. Just scroll and click in the left scrolling list to get to that subdirectory. Then click “excel.exe” in the right scrolling list, as shown in Figure 85. To have the application appear on the external applications menu as “Excel”, just fill in the “Menu Item Name” window as shown in Figure 85. If you then choose the "External application" submenu of the main window's "Edit" menu after performing these steps, you'll see "Excel" as an item in that submenu. This is shown in Figure 86 below.

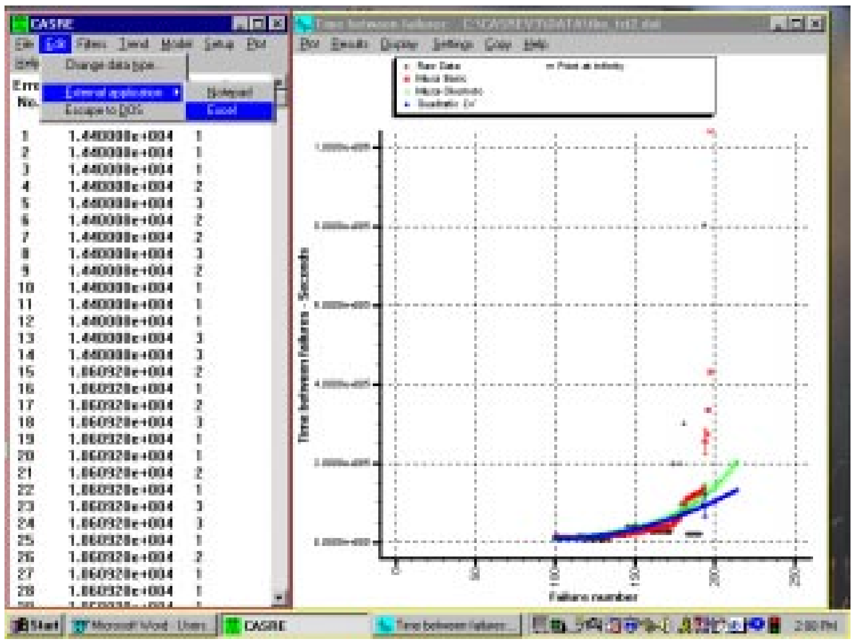

Figure 86 - Appearance of newly-added external application in "External application" submenu

4.13.2. Removing External Applications

With the exception of the "Notepad" editor, external applications can be removed from CASRE's configuration. To remove an application, choose the "Remove application" item of the main window's "Setup" menu. This dialog box is shown in Figure 87 below.

In Figure 87, all of the external applications that can be removed appear in the scrolling list box. To remove an application, perform the following steps:

1. Select the name of the external application to be removed. The name is the same name appearing in the "External application" submenu. The name of the application to remove will appear in the window at the top of the list box.

2. Click on the "Remove" button to remove the application from CASRE's configuration. The name of the application will be removed from the scrolling list box, and CASRE's configuration will be changed to remove that application.

As many applications as desired can be removed in this manner while the dialog box is displayed. When all desired applications have been removed, click on the "Done" button.

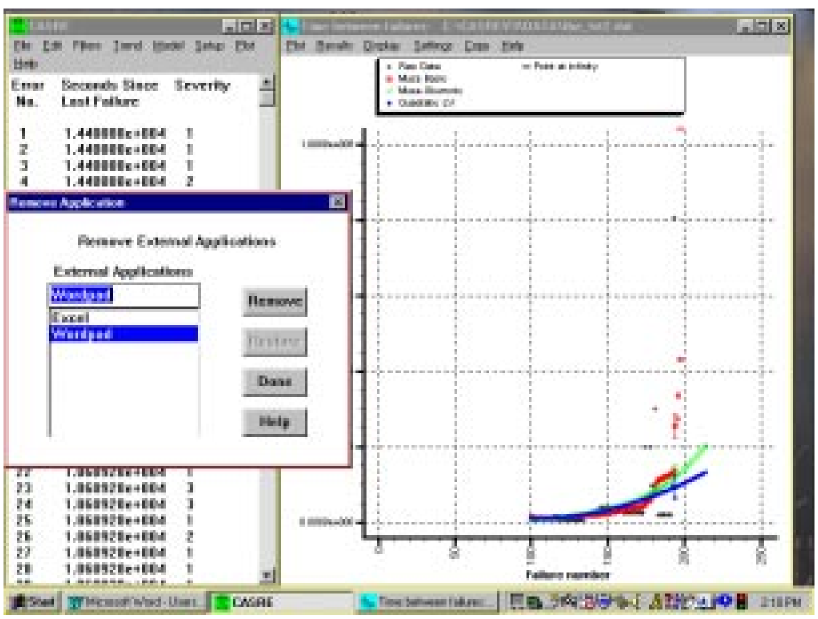

Figure 87 - Removing external applications

The most recently removed application can be restored to the configuration by clicking on the "Restore" button. In Figure 87, if applications "Wordpad" and "Excel" were to be removed in that order, "Excel" could be restored by clicking the mouse on the "Restore" button. The "Wordpad" application, however, could not be restored.

Suppose you decide that you no longer want to be able to run Wordpad from inside CASRE. For the purposes of this example, Wordpad has been set up to have the name "Wordpad" on the "External application" submenu. Just choose the "Remove application" item in the main window's "Setup" menu, and the dialog box shown in Figure 87 will appear. Use the mouse to select the name "Wordpad" in the dialog box's scrolling list, then click on the "Remove" button. If you're not removing any more applications, click on the "Done" button. The next time you select the "External application" submenu, the name "Wordpad" will no longer be on that submenu.

4.13.3. Adding User-Defined Model Combinations

As mentioned in paragraph 4.8.4, CASRE has four pre-defined model combination definitions that are a permanent part of its configuration. These are:

1. Dynamically Weighed Linear Combination - Sliding re-evaluation window, 4 observations wide (DLC/S/4)
2. Equally Weighted Linear Combination (ELC)
3. Median Weighted Linear Combination (MLC)
4. Unequally Weighted Linear Combination (ULC)

The components of each of these combinations are time between failures models, so they will only appear in the list of available models if a time between failures data set has been opened. These combination definitions cannot be deleted from CASRE's configuration. For more details on these combination models and analyses of their applicability, see "Applying Reliability Models More Effectively" by M. Lyu and A. Nikora, published in IEEE Software, vol. 9, no. 4, pp 43-52, July, 1992 [Lyu92].

You can add combination definitions to the CASRE configuration. The total number of atomic models (models that themselves are not combinations, e.g. Geometric model, Littlewood-Verrall model), permanent combination definitions, and user-defined combinations cannot exceed 50.

4.13.3.1. Selecting Components for the Combination

The following guidelines for choosing the components of a combination are based on experience with previous projects.

1. Select models whose assumptions about the testing process are different. For instance, the components of the ELC model are the Littlewood-Verrall, the Musa-Okumoto, and the NHPP models. Whereas the NHPP model assumes an upper bound on the total number of failures, the Littlewood-Verrall and the Musa-Okumoto do not. Furthermore, the NHPP and Musa-Okumoto assume that all faults discovered will be perfectly repaired (perfect debugging), while the Littlewood-Verrall model assumes that repairing a fault may induce additional faults, thereby lowering the software's reliability. Among failure counts models, the NHPP and the Schneidewind models will produce identical results if they're included in a combination. This is because if the Schneidewind model is included as part of a combination, all of the failure counts for all test intervals must be used by the model, which makes it equivalent to the NHPP model.

2. Do not choose too many models in formulating the combination. Experience indicates that a combination with three components significantly increases predictive accuracy over that of the individual components. Combinations having more than four components are not recommended.

3. Except for a research environment, avoid choosing components that are themselves combinations. Although CASRE allows combinations to be formed from other combinations, not enough work has been done to compare the predictive accuracy of these types of combinations with that of simpler combinations and atomic models.

4. Previous experience indicates that dynamically weighted combinations in which weights are based on changes in the relative accuracy measure have the best predictive accuracy, especially when the size of the re-evaluation window (used in re-assigning weights to the components) has a size between 3 and 5 observations, and the re-evaluation window is a sliding rather than a fixed window. The equally weighted linear combinations appear to have the next-best predictive accuracy.

You can define three types of combinations:

1. Statically weighted linear combination - the weights of the components have fixed values throughout the modeling run. These are referred to as SLC models. Equally weighted linear combinations are a special case of this type of combination.

2. Combinations in which weights are dynamically assigned based on comparisons of component model results over the previous N observations. These are also referred to as RLC models. In this type of combination, the results of the component models

are compared after a specific number of observations. The model predicting the smallest time to the next failure (or the highest number of failures in the next test interval) is assigned a specific weight, the model predicting the next smallest time to the next failure is assigned another weight, and so forth. The MLC and ULC models are instances of this type of model.

3. Combinations in which weights are dynamically assigned based on changes in the prequential likelihood statistic over the previous N observations. These are referred to as DLC type models. The DLC/S/4 model is an instance of this type of model.

To form a combination, choose the "Define combination" item in the main window's "Model" menu. The submenu shown in Figure 88 below will appear. Depending upon the type of combination you want to form, you'll choose one of the three items in that submenu. Each of the three submenu items will display a dialog box specific to the type of combination being formed. Completing the fields of the dialog box will define a new combination of the desired type. The following paragraphs describe the steps required to form each of these types of combinations. The combinations themselves are also described in greater detail in the following paragraphs.

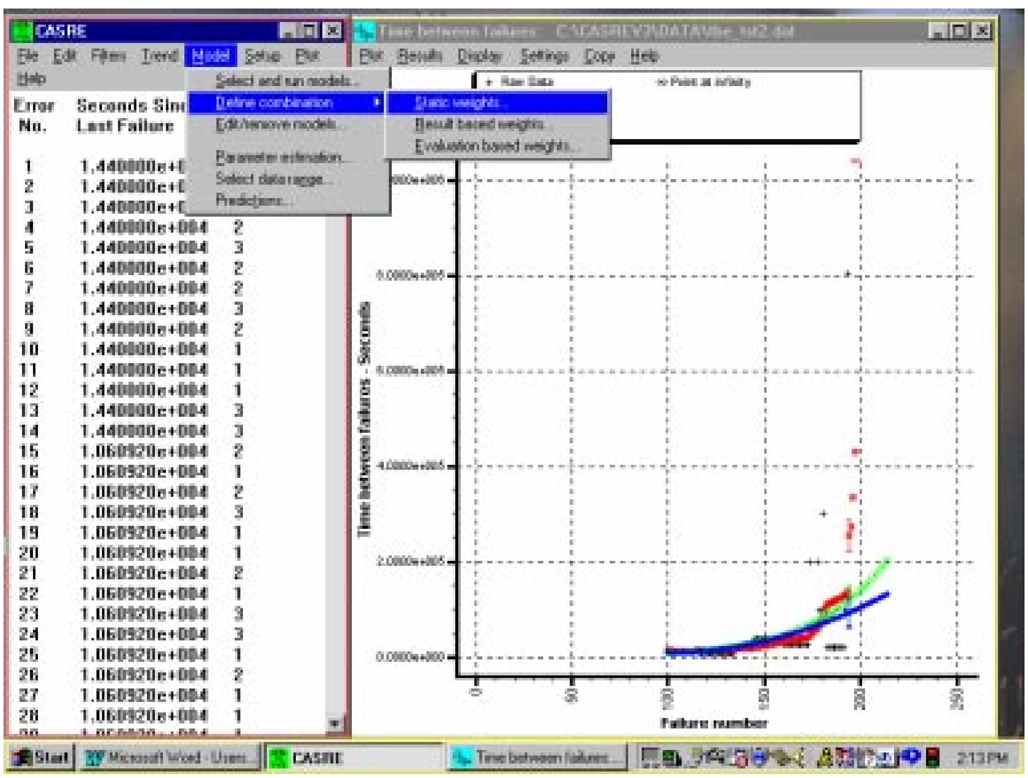

Figure 88 - Defining a new model combination - choosing the type of combination to be formed

4.13.3.2. Statically Weighted Combinations

To define a statically weighted combination, choose the "Static weight..." item from the submenu shown in Figure 88. The dialog box shown in Figure 89 below will appear. You will complete the fields in this dialog box to choose components for the combination, to assign weights to each component, and to give the combination a name. Proceed as follows to define a statically weighted combination:

1. Decide whether the combination is to be based on failure counts or time between failures type models. Do this by clicking on the appropriate radio button in the "Input Data Type" box at the bottom center of Figure 89. Clicking the "Interfailure times" radio button will display all the time between failures models in the "Available Models" list, while clicking the "Failure counts" radio button will display all of the failure counts models.

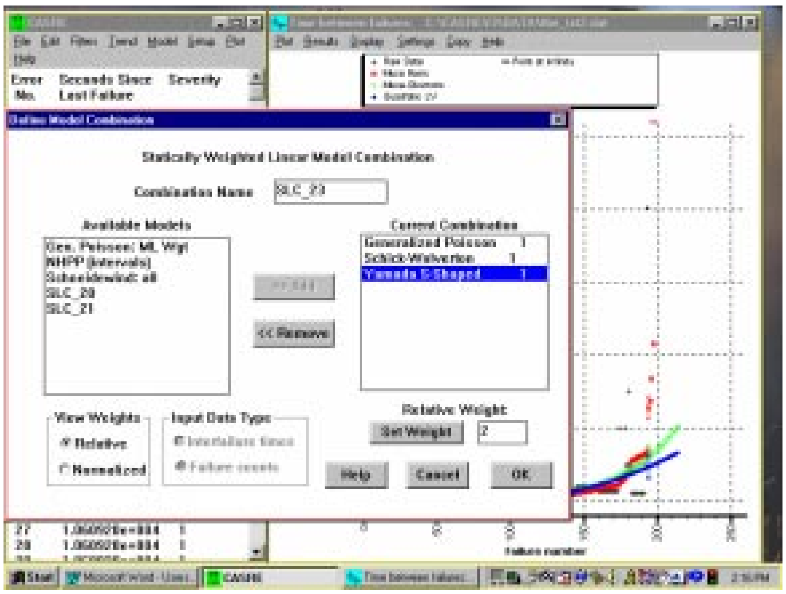

Figure 89 - Defining a statically weighted combination of failure counts models

2. Select models from the "Available Models" list and place them in the "Current Combination" list. Do this by using the mouse to highlight the name of the desired component in the "Available Models" list, then clicking on the "Add" button near the middle of the dialog box (or double-clicking on the name of the model in the “Available Models” list). The desired component will then appear in the "Current Combination" list, and will be deleted from the "Available Models" list. Figure 89 shows the Generalized Poisson, Schick-Wolverton, and Yamada S-Shaped models having been selected in this way.

   NOTE: After the first component has been inserted into the "Current Combination" list, the "Input Data Type" controls are disabled. Make sure to select the desired input data type before identifying any of the components.

3. Assign weights to each component. As shown in Figure 89 for the Generalized Poisson, Schick-Wolverton, and Yamada S-Shaped models, the default weight of each component is 1. To change the weight of a component, click on the name of that component in the "Current Combination" list. The weight of that component will appear in the "Relative Weight" edit window near the lower right of the dialog box. Use the keyboard to enter an integer between 1 and 99, then click on the "Set weight" button. This changes the weight of the selected component, and the new weight will appear after the component's name in the "Current Combination" list.

   As an example, let's set the relative weight of the Yamada S-Shaped model to 2. In Figure 89, we've selected the Yamada S-Shaped model by clicking on its name in the "Current Combination" list. After the default weight of "1" appears in the "Relative Weight" field, we set that value to "2" as shown. All that needs to be done to set the relative weight of the Yamada S-Shaped model to 2 is to click on the "Set weight" button.

4. Give the combination definition a name. CASRE assigns a default name to the definition, based on the type of the combination (SLC, DLC, or RLC) and the total number of models, both atomic and combination, in CASRE's configuration. To change the name, use the mouse to select the "Combination Name" field at the top of the dialog box. Use the keyboard to enter a new name or to change the existing one. **Note that only the first 20 characters of this field are used as the name.**

5. Select the "OK" button to complete the combination definition. The new definition will become part of the CASRE configuration, and will be available during the remainder of the current session as well as during subsequent sessions.

While defining a model combination, you might decide to remove the name of a component from the "Current Combination" list. To do this, use the mouse to choose the name of the component in the "Current Combination" list, then click on the "Remove" button at the center of the dialog box (or double-click on the name of the model in the “Current Combination” list). The selected component is then removed from the combination - its name disappears from the "Current Combination" list and reappears in the "Available Models" list.

The weight for each component is a relative weight. For example, suppose that the Generalized Poisson Process (GP) model has been selected with a weight of 1, the Schick-Wolverton (SW) model with a weight of 1, and the Yamada S-Shaped (YM) model with a weight of 2. The output of the combination is then (1 * GP output + 1 * SW output + 2 * YM output)/4. The weights for each component can be displayed in a normalized form. To do this, click on the "Normalized" radio button in the "View Weights" box at the bottom left of the dialog box. The weights for each component will then be displayed in a normalized form. Continuing the example, the weight for the GP model will be displayed as .25, the weight of the SW model as .25, and the weight of the YM model as .5. **When displaying normalized weights, all controls in the dialog**

box except the "Relative" radio button in the "View Weights" box are disabled. To resume adding components to or removing them from the combination, first click on the "Relative" radio button.

CASRE prevents you from assigning the name of an existing model to a combination definition. Suppose that one of the components is itself a combination having the name "SLC_20" or having the combination "SLC_20" as one of its components. The combination definition is then prevented from being given that name.

4.13.3.3. Dynamically Weighted Combinations Based on Comparisons of Modeling Results

To define this type of dynamically weighted combination, choose the "Result-based weight..." item from the submenu shown in Figure 88. The dialog box shown in Figure 90 below will appear. Use the fields in this dialog box to choose components for the combination, assign weights to each component, and give the combination a name.

Proceed as follows to define a combination whose weights are based on ranking model results:

1. Decide on the type of the combination (failure counts or time between failures) as for statically weighted combinations.

2. Select components of the combination as for statically-weighted combinations. For result-based combinations, the name of each selected component appears in the "Current Combination" list shown in Figure 90, just as for statically-weighted combinations. To show that it can be done, note that one of the components is itself a combination (the DLC/S/4 model). Weights are handled differently than for static combinations. In result-based combinations, weights are not directly assigned to a particular model. Rather, a set of weights is specified. A component assumes a weight, based on how its results compare with the results of the other components.

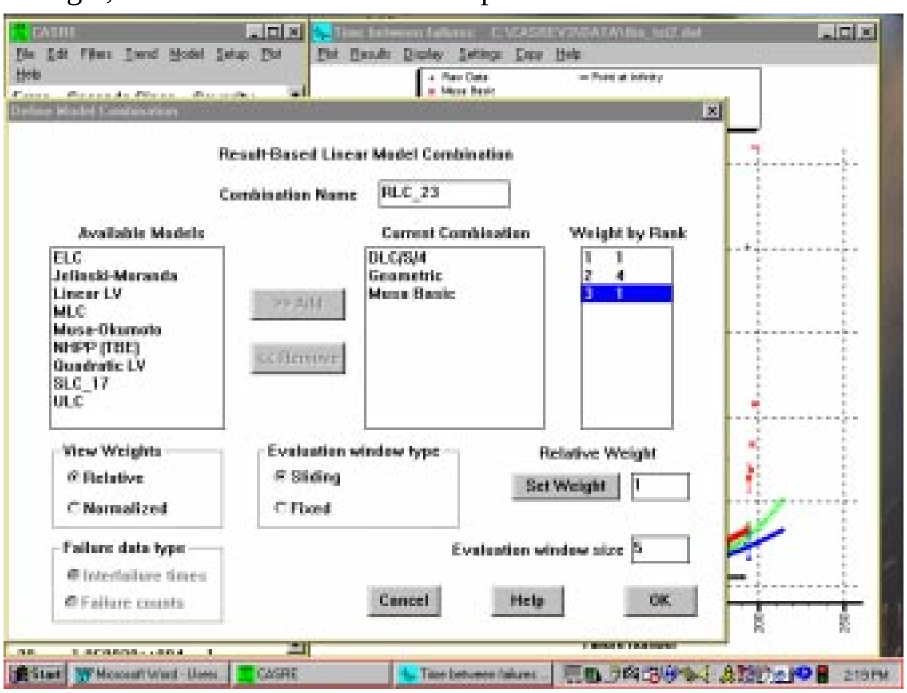

Figure 90 - Defining a combination based on ranking model results for time between failures models

In Figure 90, we see that the entries in the "Weight by Rank" column read (1 1), (2 4), and (3 1), where (x y) denotes a possible rank (x) and the weight (y) given to the model having that rank. If the "Fixed" type of re-evaluation window has been chosen instead of the "Sliding" window, the components would run for the number of observations specified in the "Evaluation window size" edit field. In this example, this number is 5. This window is positioned below the "Relative weight" window in the lower right corner of Figure 90. The weights for each component are re-assigned every 5 observations, based on a ranking of the model results at the end of that 5 observation interval. Suppose that at the end of one of these 5 observation intervals, the Geometric model predicts the smallest time to the next failure. CASRE's ranking scheme ranks this model a rank of 1 over that interval. The Geometric model will then be given a weight of 1 over the next interval of 5 observations. Similarly, if the Geometric model had predicted the largest time to the next failure at the end of that interval, its rank would be 3, and its weight for the next interval would be 1.

If the re-evaluation window had been a sliding window, but with everything else the same, the model weights would be recomputed after every observation, based on the way in which the model results had changed over the previous 5 observations. For example, if the **average** time to the next failure for the Geometric model, averaged over the previous 5 observations, was lower than the **average** time to the next failure for the other two models, the Geometric model would be given a rank of 1 and a weight of 1.

3. Specify the values from which a component's weight may be chosen. For each component, an (x y) pair is inserted into the "Rank by Weight" list to the right of the dialog box. For each rank, the default weight is 1. To change the weight assigned with a particular rank, first use the mouse to highlight the desired entry in the "Weight by Rank" list. For example, Figure 90 shows the third entry of the "Weight by Rank" list being highlighted. This indicates that the weight associated with a ranking of "3" can be changed. The weight associated with a ranking of “2” has already been changed to 4 from the default value of 1.

   Upon highlighting the desired entry in the "Weight by Rank" list, the weight currently associated with that ranking will appear in the "Relative Weight" edit field in the lower right of the dialog box. Use the keyboard and the "Set weight" button to set the new weight, as for statically weighted combinations.

4. To remove a component from the current combination definition, use the same procedure as for statically weighted combinations.

5. Give the combination a name, as for statically weighted combinations.

For result-based combinations, a default value of 5 is given to the size of the re-evaluation window. By default, the re-evaluation window is defined as a sliding window. These default values have worked well for previous projects. However, they can be changed if desired. To specify the type of re-evaluation window, simply click on the desired radio button in the "Evaluation window type" box. To change the size of the re-evaluation window, click the mouse on the "Evaluation Window Size" edit field, and use the keyboard to change the value.

4.13.3.4. Dynamically Weighted Combinations Based on Prequential Likelihood Changes

To define this type of dynamically weighted combination, choose the "Result-based weight..." item from the submenu shown in Figure 88. The dialog box shown in Figure 91 below will appear. Use the fields in this dialog box to choose components for the combination, assign weights to each component, and give the combination a name.

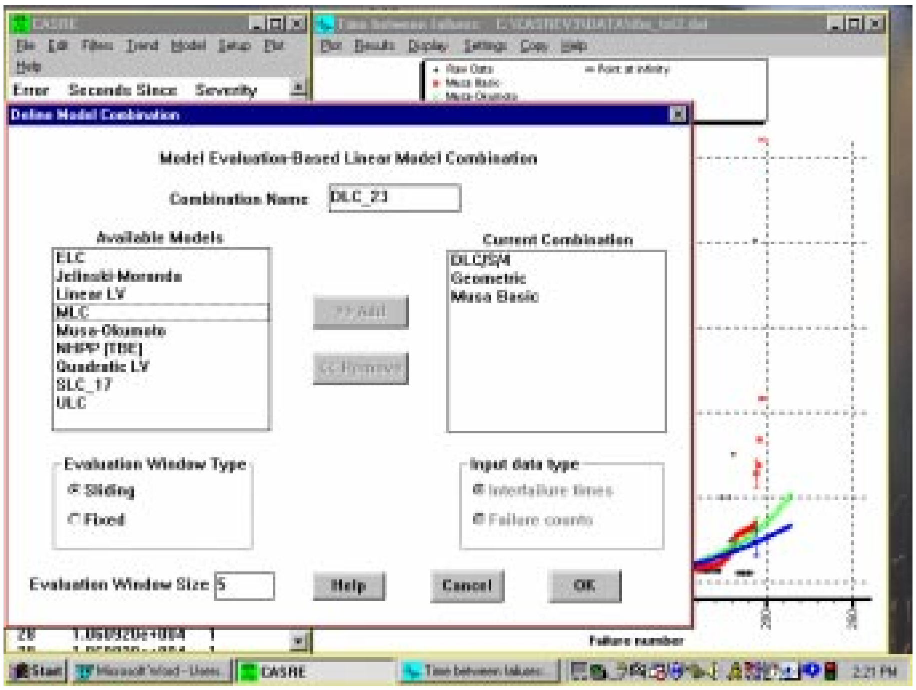

Figure 91 - Defining a combination, based on changes in prequential likelihood, for time between failures models

Proceed as follows to define a combination in which the weights of the components are dynamically computed, based on changes in the relative accuracy measure:

1. Select each component of the combination using the same procedure as for statically weighted combinations.
2. As with the result-based combinations, the re-evaluation window is given a default type of "Sliding", and the default value of its size is 5. Previous experience indicates that these default values should work well for most testing efforts. However, they may be changed, using the same procedure as for result-based combinations.
3. Removing components from the definition and naming the model work as for the other two types of combinations.

For this type of combination, there is no need for assigning or displaying weights in the dialog box, since they are dynamically computed and assigned.

4.13.4. Removing and Modifying User-Defined Combinations

After a model combination definition has been made, it may be modified or removed from the CASRE configuration by selecting the "Edit/Remove model..." item from the main window's "Model" menu. The dialog box shown in Figure 92 below is displayed after this menu item has been chosen.

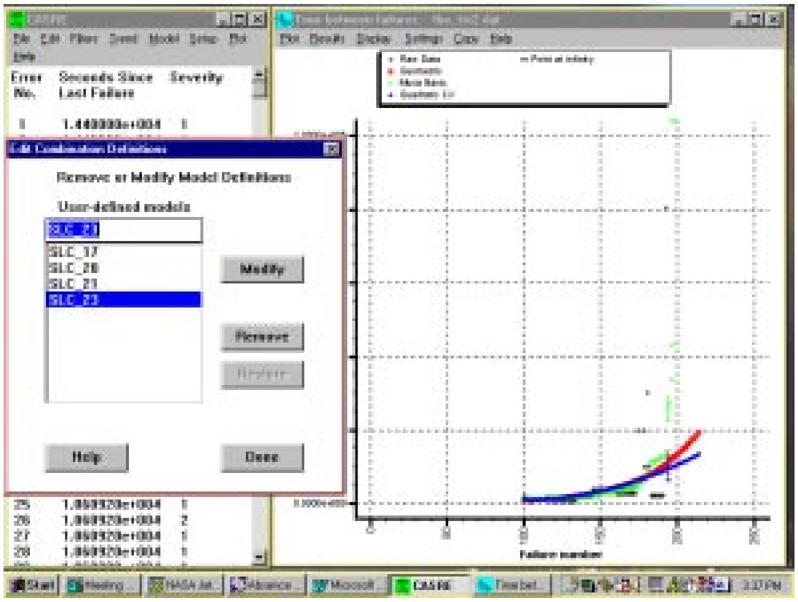

Figure 92 - Removing and modifying combination model definitions

To modify a combination definition, use the mouse to highlight the name of the desired model in the "User-defined models" list, then click on the "Modify" button. One of the three combination definition dialog boxes shown in Figures 89-91 will appear, depending upon the type of the combination definition. Use the combination definition dialog box as described in the preceding paragraphs to modify the combination definition. Clicking the "OK" button in the combination definition dialog box will change the CASRE configuration to reflect the changes to that combination's definition, and will then return to the dialog box shown in Figure 92. Clicking the "Cancel" button in the combination definition dialog box will leave the original combination definition in place, and return to the dialog box shown in Figure 92.

To remove a combination definition, use the mouse to highlight the name of the definition to be removed in the "User-defined models" list, then click on the "Remove" button. Figure 92 shows the "SLC_23" combination model, initially defined in Figure 90, being removed. The "Restore" button restores the last combination definition that was deleted, operating in the same manner as the "Remove" button in the dialog box for removing external applications from the CASRE configuration.

After modifying or removing a combination definition, another definition may be chosen to modify or remove. After all modifications and deletions have been completed, click on the "Done" button. This will remove the dialog box shown in Figure 92 from the screen.

Note: The "User-defined models" list contains only combinations that have been defined by the user. The DLC/S/4, ELC, MLC, and ULC combination definitions cannot be removed from the CASRE configuration. In addition, only those combinations that have not been selected to run appear in this list. For instance, suppose that the RLC_17 combination had been selected to run. Also suppose that earlier in the current CASRE session, a failure counts combination that we'll call SLC_13 had been created, that a failure counts data file had been in use, and that the SLC_13 combination had been selected to run on that data. In that case, you wouldn't be able to remove these two combination definitions from the CASRE configuration until after bringing up the "Select Models" dialog (as in paragraph 4.8.4) and removing those definitions from the list of "Models to Run" in that dialog box.

4.14. Setting Significance Levels for Goodness-of-Fit and Trend Tests

You can set the significance levels for the Chi-Square goodness-of-fit and trend tests from the main window's "Setup" menu. To set the significance level for the Chi-Square test, select the "GOF Significance" item in the main window's "Setup" menu. You'll get the dialog box shown in Figure 93 below.

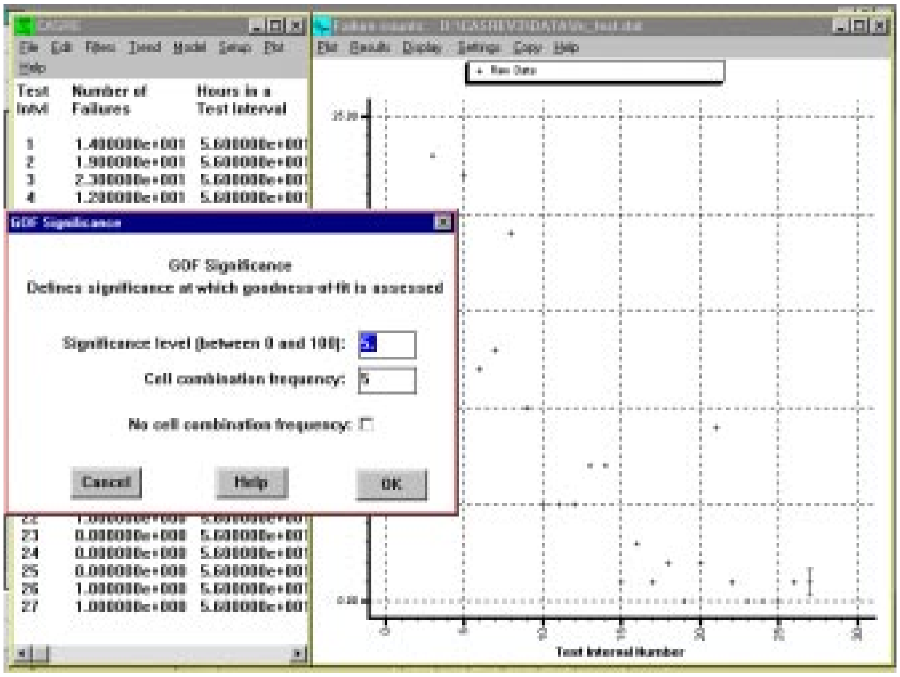

Figure 93 - Setting Chi-Square Test Significance

Enter the significance level (in percent) at which you want to determine the fit of the model results to the data. In Figure 93, for instance, we've specified that we want to know whether the model results fit the data at a 5% significance level (the default value). The next time models are run and the Chi-Square goodness-of-fit test is applied, the plot showing the results of the Chi-Square test will specify whether the model results fit the data at the 5% significance level. This significance level will also be saved in the CASRE configuration file so that it can be used in future sessions.

You also use this dialog box to enter the cell combination frequency (the default value is given as 5). This tells CASRE to group the observations of failure counts into intervals having at least the number of failures you specify. This is done to increase the statistical validity of the results. If you don't want CASRE to group the data, check the checkbox labelled "No cell combination frequency". However, changing the default value is not recommended unless you understand the implications of doing so.

The significance value for the Kolmogorov-Smirnov test, used for time between failures data, cannot be changed in this version of CASRE.

The significance level at which results are reported for the Laplace Test can also be changed. This is done by selecting the “Laplace Test Sig” item in the main window’s “Setup” menu and completing the dialog box that appears, shown in Figure 94 below. In Figure 94, we’ve specified that we want to the values of the Laplace Test at which we can reject or accept the null hypothesis at the 5% significance level (the default value). The next time the Laplace Test is applied, the plot showing the results of the Laplace Test will show the values of the statistic at which the null hypothesis can be rejected or accepted at a 5% significance level. This significance level will also be saved in the CASRE configuration file so that it can be used in future sessions.

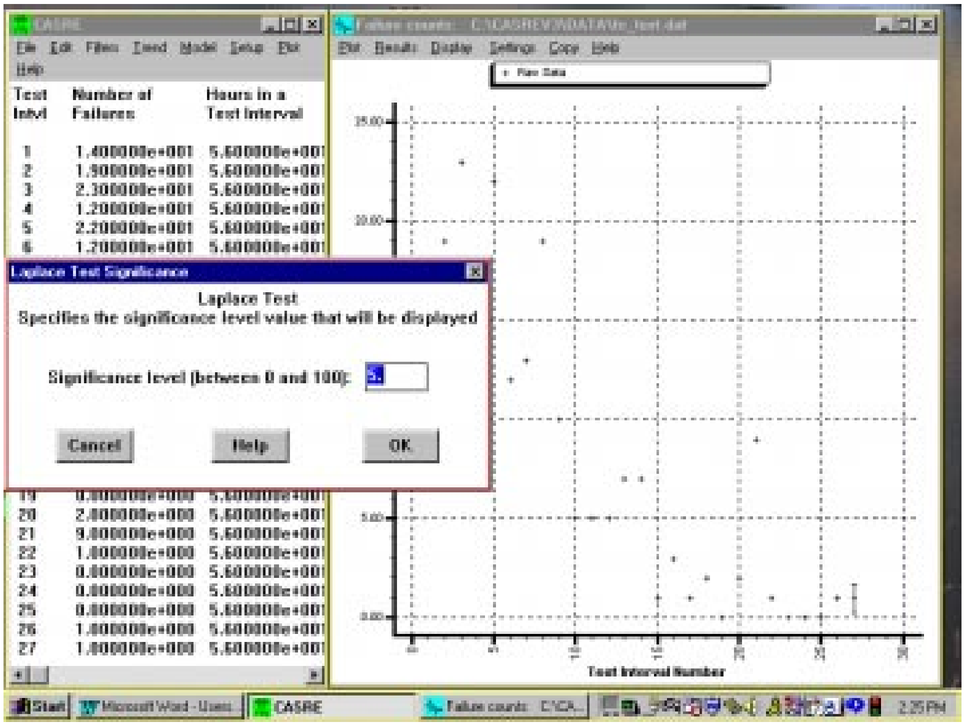

Figure 94 - Setting Laplace Test Significance

4.15. Setting the Length of the File Menu's File Name List

Recall that the File menu contains a list of the most recently opened files (shown in Figure 16). Selecting a name from this list opens that file in exactly the same way that opening it with the File menu's "Open" item does. This provides a shortcut method of opening a file, providing that it has been previously opened.

You can specify the number of files that can be kept in this list using the Setup menu's "Remember Most Recent Files" item. Selecting this item will bring up the dialog box shown in Figure 95 below. You select the maximum length of the file name list by entering a number between, and including, 0 and 9 in the dialog box's entry field. If you enter a "5", for instance, the maximum number of file names that will appear in the File menu's list is 5. The default length of the file name list is 5.

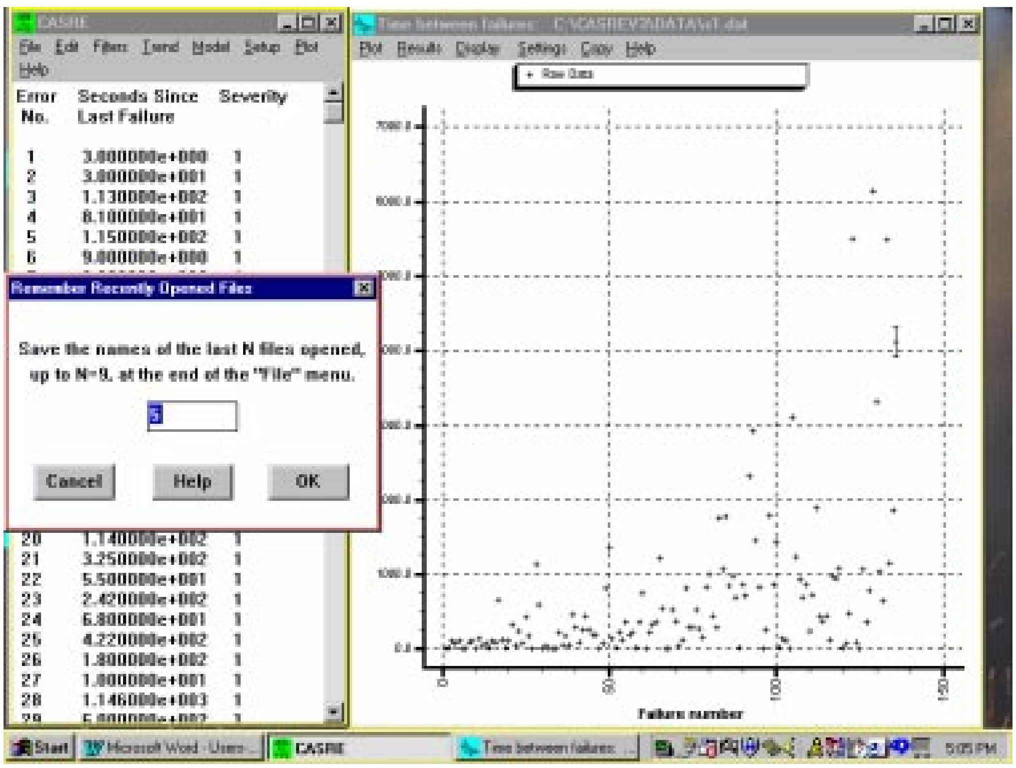

Figure 95 - Setting the Maximum Length of the File Name List

4.16. Ending the CASRE Session

To end a CASRE session, either choose the "Exit" item from the main window's "File" menu, or press the “F3” function key. If the contents of the main window have been modified by applying one or more filters, a message box will ask you whether the contents of the main window should be saved. Clicking on the "Yes" button will update the disk file associated with the main window's contents before ending the CASRE session. Clicking on the "No" button will end the CASRE session, leaving the contents of the disk file unchanged. Finally, selecting the "Cancel" button will cancel the "Exit" operation, leaving CASRE active and the displays unchanged.

If the contents of the main window have not been modified, selecting the "Exit" menu item will end the CASRE session.

4.17. Getting Help

Each of the CASRE windows has a "Help" menu. The "Help for help" menu item displays a standard "Help" window in which the operation of the Help system is summarized. Selecting the "Help index" menu item displays the index of topics that are available. The index for each window is organized in the same manner as the menu tree for that window.

Finally, most dialog boxes have a "Help" button which, when selected, displays a window in which the dialog box's purpose and an explanation of its controls are given.

Appendix A. References

Readers interested in more details of software reliability engineering theory and practice should consult the following works.

[Abde86] A. A. Abdel-Ghaly, P. Y. Chan, and B. Littlewood; "Evaluation of Competing Software Reliability Predictions," IEEE Transactions on Software Engineering; vol. SE-12, pp. 950-967; Sep. 1986.

[Lyu92] M. Lyu, A. Nikora, "Applying Reliability Models More Effectively", IEEE Software, vol. 9, no. 4, pp 43-52, July, 1992

[Lyu96] M. Lyu ed., Handbook of Software Reliability Engineering, New York: McGraw-Hill, 1996, ISBN 0-07-039400-8.

[Musa87] John D. Musa., Anthony Iannino, Kazuhiro Okumoto, Software Reliability: Measurement, Prediction, Application, New York: McGraw-Hill, 1987; ISBN 0-07-044093-X.

[Niko95] A. Nikora, M. Lyu, "An Experiment in Determining Software Reliability Model Applicability", proceedings of the Sixth International Symposium on Software Reliability Engineering, Toulouse, France, October 24-27, 1995

[NSWC83] W. H. Farr, “A Survey of Software Reliability Modeling and Estimation”, Naval Surface Warfare Center Technical Report TR 82-171, Naval Surface Warfare Center, September, 1983

[Scha79] R. E. Schafer, J. F. Alter, J. E. Angus, S. E. Emoto, “Validation of Software Reliablity Models”, Rome Air Development Center Technical Report RADC-TR-79-147, 1979

Appendix B. Glossary of Software Reliability Terms

Atomic Model A software reliability model that has only itself as a component (see Combination Model). Examples are the Schneidewind Model, the Musa Basic Model, and the Littlewood-Verrall Model.

Bias A systematic error introduced into sampling or testing by selecting or encouraging one outcome over another.

Combination Model A software reliability expressed as a linear combination of two or more other software reliability models. Each component of the model is assigned a weight, and the sum of the weights is always 1. Weights may be statically or dynamically assigned. Examples of these models are the DLC/S/4, ELC, MLC, and ULC models, which have the Littlewood-Verrall, Musa-Okumoto, and NHPP models as their components.

Failure An observed departure from system behavior as specified by the requirements specification for that system.

Failure Counts and Test Interval Lengths One type of failure data used by the models implemented in CASRE. For each interval during which a piece of software is tested, the following information is recorded:

- Number of failures observed during the test interval
- The length of each test interval

Failure frequency figures can be derived from this type of data.

Failure Frequency The number of failures that occur in a specified time period divided by the length of the time period. For instance, if you observe 6 failures over a 10-hour period, the failure frequency is 0.6 failures per hour.

Interfailure Time The length of time between two successive failures. An example would be that the second failure occurred 2.7 hours after the first failure. This approach gives an exact distribution of when failures occurred, while failure counts and test interval lengths data cannot provide this information. In addition to failure counts and test lengths, CASRE implements models that use this type of failure data.

Kolmogorov Distance The maximum absolute vertical distance between two cumulative distribution functions (see Kolmogorov-Smirnov Test).

Kolmogorov-Smirnov Test A test that can be used to determine how closely a model's predictions fit the actual data on which they are based. Cumulative frequency distributions are computed for both the model results and the data, and the maximum absolute vertical distance between the two distributions lets you determine how well the predictions fit the actual data.

Least Squares

A method of performing regression analysis. A software reliability model's parameters can be estimated using the least-squares technique. Unlike the maximum likelihood method, least squares can be used when the joint distribution of the random variables is not specified.

By way of an example, suppose we have pairwise uncorrelated random variables $T_1, T_2, ..., T_n$ with means $\beta_0 + \beta_1 t_1, \beta_0 + \beta_1 t_2, ..., \beta_0 + \beta_1 t_n$ and variances σ^2. The values of β_0 and β_1 that minimize the following sum of squares are defined to be the least-squares estimators of β_0 and β_1.

$$\sum_{i=1}^{n} (T_i - \beta_0 - \beta_1 t_i)^2$$

The same idea holds for non-linear functions. For instance, consider the Nonhomogeneous Poisson Process model. The form of this model is:

$$m(t) = a(1 - e^{-bt})$$

where a is the total estimated number of errors, b is the proportionality constant, and m(t) is the estimated number of failures found by time t. Least squares estimators for a and b are values that minimize the following expression:

$$\sum_{1}^{n} [\, f_i - (m(t_i) - m(t_{i-1})\,]^2$$

where n is the total number of test intervals, f_I represents the actual number of failures found in the i'th interval, and $m(t_i)$ is the estimated number of errors in the i'th interval.

Likelihood Function

Given n random variables $T_1, T_2, ..., T_n$, the likelihood function of those n random variables is defined to be the joint probability density function of those variables:

$$L(\theta) = f_{T_1 T_2 \ldots T_n}(t_1, t_2, \ldots, t_n; \theta)$$

This density is considered to be a function of the parameter θ. If T_1, T_2,..., T_n is a random sample from the probability density function

$f(t,\theta)$, then the likelihood function is written as:

$$L(\theta) = f(t_1;\theta)f(t_2;\theta)...f(t_n;\theta)$$

Maximum Likelihood A method of estimating the parameters of a software reliability model. Using this method, an estimate of a parameter is obtained by finding the value for that parameter that maximizes the likelihood function.

Model or Prediction Noise Noise inserted into a model's predictions by the model itself.

Noise Irrelevant or meaningless information occurring randomly along with desired information.

Parameter A parameter is a quantity which indexes a family of probability density functions. For instance, suppose we have a family of density functions given by $f(x; \lambda) = \lambda e^{-\lambda x} I_{(0,\infty)}(x)$, with $\lambda > 0$, then for each $\lambda > 0$, $f(\bullet;\lambda)$ is a distinct probability density function. λ is the parameter, and as λ ranges over the positive numbers, the collection $\{f(\bullet;\lambda):\lambda>0\}$ is a parametric function of density functions.

Reliability The probability of running a system without failure for a specified time in a specified environment. If the instantaneous failure rate for a system is given as z, and if a time t is specified as the amount of time for which the system will run, the reliability for that system over the interval of time t is given as:

$$R = e^{-\int_0^t z(x)dx}$$

Reliability Growth Model As a software system is tested, the reliability will tend to increase as failures are observed and the faults causing those failures are removed. This tendency of a software system's reliability to increase is the reason that software reliability models are often referred to as software reliability growth models.

Severity Severity indicates the impact of the system's failure or parts thereof on the mission, the system's users and environment, or both. For software developed under MIL-STD 2167A, severity ranges from 1 to 5.

Smoothing Operation A method of reducing the amount of noise in a set of data. One specific way of removing noise is by using a Hann window, which is a type of moving average.

Appendix C. Failure Data File Formats

There are two types of input that CASRE can accept. The first of these is observations of times between successive failures; the second is failure counts per test interval and test interval length. These input types are based on those accepted by version V of SMERFS. Each of these files is an ASCII text file; their formats are given below, with examples of each type. Before going on to the examples, however, we note that the following information should also be collected to enhance the accuracy of model predictions. This data is based on the recommended minimal set of data described in the ANSI/AIAA recommended practices document, ANSI/AIAA R-013-1992, "Software Reliability", as well as personal experience.

1 The date and time at which each failure was found, and the test interval during which the software was run that produced that failure.

2 The component being tested, referenced to a requirements specification or design document. As a matter of practicality, components should be larger than 2000 uncommented physical lines of source code. The reason for collecting this information is in case components are tested sequentially (e.g., first A, then B, then C,...), the reliability of the components can be separately modeled and then combined into a reliability estimate for the entire system. This will increase the accuracy of the predictions.

3 The date and time at which the testing method changed (e.g. a change from white box to black box testing). The reason for this is that the perceived reliability of the system depends on how it is executed.

4 The date and time at which the test environment changed. This includes changes in staffing level, the addition of test equipment, or upgrades to existing equipment. The accuracy of model predictions depends on the accuracy of the time between successive failures or failure count/test interval length data applied to the model. The reason for collecting this information is to more accurately characterize the length of a test interval.

5 The date and time at which the software being tested changes significantly. **This usually does not include fault repairs**, since fault repairs tend to be fairly localized with minimal change to the overall system. A good rule of thumb is that a significant change means that 15% or more of the software has changed, as measured in uncommented lines of source code or the number of object instructions. If a piece of software has changed significantly, the testing method and test focus may change also, which means that the reliability of the changed software should be measured separately from that of the previous version.

6 The severity of each failure, if a severity classification scheme has been defined for the test effort. This may allow estimation of the reliability for each class of failure.

Specifically, if enough failures are observed, the predicted failure intensity for mission-critical errors may be more accurately estimated.

Much of this information should be available from problem tracking systems as well as **correctly completed** test logs. Experience shows that most organizations will more easily record failure history data in terms of failures counts and test interval lengths.

C.1 Time Between Failures Input File Format

The first row in a file of time between failures data must be one of the following seven words followed by a carriage return: Seconds, Minutes, Hours, Days, Weeks, Months, Years. The first row names the units in which the times between failures are given. The second through last rows in a file of time between failures data have the following fields:

Error Number (integer)   Time since last failure (floating point)   Error Severity (integer)

There may be as many rows as desired in a file.

The following is an example of an input file having this format:

```
Seconds
    1        45.2    3
    2        34.78   2
    3,26.8 3
    4  50    1
```

Field separators can be either tabs, commas, or spaces. Error numbers start with 1 and continue in ascending order through the remainder of the file. Gaps in the sequence of error numbers are not permitted, thereby prohibiting sequences such as 1, 2, 3, 5, ... For time between failures data, an error number cannot appear more than once in the file. Error severities range from 1 to 9.

C.2 Failure Counts Input File Format

The first row in a file of failure counts must be one of the following seven keywords: Seconds, Minutes, Hours, Days, Weeks, Years. For failure count data, this keyword names the units in which the lengths of each test interval are expressed.

The second through last rows in a file of failure count data have the following fields:

Interval Number	**Number of Errors**	**Interval Length**	**Error Severity**
(int)	(float)	(float)	(int)

Remember to also include those test intervals for which no failures were seen. For each interval in which no failures were seen, give the interval number, set the number of errors to 0.0, and set the severity to 0, as well as completing the other fields.

The following is an example of failure count data file:

Hours			
1	5.0	40.0	1
1	3.0	40.0	2
1	2.0	40.0	3
2	4.0	40.0	1
2	3.0	40.0	3
3	7.0	40.0	1
4	5.0	40.0	1
5	4.0	40.0	1
6	4.0	40.0	1
7	3.0	40.0	1

Field separators are the same as those for time between failures data files. Note that there may be several records having the same interval number. This is because during a given test interval, there may be errors in more than one severity category observed. In the above example, we see that during interval 1, there were 5 errors of severity 1, 3 of severity 2, and 2 of severity 3. Also note that for the second test interval, there is no record for severity category 3. This is because there were no errors of this severity observed during this interval. The "number of errors" field is a floating point value because at some point, a filter or smoothing operation may be applied to the data, which could turn an integral value into a floating point number. Interval numbers start with 1, and continue in ascending order. As with time between failures data files, it is assumed that there are no gaps in the sequence of interval numbers. This prohibits sequences of interval numbers such as 1, 1, 1, 3, 4, 4, ... The severity field is the same as that for time between failures data.

Appendix D. CASRE Configuration File Format

The external applications that can be invoked from within CASRE, as well as the available models, are kept in a configuration file called CASRE.INI. This file is usually kept in the same subdirectory in which Windows is found. The configuration file listed below is the one shipped with CASRE. Lines of text starting with "//" represent comments that are not included in the configuration file.

```
[models]
//
//      This section names and describes the available models.
//      Immediately following each model name is a three-digit
//      value, XYZ.  X = 0 represents an atomic model (i.e., one
//      that is not part of any combination), while X not equal
//      to zero represents a combination.  X = 1 represents a
//      statically weighted combination, X = 2 represents a com-
//      bination in which weights are based on comparisons of
//      model results, while X = 3 represents combinations whose
//      weights are based on changes in the relative accuracy
//      measure.  Y = 0 represents a model taking time between
//      failures as input, while Y = 1 represents a model whose
//      input is failure counts and test interval lengths.  Z = 0
//      represents a model that cannot be deleted from the
//      configuration, and Z = 1 represents models that can be
//      removed from the configuration.  Note that none of the
//      atomic models have Z = 1, and that the four permanent
//      combination definitions, DLC/S/4, ELC, MLC, and ULC
//      also have Z = 0.
//
//      For atomic models, there are nine values separated by commas
//      following the three-digit value.  These values are flags
//      indicating whether the model can compute the low 95% con-
//      fidence values for the first three parameters, the most
//      likely values for  the model parameters, and the high 95%
//      confidence values for the model parameters.  A "0" means
//      that the value cannot be computed, while a "1" means that
//      it can be computed.  This is done because not every model
//      in SMERFS, which forms the basis for CASRE's modeling cap-
//      abilities, is able to compute low and high 95% confidence
//      values for the model parameters.
//
Musa Basic=000,1,1,0,1,1,1,1,1,0
Musa-Okumoto=000,0,0,0,1,1,1,0,0,0
Geometric=000,1,1,1,1,1,1,1,1,1
```

```
Schneidewind: all=010,0,0,0,1,1,0,0,0,0
Schneid. - Last n=010,0,0,0,1,1,0,0,0,0
Schneid. - Cum. 1st =010,0,0,0,1,1,0,0,0,0
Schick-Wolverton=010,1,1,1,1,1,1,1,1,1
Generalized Poisson=010,1,1,1,1,1,1,1,1,1
Gen. Poisson: ML Wgt=010,1,1,1,1,1,1,1,1,1
NHPP (intervals)=010,1,1,0,1,1,0,1,1,0
NHPP (TBE)=000,0,0,0,1,1,0,0,0,0
Linear LV=000,0,0,0,1,1,1,0,0,0
Quadratic LV=000,0,0,0,1,1,1,0,0,0
Yamada S-Shaped=010,1,1,0,1,1,0,1,1,0
ELC=100
ULC=200
MLC=200
DLC/S/4=300

[applications]
//
//	This section of the configuration file names and describes
//	each of the external applications, except for the "Notepad"
//	editor, that can be invoked from within CASRE.  The name
//	at the left of the "=" sign is the name of the application
//	as it appears in the "External application" submenu.  The
//	quantity to the right of the "=" sign is the completely
//	qualified file name of the application.
//
Write=c:\windows\write.exe


[TimeUnitConversions]
//
//	This section of the configuration file contains
//	values that are used to convert one time unit to
//	another with the "Change time unit" filter.  If
//	you're converting minutes to seconds, for instance,
//	you'll be using the "MinutesToSeconds" conversion
//	factor, which will multiply each time between failure
//	or test interval length by 60.  You can modify these
//	conversion factors by using a text editor if you
//	need to.  If you do, make sure that you keep all
//	conversion factors consistent with the environment
//	in which you're doing reliability measurement.  For
//	example, if you change WeeksToHours from 168.0 to 56.0,
//	make sure that you change DaysToHours, MonthsToHours,
```

```
//	and YearsToHours to be consistent with the new value for
//	WeeksToHours.
//
YearsToMonths=12.0
YearsToWeeks=52.0
YearsToDays=365.0
YearsToHours=8760.0
YearsToMinutes=525600.0
YearsToSeconds=31536000.0
MonthsToWeeks=4.33333
MonthsToDays=30.41667
MonthsToHours=730.0
MonthsToMinutes=43800.0
MonthsToSeconds=2628000.0
WeeksToDays=7.0
WeeksToHours=168.0
WeeksToMinutes=10800.0
WeeksToSeconds=604800.0
DaysToHours=24.0
DaysToMinutes=1440.0
DaysToSeconds=86400.0
HoursToMinutes=60.0
HoursToSeconds=3600.0
MinutesToSeconds=60.0

//	This section identifies the directory in which the
//	help files are found.
//
[HelpDir]
Directory=c:\windows\

//	This section gives the values for the scaling factor
//	used to control the size of the printout when printing
//	a plot from a saved file.
//
[PlotScaleFactor]
ScaleFactor = 0.8

//	This section gives the significance values for the goodness
//	of fit tests for each type of data.  After running a set of models,
//	the goodness of fit tests that are run will indicate whether the
//	model results fit the data at the significance levels indicated
//	in this section.  These significance levels can be changed
//	through the main window's "Setup" menu.  The default
```

```
//	significance levels are 5%, as shown below.
[GOFSignificance]
TBESignificance=5.00
FCSignificance=5.00

//	This section gives the significance levels for the Laplace
//	Test for each type of data.  When performing the Laplace
//	Test, the results will indicate the value of the test statistic
//	for which the null hypothesis (occurrences of failures follow
//	a Homogeneous Poisson Process) can be rejected at the
//	specified significance level. These significance levels can
//	be changed through the main window's "Setup" menu.
//	The default significance levels are 5%, as shown below.
[LaplaceTestSignificance]
TBESignificance=5.00
FCSignificance=5.00

//	This section specifies the maximum length of the list of filenames
//	kept in the File menu as well as the names of those files.  Each of
//	the filenames in this section will be placed in the list at the bottom
//	of the File menu – the file can then be opened simply by picking its
//	name off of the File menu's list.
[RecentFiles]
NMostRecent=5

//	The following section describes each of the combination
//	definitions in greater detail.  For each definition, the
//	name is given in square brackets (e.g. [ELC]).  The
//	number of components in the combination is given by the
//	value of "Modelcount".  For dynamically weighted combin-
//	ations, the type of re-evaluation window is specified
//	by the "WindowType" field, which will be either "Fixed"
//	or "Sliding" (the default value).  For statically weighted
//	combinations, this field will be "None".  For dynamically-
//	weighted models, the size of the re-evaluation window is
//	given by the value of "WindowSize".  For statically weighted
//	combinations, this value is 0.  Each of the components of
//	the combination is then identified and described, in the
//	same fashion as it was in the "[models]" section.
//
//	Finally, weights for each component are given in the
//	"Weights" section.  In the ELC model, which is a
//	statically weighted combination, the weight of the
//	Littlewood-Verrall model is given by the first value
```

```
//	in the "Weights" list, the weight for the Musa-Okumoto
//	model is given by the second value in the "Weights" list,
//	and the weight for the Nonhomogeneous Poisson Process model
//	for time between failures data is given by the third value
//	in the "Weights" list.  For dynamically weighted combin-
//	ations in which weights are assigned based on ranking model
//	results, the weight associated with each rank is given in
//	the "Weights" list.  For example, consider the ULC
//	combination.  The component model having a rank of 1 at
//	the end of a re-evaluation interval would receive a weight
//	of 1, the component model having a rank of 2 would receive
//	a weight of 4, and the component model having a rank of
//	3 would receive a weight of 1.  In particular, suppose that
//	at the end of the fourth re-evaluation interval (which in
//	this case is 4 observations, given by the WindowSize value),
//	the Littlewood-Verrall predicted the smallest time to the
//	next failure.  This model would then get a rank of 1, and
//	receive a weight of 1 in the combination.
//
//	The "Weights" list for dynamically weighted models in
//	weights are computed and assigned based on changes in
//	the prequential likelihood statistic is a single-item list,
//	and the value of that one item is 0.
//
[ELC]
Modelcount=3
WindowType=None
WindowSize=0
Quadratic LV=000
Musa-Okumoto=000
NHPP (TBE)=000
Weights=1,1,1

[ULC]
Modelcount=3
WindowType=Sliding
WindowSize=4
Quadratic LV=000
Musa-Okumoto=000
NHPP (TBE)=000
Weights=1,4,1

[MLC]
Modelcount=3
```

WindowType=Sliding
WindowSize=4
Quadratic LV=000
Musa-Okumoto=000
NHPP (TBE)=000
Weights=0,1,0

[DLC/S/4]
Modelcount=3
WindowType=Sliding
WindowSize=4
Quadratic LV=000
Musa-Okumoto=000
NHPP (TBE)=000
Weights=0

Appendix E. Model Timing Information

This appendix contains information on how long it takes the software reliability models in CASRE to run for some of the sample data sets included on the distribution diskette. The file "fc_test.dat" was used for failure count models, and the files "tbe_tst2.dat", "s1.dat", s2.dat", and "s3.dat" were used for the time between failure models, including the permanent combinations. The table below gives the modeling start point, the modeling end point, and the data range used for the initial parameter estimates for each data set, as well as the time to run each model. All models were run using maximum likelihood parameter estimation.

Failure data File / Model	fc_test.dat First point: 1 Last pt: 27 Init parms: 1-13	tbe_tst2.dat First pt: 100 Last pt: 194 Init parms: 100-150	s1.dat First pt: 1 Last pt: 136 Init parms: 1-68	s2.dat First pt: 1 Last pt: 86 Init parms: 1-43	s3.dat First pt: 1 Last pt: 207 Init parms: 1-103
Generalized Poisson	4 s	N/A	N/A	N/A	N/A
Generalized Poisson with ML estimate of inteval weighting	4 s	N/A	N/A	N/A	N/A
Schick-Wolverton	4 s	N/A	N/A	N/A	N/A
Nonhomogeneous Poisson (failure counts data)	3 s	N/A	N/A	N/A	N/A
Schneidewind – all intervals	2 s	N/A	N/A	N/A	N/A
Schneidewind – omit first s-1 intervals	2 s	N/A	N/A	N/A	N/A
Schneidewind – total first s-1 intervals	2 s	N/A	N/A	N/A	N/A
Yamada S-Shaped	2 s	N/A	N/A	N/A	N/A
Geometric	N/A	10 s	22 s	8 s	40 s
Jelinski-Moranda	N/A	5 s	8 s	4 s	12 s
Linear Littlewood-Verrall	N/A	160 s	195 s	150 s	370 s
Quadratic Littlewood-Verrall	N/A	160 s	195 s	150 s	370 s
Musa Basic	N/A	3 s	4 s	3 s	6 s
Musa-Okumoto	N/A	4 s	7 s	4 s	12 s
Nonhomogeneous Poisson (interfailure times data)	N/A	5 s	4 s	2 s *	3 s *

Failure data File / Model	fc_test.dat First point: 1 Last pt: 27 Init parms: 1-13	tbe_tst2.dat First pt: 100 Last pt: 194 Init parms: 100-150	s1.dat First pt: 1 Last pt: 136 Init parms: 1-68	s2.dat First pt: 1 Last pt: 86 Init parms: 1-43	s3.dat First pt: 1 Last pt: 207 Init parms: 1-103
For the four models below, the given timings assume that the component models have already been run					
DLC/S/4	N/A	3 s	4 s	-	-
ELC	N/A	3 s	4 s	3 s *	3 s *
MLC	N/A	3 s	4 s	-	-
ULC	N/A	3 s	4 s	-	-

In this table, each entry indicates the number of seconds it took for a model to run with a particular data set. An asterisk after the timing indicates that valid estimates were not produced, a "-" indicates that the combination model could not be run (parameter estimates for one or more of the components might not have converged, for example), and a "N/A" indicates that a model is not applicable to the type of data contained in a failure data file.

The times given in this table were obtained by running the models on a Compudyne 4DX/33 computer. This machine has an 80486DX CPU, 8MB of RAM, and a clock rate of 33MHz. For the failure count data file, the number of future intervals for which failure counts were predicted was 15. For time between failures data sets, the number of future failures for which interfailure times were predicted was 20.

Appendix F. Converting dBASE Files to ASCII Text

When you're making input data files for CASRE, it is likely that you'll be converting database files to ASCII text files, then opening those text files within CASRE. In this appendix, we'll demonstrate how you'd convert a dBASE III or dBASE IV file to a delimited ASCII text file that you could open in CASRE. We'll make the following assumptions about the database file:

1. The database file is already in the format spelled out in Appendix 6. That is to say, if you're working with failure count data, among the fields in the database file are those required by the failure count format given in Appendix 6. The same applies to time between failures data.

2. All of the queries of the database that need to be done to produce the file mentioned in point 1 have already been done. It may be the case that your problem tracking database is distributed among two or more data tables. **We don't address the issue of how to query the database in this user's guide, since this would get into a discussion of how to design a database. Such a discussion is beyond the scope of this user's guide.**

We'll give an example of how to make a delimited ASCII text file for both failure count data and time between failures data.

Failure Count Data

We'll start out by assuming that you have a dBASE III or dBASE IV file in the following format:

```
Structure for database: C:\DOCUMENT\SMERFSUI\FC_TEST.DBF
Number of data records:       0
Date of last update   : 08/25/93
        Field   Field Name      Type        Width   Dec   Index
        1       INTERVAL        Numeric     5             N
        2       FAILURES        Float       10      4     N
        3       INT_LENGTH      Float       10      4     N
        4       SEVERITY        Numeric     1             N
        5       TESTER          Character   30            N
** Total **             56
```

This is the format for a dBASE IV file. For dBASE III, the Float field types would also be Numeric. The first field, "INTERVAL", identifies the test interval, and starts with a value of 1. The FAILURES field counts the number of failures observed during INTERVAL having a severity specified in the SEVERITY field. This means that for any particular INTERVAL, you can have one or more records. For instance, in INTERVAL 5 you might have 6 FAILURES of SEVERITY 1 and 2 FAILURES of SEVERITY 4. This would constitute two separate records for INTERVAL 5. SEVERITY identifies the severity of failures observed during a particular INTERVAL, and can have values between and including 1 and 9.

You'll need the first four fields of this particular dBASE file to create a failure count data file. Assuming that you want to write the contents of the complete file to disk, you'd start dBASE, open up the database file, and enter the following dBASE command at the dot prompt:

copy to <filename> type delimited with blank fields interval, failures, int_length, severity<CR>

This command would copy the contents of the entire database file to the file identified by <filename>, and the fields in the resulting ASCII text file would be separated by blanks. All that you have to do after the ASCII file has been produced is use your favorite text editor and insert a new first line. In this first line, you'll enter one of the seven keywords given in Appendix 6 specifying the time units for the lengths of the test intervals (e.g. minutes, hours). After putting in this new first line, you'll be done after you save the file. **If your text editor gives you a choice of formats in which to open and save the file (e.g. Word, WordPerfect), open it as an ASCII text file, and save it as an ASCII text file when you're done.**

If you wanted to create a file that does not have the first 200 test intervals for some reason, you'd follow the same steps except that after opening the file, you'd enter the following commands:

1. go top<CR>

2. replace interval with (interval-200) while .not. eof()<CR>

   This second command makes sure that the file you create starts with a test interval number of 1 instead of 201. This is required to comply with the failure counts data format described in Appendix 6.

   Note that you'll be changing the database file in this step. Suppose that the name of the file is "test.dbf". If you don't want to change this file, first copy it to a new file, say "test_tmp.dbf" before starting dBASE. After you start dBASE, then work with "test_tmp.dbf" instead of "test.dbf".

3. copy to <filename> type delimited with blank fields interval, failures, int_length, severity for interval .ge. 1<CR>

After completing the steps above, you'll have produced a delimited ASCII text file containing data from interval number 201 through the end of the file. In the text file, interval number 201 would be identified as interval 1, interval 202 in the database file would be identified as interval number 2 in text file, and so forth. To complete the creation of the text file so that it can be used by CASRE, remember to add a new first line as described above.

Time between failures

Now suppose that you have a database file of time between failures data in the following format:

Structure for database: C:\DOCUMENT\SMERFSUI\TBF_TEST.DBF
Number of data records: 0
Date of last update : 08/25/93

Field	Field Name	Type	Width	Dec	Index
1	ERROR_NUM	Numeric	5		N
2	TIME2FAIL	Float	10	3	N
3	SEVERITY	Numeric	1		N
4	FAIL_DATE	Date	8		N
5	PROB_DESC	Character	254		N

** Total ** 279

You'll need the first three fields to make a times between failures data set according to the format given in Appendix 6.

To create a blank-delimited ASCII text file that CASRE can work with, you'll first start dBASE and open up the database file that you want to work with. If you want to put the entire file into the ASCII text file, you'll enter the following command at the dot prompt:

copy to <filename> type delimited with blank fields error_num, time2fail, severity<CR>

This will produce an ASCII text file for time between failures data with blanks separating the fields. To complete production of a text file that CASRE can use, remember to add a new first line as described above for failure count data.

Suppose you want to produce a text file that ignores the first 50 errors in the database file. Start dBASE, open the file you want to use, and enter the following commands at the dot prompt:

1. go top

2. replace error_num with (error_num-50) while .not. eof()<CR>

   This second command makes sure that the file you create starts with an error number of 1 instead of 51. This is required to comply with the time between failures data format described in Appendix 6.

The note given above about modifying the database file for failure counts also applies here. If you don't want to change the master database file, first copy it to a working file before starting dBASE. After starting dBASE, work only with the working file.

3. copy to <filename> type delimited with blank fields error_num, time2fail, severity for error_num .ge. 1<CR>

After completing these steps, you'll have produced an ASCII text file containing time between failures data which omits the first 50 failures in the database file. Error number 51 in the database file will correspond to error number 1 in the text file, error number 52 in the database file will correspond to error number 2 in the text file, and so forth. To complete production of a text file that CASRE can use, remember to add a new first line as described above for failure count data.

Appendix G. Using the Wordpad Editor

Wordpad is an editor that is included with Windows95. You can use Wordpad to modify failure data files. In this appendix, we'll demonstrate a session with Wordpad, modifying one of the failure data files that is included on the CASRE distribution disk. We assume that CASRE has been set up to include Wordpad as one of the applications available on the “External applications” submenu of the “Edit” menu.

To start Wordpad from within CASRE, simply click the mouse on the "Edit" menu in the main window. Holding the left mouse button down, drag the mouse down to the "External applications" menu item, at which point a submenu appears. Still holding the left mouse button down, drag the mouse down to the "Wordpad" menu item in the submenu, then release the left mouse button. At this point the Write window appears, as does a message box. **Don't click on the "OK" pushbutton in the message box until you're done using Wordpad.** Instead, move the mouse onto the Wordpad window, then click the left mouse button. This will bring the Wordpad window to the top of the display, at which point you'll be able to use this application. We say that Wordpad is active at this point.

Once Wordpad is active, you can open a file for editing. We'll choose one of the data files, "c:\casrev3\data\fc_test.dat". Use the mouse to select Wordpad's "File" menu, then select the "Open" menu item. You'll see the dialog box shown in Figure 96 below.

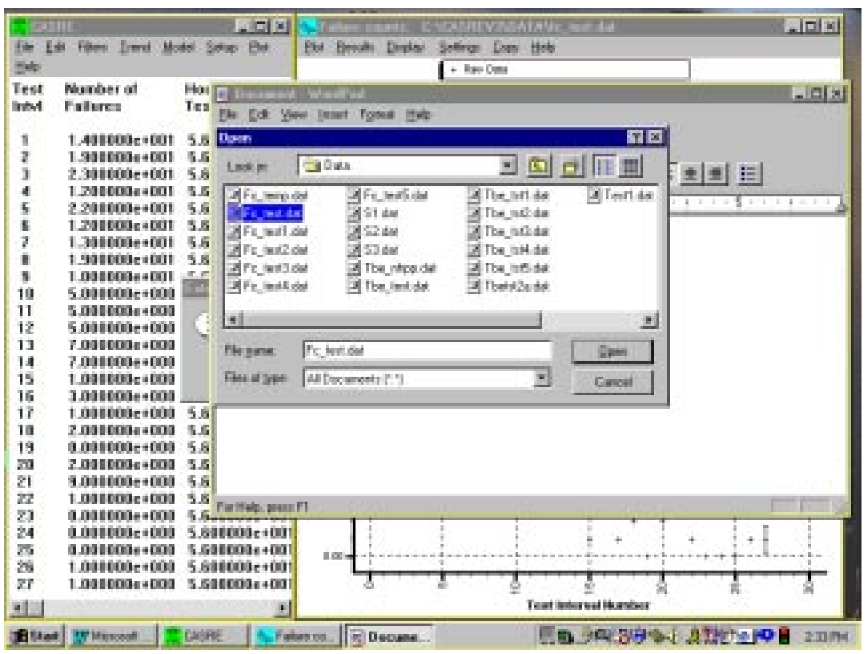

Figure 96 - Opening a file in Wordpad

First you'll have to select the subdirectory in which the file is found. You'll first use the mouse to point and click on the folders in the dialog box’s scrollable window to get to the subdirectory you want to reach. Once you’ve reached the subdirectory, you’ll scroll through the window to find the file, and then double-click on its name. This will open the file for editing, as shown in Figure 97 below.

To start editing the file, simply place the mouse somewhere in the window and click on the left mouse button. The cursor, indicated by a bold vertical line, will appear at the point where you've positioned the mouse. You can use the left, right, up, and down cursor keys on the keyboard to move through the file. You can also use the "Page Up" and "Page Down" keys. You'll find that Wordpad works like most other screen editors. Any input from the keyboard is inserted at the current position of the cursor. You can use the backspace and delete keys to remove characters.

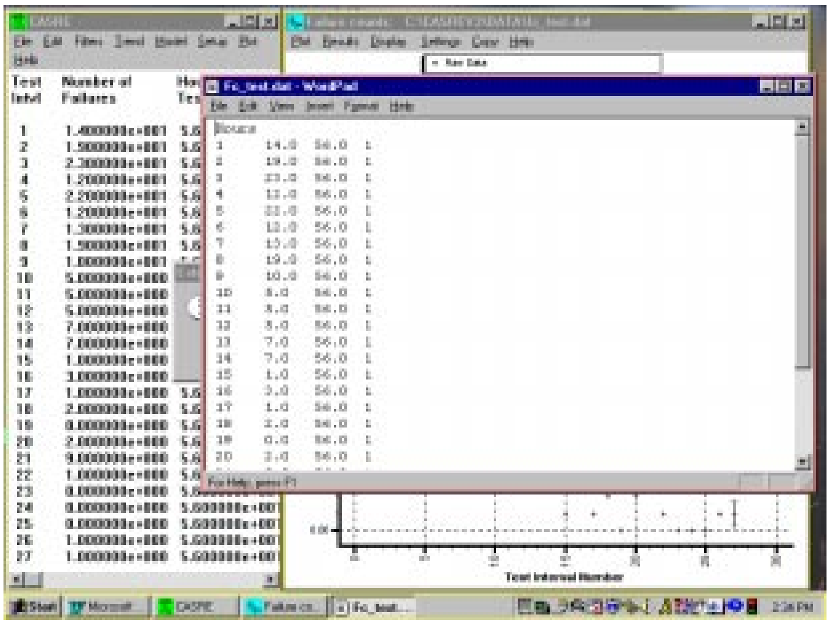

Figure 97 - Contents of file displayed in Wordpad window

You can also highlight blocks of text by using the mouse. Simply use the mouse to position the cursor at the start of the block. Depress the left mouse button, and hold it down as you the move the mouse to where you want the end of the block to be. An example of a highlighted block is given in Figure 98 below. The block is identified by the portion of the window displayed in reverse video. Once you've highlighted a block, you can do several things with it:

1. Delete it by pressing the delete key on the keyboard.

2. Replace it with text by starting to enter input at the keyboard. The highlighted block will be replaced with the characters you enter at the keyboard.

You can also unhighlight the block simply by positioning the mouse in the Wordpad window and clicking the left mouse button somewhere in the window. The block will then be unhighlighted, and the area displayed in reverse video (white on black) will then be displayed in normal video (black on white).

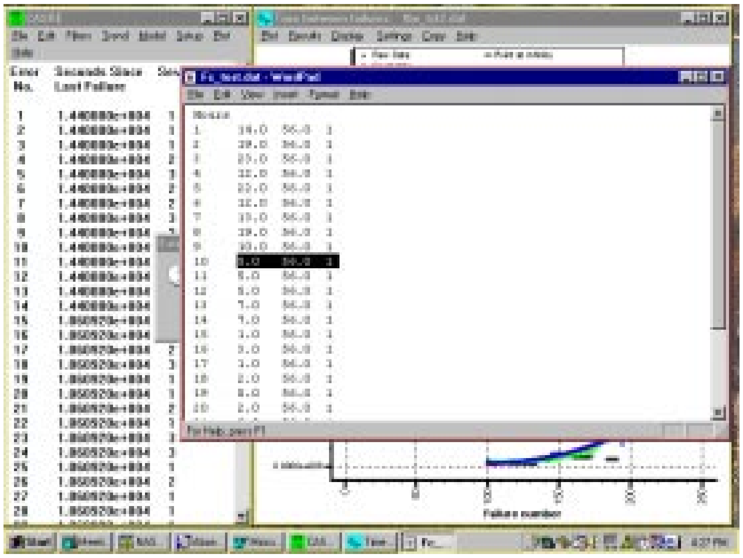

Figure 98 - Highlighting an area of text

After you've made any desired modifications to the file, you can save the changes simply by using the mouse to select the "File" menu, then highlighting the "Save" item in that menu. Since you're working with ASCII text, you'll be asked whether or not you should save the document as text, as shown in Figure 99 below. Make sure to press the "Text Document" button on the message box that comes up. The changes you've made to the file will then be saved. You can then end the Wordpad session by using the mouse to select the "File" menu and then selecting the "Exit" item in that menu.

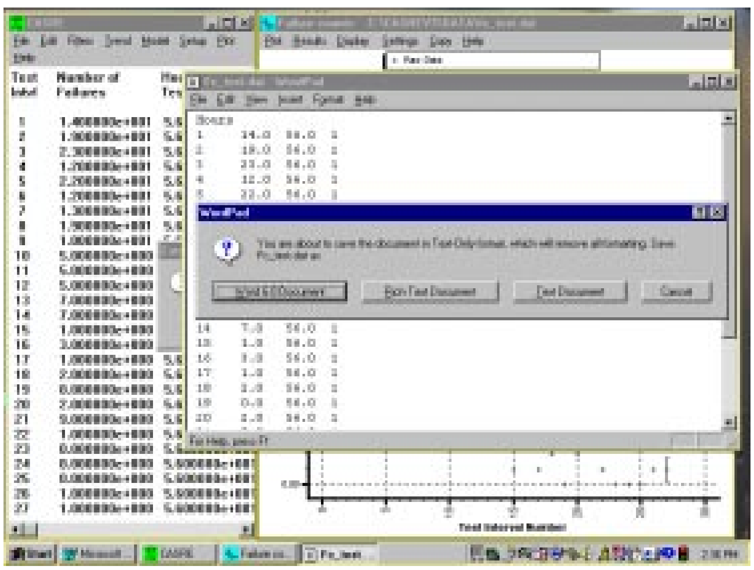

Figure 99 - Wordpad Prompt to Save File as Text

If you've made changes that you don't want to save, use the mouse to select the "File" menu, and then select the "Exit" item. You'll then see a message box asking you whether you want to save your changes - use the mouse to click on the "No" button in the message box if you've made mistakes that you don't want to save.

If you started Wordpad from within CASRE, remember that a message box appeared at the same time that the Wordpad window appeared on screen. After you're finished with Wordpad, the message box will still be there. Use the mouse to click on the "OK" button in that message box so you can proceed with the remainder of your CASRE session.

This has been a very brief introduction to the Wordpad editor. For more information on this editor, as well as on the Notepad editor included with Windows 3.1, Windows 3.11, and Windows95, see the appropriate Windows user's guide.

Appendix H. Using Windows Help

This appendix provides a brief summary on how to use the Help mechanism. Since CASRE's Help facility is built around the Windows Help facility, see the Windows user's guide for more details.

When you choose the "Help index" item in one of CASRE's top-level "Help" menus, a window will appear as shown in Figure 100 below in which you'll see a list of topics that are printed in comparatively large type and are underlined. As you're scrolling through the list of topics, the shape of the mouse cursor will change from an arrow to a hand every time you reach an item for which there is help text. To get information about a particular topic, move the mouse to the name of that topic (scrolling as described in Appendix 10 if necessary), and then click on the name of that topic. The help text for that topic will then be displayed in the window. For many topics, the names of related topics are at the end of the help text. If the names of those related topics are in larger type than the help text, and if they're underlined, you can go directly to a related topic by clicking on the name of the topic.

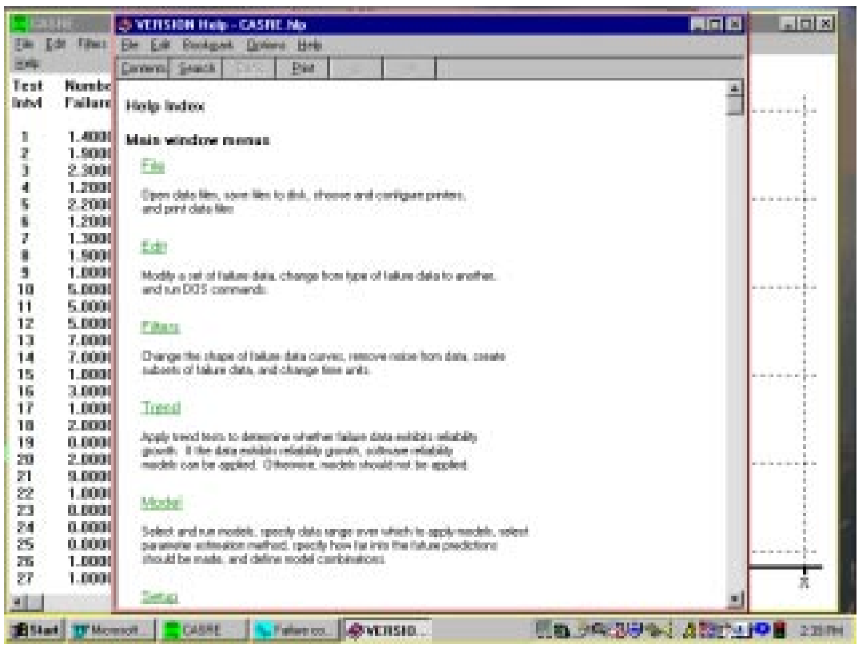

Figure 100 - CASRE Help window

You'll see five buttons across the top of the help window. You'll be concerned mainly with the ones labelled "Contents", "Back", and "History". If you use the mouse to click on the one labelled "Contents", the help index for the currently-active CASRE window will be displayed. If you click on the one labelled "Back", you'll display the most recent help topic you viewed. If you click on the "History" button, you'll display the last 40 help topics you've viewed.

To remove the "Help" window from the screen, use the mouse to select the "File" menu, then select the "Exit" item in that menu.

Appendix I. Model Assumptions

The assumptions made by the models implemented in CASRE are given in this appendix. The assumptions are for atomic models only. These assumptions are taken from the SMERFS help files and user documentation. The table below summarizes the properties of the models.

Model \ Assumptions	Constant Fault Size	Perfect Debugging	Upper Bound on Number of Failures
Geometric	No	No	No
Jelinski-Moranda	Yes	Yes	Yes
Littlewood-Verrall (linear and quadratic versions)	No	No	No
Musa Basic	Yes	Yes	Yes
Musa-Okumoto	Yes	No	No
Nonhomogeneous Poisson Process for Times Between Failures	Yes	Yes	Yes
Generalized Poisson (include Schick-Wolverton)	Yes	Yes	Yes
Nonhomogeneous Poisson Process for Failure Counts	Yes	Yes	Yes
Schneidewind (all three variants)	Yes	Yes	Yes
Yamada S-Shaped	No	Yes	Yes

Notes: Each model assumes that the software usage during test is similar to that during operations. Each model also assumes that detections of failures are independent of each other. Finally, the Schneidewind model assumes that test intervals are all the same length.

Time between failure models assumptions

Geometric model:

1. The software is operated in a similar manner as the anticipated operational usage.

2. There is no upper bound on the total number of failures (i.e., the program will never be error-free).

3. All faults do not have the same chance of detection.

4. The detections of faults are independent of one another.

5. The failure detection rate forms a geometric progression and is constant between failure occurrences.

Jelinski-Moranda model:

1. The software is operated in a similar manner as the anticipated operational usage.

2. The rate of failure detection is proportional to the current fault content.

3. All failures are equally likely to occur and are independent of each other.

4. Each failure is of the same order of severity as any other failure.

5. The failure rate remains constant over the interval between failure occurrences.

6. Faults are corrected instantaneously without introduction of new faults into the program.

7. The total number of failures expected to be seen has an upper bound.

Littlewood-Verrall model (Linear and Quadratic versions):

1. The software is operated in a similar manner as the anticipated operational usage.

2. Successive time-between-failures are independent random variables, each having an exponential distribution. The distribution for the i-th failure has a mean of 1/lamda(i).

3. The lamda(i) form a sequence of independent variables, each having a gamma distribution with the parameters alpha and phi(i). Phi(i) has either the form:

$$\text{beta}(0) + \text{beta}(1) * i \quad \text{(linear)}$$
or
$$\text{beta}(0) + \text{beta}(1) * i^2 \text{ (quadratic)}$$

In other words, the program may get less reliable as testing progresses if more faults are inserted into the program than are removed during fault correction. This is the model's way of trying to take into account the "goodness" of the parameter.

4. There is no upper bound on the total number of failures (i.e., the program will never be error-free).

Musa Basic model:

1. The software is operated in a similar manner as the anticipated operational usage.

2. The detections of failures are independent of one another.

3. All software failures are observed (i.e., the total number of failures has an upper bound).

4. The execution times (measured in cpu time) between failures are piecewise exponentially distributed.

5. The hazard rate is proportional to the number of faults remaining in the program.

6. The fault correction rate is proportional to the failure occurrence rate.

7. Perfect debugging is assumed.

Musa-Okumoto model:

1. The software is operated in a similar manner as the anticipated operational usage.

2. The detections of failures are independent of one another.

3. The expected number of failures is a logarithmic function of time.

4. The failure intensity decreases exponentially with the expected number of failures experienced.

5. There is no upper bound on the number of total failures (i.e., the program will never be error-free).

Non-Homogeneous Poisson Process model:

1. The software is operated in a similar manner as the anticipated operational usage.

2. Every failure has the same chance of being detected as any other failure.

3. The cumulative number of failures detected at any time follows a Poisson distribution with mean m(t). That mean is such that the expected number of failures in any small time interval about t is proportional to the number of undetected failures at time t.

4. The mean is assumed to be a bounded non-decreasing function with m(t) approaching in the limit, "a" (the expected total number of failures to be, eventually, detected in the testing process), as the length of testing goes to infinity.

5. Perfect debugging is assumed.

Failure counts models assumptions

Generalized Poisson model (includes Schick-Wolverton model):

1. The software is operated in a similar manner as the anticipated operational usage.

2. The expected number of failures occurring in any time interval is proportional to the fault content at the time of testing, and to some function of the amount of time spent in failure testing.

3. All failures are equally likely to occur and are independent of each other.

4. Each failure is of the same order of "severity" as any other failure.

5. Faults are corrected at the ends of the testing intervals, without introducing new faults.

Non-Homogeneous Poisson model for failure counts data:

1. The software is operated in a similar manner as the anticipated operational usage.

2. The number of faults detected in each of the testing time periods are independent.

3. Every failure has the same chance of being detected as any other failure.

4. The cumulative number of failures detected at any time follows a poisson distribution with mean m(t). That mean is such that the expected number of failures in any small time interval about t is proportional to the number of undetected failures at time t.

5. The mean is assumed to be a bounded non-decreasing function with m(t) approaching in the limit, "a" (the expected total number of failures to be, eventually, detected in the testing process), as the length of testing goes to infinity.

6. Perfect debugging is assumed.

Schneidewind model (all three variants):

1. The software is operated in a similar manner as the anticipated operational usage.

2. All failures are equally likely to occur and are independent of each other.

3. The fault correction rate is proportional to the number of faults to be corrected.

4. The mean number of detected failures decreases from one testing interval to the next. The total number of failures expected to be seen has an upper bound.

5. All testing periods are of the same length.

6. The rate of fault detection is proportional to the number of faults within the program at the time of test. The failure detection process is a non-homogeneous poisson process with an exponentially decreasing failure rate.

7. Perfect debugging is assumed.

Yamada S-shaped model:

1. The software is operated in a similar manner as the anticipated operational usage.

2. A software system is subject to failures at random caused by faults present in the system.

3. The initial fault content of the software system is a random variable.

4. The time between failures (k - 1) and k depends on the time to failure (k - 1).

5. Each time a failure occurs, the fault which caused it is immediately removed, and no other faults are introduced.

6. The total number of failures expected to be seen has an upper bound.

Appendix J. Submitting Change Requests

To suggest changes to CASRE, please supply the following information to the author. Although the author can be reached via surface mail, e-mail, or fax, e-mail is preferred. Change requests will be prioritized based on their perceived utility and available funding. Notification to users of new versions of CASRE will be through TBD; documentation for each new version will include a summary of the change requests that were incorporated.

1. **Requester's name, address, e-mail address, telephone number, and fax number** - We ask for this information in case we need to get in touch with you for more details about your recommendation.

2. **CASRE version number** - specify the version number of your copy of CASRE here. You can find this version number in the "About" dialog box in the main window. There are two reasons we're asking for this information:

   a. If you're using an older version of CASRE, it may be the case that the change you're requesting has already been implemented in the current version. In that case, we can let you know.

   b. If you're using the current version, and the change you're requesting has already been implemented, we may need to think about writing a better description of that capability in the user's guide.

3. **Suggested change** - Describe the change or addition you'd like to see in CASRE. Describe it in as much detail as you can. If appropriate, please reference it to existing CASRE capabilities, and state why you feel the change is important. You might want to describe the situations in which you'd use the new features and how often you encounter those situations.

Appendix K. Prequential Likelihood

In this appendix we examine prequential likelihood in somewhat greater detail than we did in paragraph 4.9.2. Recall that the equation for prequential likelihood is:

$$PLR_n = \frac{p(t_{j+n},\dots,t_{j+1} \mid t_j,\dots,t_1,A)}{p(t_{j+n},\dots,t_{j+1} \mid t_j,\dots,t_1,B)} = \frac{PL_n^A}{PL_n^B}$$

which can be reduced by using Bayes' rule to:

$$PLR_n = \frac{p(A \mid t_{j+n},\dots,t_1)}{p(B \mid t_{j+n},\dots,t_1)} = \frac{w_A}{1 - w_A}$$

This is the posterior odds ratio, where w_A is the posterior belief that A is true after making predictions with both A and B and comparing them with actual behavior. If $PLR_n \rightarrow \infty$ as $n \rightarrow \infty$, model B is discarded in favor of model A.

As seen in the first equation, we are predicting a series of future events ($t_{j+1},t_{j+2},\dots,t_{j+n}$) based on previous actual events ($t_1,t_2,\dots,t_j$). Suppose we're considering two software reliability models, A and B. We want to determine which one is more likely to produce accurate predictions. The initial portion of the data range, represented by ($t_1,\dots,t_j$) are the <u>actual</u> failure times used to estimate the parameters for each of the models A and B. The remainder of the range, represented by ($t_{j+1},t_{j+2},\dots,t_{j+n}$), are the <u>predicted</u> failure times used for models A and B. Using Bayes' rule, the probability of model A or model B being correct based on historical/actual data can be combined with the predictive/conditional probability of model A or model B being correct to give the probability that model A or model B is correct based on the combination of both actual and predictive (posterior) probability. Figure 101 on the next page is a depiction of this combination of probabilities for <u>actual</u> and <u>predicted</u> failure times for models A and B, where:

P(A)	the probability that actual results are best described by model A.
P(C|A)	the probability that model A will give the predicted values $t_{j+1},t_{j+2},\dots,t_{j+n}$.
P(¬C|A)	the probability that model A will not give the predicted values $t_{j+1},t_{j+2},\dots,t_{j+n}$.
P(C)	the probability that model A or model B will give the predicted values $t_{j+1},t_{j+2},\dots,t_{j+n}$.
P(A)	the probability that model A best describes the actual $t_1,t_2,\dots,t_j$ and the predicted $t_{j+1},t_{j+2},\dots,t_{j+n}$.

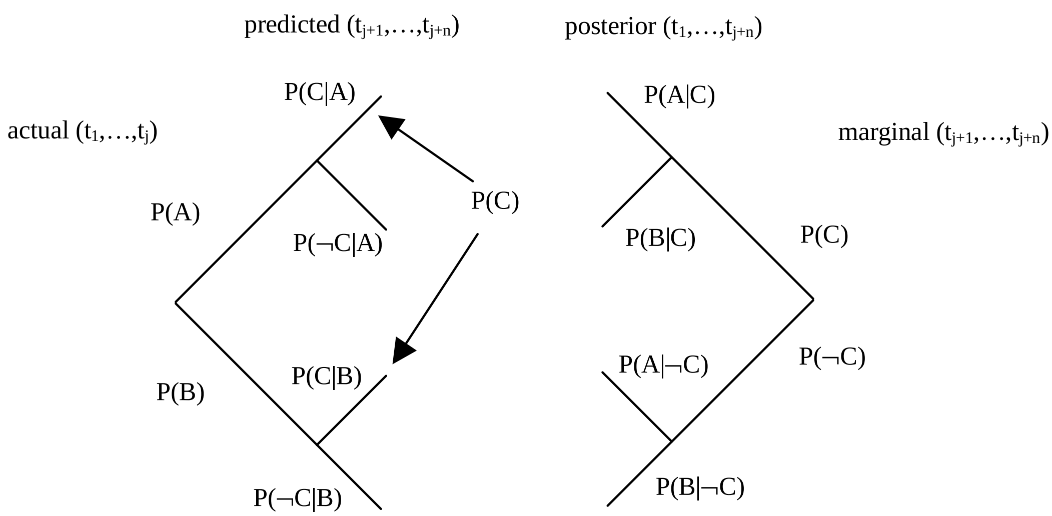

Figure 101 – Bayesian Inference of Model Applicability

We can determine the probabilities as follows:

$$P(A \mid C) = \frac{P(A)P(C \mid A)}{P(C)} \Rightarrow P(C \mid A) = \frac{P(C)P(A \mid C)}{P(A)}$$

$$P(B \mid C) = \frac{P(B)P(C \mid B)}{P(C)} \Rightarrow P(C \mid B) = \frac{P(C)P(B \mid C)}{P(B)}$$

$$PLR_n = \frac{P(C \mid A)}{P(C \mid B)} = \frac{\dfrac{P(C)P(A \mid C)}{P(A)}}{\dfrac{P(C)P(B \mid C)}{P(B)}} = \frac{P(t_1 \ldots t_j, B)P(t_1 \ldots t_{j+n}, A)}{P(t_1 \ldots t_j, A)P(t_1 \ldots t_{j+n}, B)}$$

Appendix L. Copying Windows to the Clipboard

In preparing software reliability reports, you might want to include one or more CASRE plots. In many instances, you can take snapshots of the screen or of the active window and paste the snapshots directly into your report. Otherwise, you can use the capabilities of the Windows clipboard and the PC Paintbrush that comes with WindowsNT, Windows95, and Windows 3.1 to make intermediate files that can be copied into your report. If you need to make intermediate files, follow these steps to create bitmap or .PCX files that you can bring into your word processor. We assume that CASRE has been started, and that the graphic display window is showing the failure data or model results that you want to see.

1. Move the mouse cursor to the graphic display window, and click the left button mouse to make the graphic display window the current window.

2. Press the "Alt" and "Prt Scrn" keys simultaneously. This will copy the graphic display window to the clipboard.

3. In the program group called "Main" or "Accessories", you should be able to find an icon labelled either “Paint” or "Paintbrush". This is the icon for the "Paintbrush" program.

   Double click on the "Paintbrush" icon. This will start the Paintbrush program.

4. You'll probably want to adjust the size of the "Paintbrush" window. You can do this by:

   a. Moving the mouse cursor to the button in the upper right hand corner that has an up-arrow symbol (Windows 3.1) or two somewhat overlapping window symbols (Windows95 and Windows NT) on it, and then clicking the left mouse button once,

   or

   b. Move the mouse cursor until it touches the border of the "Paintbrush" window and turns into a double-headed arrow. Hold down the left mouse button and move the cursor to change the size of the window as described in paragraph 4.3.1.

   You'll want to experiment with the size of the "Paintbrush" window until you find a size that you like.

5. Move the mouse cursor into the paintbrush window, then press the "Ctrl" and "V" keys simultaneously. This will copy the contents of the clipboard into the Paintbrush window.

6. Now you can save the file as either a bitmap or ".PCX" file. To do this, you'll use the "Save as..." menu item in Paintbrush's "File" menu. Selecting this menu item will bring up the dialog box shown in Figure 102 on the next page.

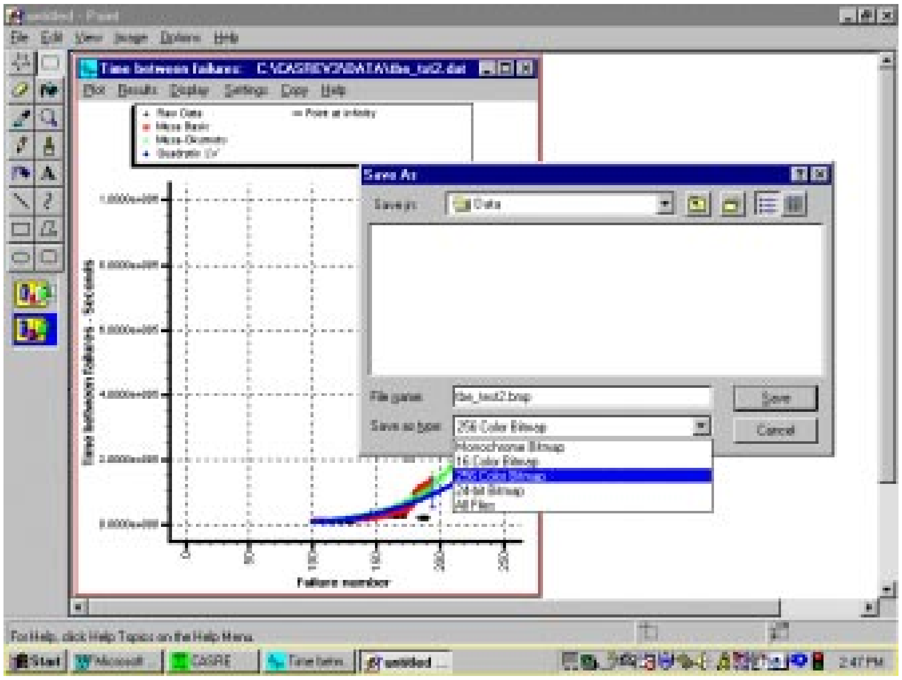

Figure 102 - Saving a file in PC Paintbrush

You can enter the name of the file in the "File Name:" edit area in the dialog box. Just move the mouse cursor to that area, click the left mouse button, and you can enter the file name from the keyboard.

You can save the file as a bitmap (".BMP") or as a ".PCX" file (Windows 3.1). The "PCX" file is a format specific to PC Paintbrush which occupies less disk space than a bit map, and can be read by some word processors. In Windows 3.1, you use the controls in the area labelled "Save File as Type" in the lower left corner of the dialog box. In Windows95 and WindowsNT, just select the “Save as” item from the “File” menu, and pick the file type from the list shown in Figure 102. Just scroll through the list of file types until you find the one you want, then click the mouse on that entry. **If space is any consideration, we recommend using either 16 Color bitmaps or PCX files (if available).**

7. After you've named the file and picked the file type, click the mouse on the "OK" button in the upper right corner of the dialog box. This will save the image to the disk file you've specified. You can then use your word processor to bring that particular file into your document.

You can learn more about the Paintbrush program by consulting your Windows 3.1, Windows 3.11, Windows95, or WindowsNT User's Reference.