A Systematic and Comprehensive Tool for Software Reliability Modeling and Measurement

Michael R. Lyu	Allen P. Nikora	William H. Farr
Applied Research Area	Jet Propulsion Laboratory	Strategic Systems Department
Bell Communications Research	California Institute of Technology	Naval Surface Warfare Center

Abstract

Sufficient work has been done to demonstrate that software reliability models can be used to monitor reliability growth over a useful range of software development projects. However, due to the lack of appropriate tools, the application of software reliability models as a means for project management is not as widespread as it might be. The existing software reliability modeling and measurement programs are either difficult for a non-specialist to use, or short of a systematic and comprehensive capability in the software reliability measurement practice. To address the ease-of-use and the capability issues, we have prototyped a software reliability modeling tool called CASRE, a Computer-Aided Software Reliability Estimation tool. Implemented with a systematic and comprehensive procedure as its framework, CASRE will encourage more widespread use of software reliability modeling and measurement as a routine practice for software project management. This paper navigates through the CASRE tool to demonstrate its functionality and capability.

1: Introduction

Software reliability is increasingly the determining factor of overall system reliability. This trend makes the estimation of software reliability more and more critical to most large-scale projects in their reliability engineering aspects. Traditionally, software reliability modeling is a set of techniques that apply probability theory and statistical analysis to predict the reliability of software products, both quantitatively and objectively. This approach is similar to that which has been applied to hardware reliability; however, software failure process is quite different from that of hardware. A software reliability model specifies the general form of the dependence of the failure process on the principal factors that affect it: fault introduction, fault manifestation, failure detection and recovery, fault removal, and operational environment.

The primary goal of software reliability models is to assess current reliability and forecast future reliability, based on rational assumptions for the application of statistical inference techniques to the observed failure data. A major difficulty in software reliability engineering practice is, however, to analyze the particular context in which reliability estimation is to take place so as to decide *a priori* which model is likely to be trustworthy. Due to the intricacy of human activities involved in software development and operation process, as well as the uncertain nature of software failure patterns, such a priori determinations have never been conclusive. It has been shown that there is no best software reliability model for every case under all circumstances. As a result, practitioners are left in a dilemma as to which software reliability models to choose, which procedures to apply, and which prediction results to trust, while contending with varying software development and operation practices.

Since the engagement and application of software reliability models and the evaluation and interpretation of model results involve tedious computation-intensive tasks, we believe the only practical usage of reliability models is through software tools. For this purpose, we design a software reliability modeling tool, called Computer-Aided Software Reliability Estimation (CASRE) system [1], for an automatic and systematic approach in estimating software reliability.

2: Objective of the software tool

Several software reliability tools are currently available for users to apply one or more of the known software reliability models to a development effort [2]. Popular tools include Statistical Modeling and Estimation of Reliability Functions for Software (SMERFS), and Software Reliability Modeling Programs (SRMP). In addition to allowing users to make reliability estimates, these tools also allow users to determine a particular model's applicability to a set of failure data.

A major issue in modeling software reliability, however, lies in the ease-of-use of currently available tools. Nearly all current tools have command-line interfaces, and do not take advantage of the high-resolution displays that would allow the construction of menu-driven or direct-manipulation user interfaces. Command-line interfaces can make it more difficult for users to remember the specific steps required to accomplish a task, while a menu-driven interface can be set up to guide users through the necessary steps. Finally, most currently available tools display their outputs in tabular form. Although some tools provide a graphical display of results, in most cases this is of limited utility in that the display resolution is low since character-based graphics are used in painting the screen. In addition, the variety of results that may be graphically displayed is usually limited to interfailure times or failure intensities. In measuring software reliability, it is useful to see high-resolution displays of these quantities, as well as cumulative number of errors, the reliability growth curve, and the results of statistical methods used to determine whether the model being executed is appropriate for the current project.

CASRE is implemented as a software reliability modeling tool that addresses the ease-of-use issue as well as other issues. Figure 1 shows the high-level architecture for CASRE. CASRE is currently executed in a DOS Windows environment. The command interface is menu driven; users are guided through the selecting of a set of failure data and executing a model by selectively enabling pull-down menu options. Modeling results are also presented in a graphical manner. After one or more models have been executed, the predicted failure intensities or interfailure times are drawn in a graphical display window. Users can manipulate this window's controls to display the results in a variety of ways, including cumulative number of failures and the reliability growth curve. Users may also display the results in a tabular fashion if they wish.

In addition, CASRE is facilitated with a useful functionality. Namely, results from different models can be combined in various ways to yield reliability estimates whose predictive quality is better than that of the individual models themselves [3]. CASRE incorporates our findings that prediction accuracy may be increased by combining the results of several models in a linear fashion. Moreover, CASRE allows users to define their own combinations and record them as part of the tool's configuration. Weights for the components of the combination may be static or dynamic, and may be based on statistical techniques used to determine the applicability of a model to a set of failure data. Once combination models have been defined, the steps required to execute them are no different than executing a simple model.

3: CASRE Context

CASRE comprises seven major functional areas:

1. File operations ("File" menu)
2. Editing operations ("Edit" menu)
3. Transformation and smoothing ("Filter" menu)
4. Model selection and application ("Model" menu)
5. Program configuration ("Setup" menu)
6. Help system ("Help" menu)
7. Graphics display window

3.1: File operations

There are six operations that can be performed by selecting items from the File pull-down menu. These are: "Open", "Save", "Save as", "Setup printer", "Print", and "Exit". Their definition and usage follow the usual conventions in modern user-interface designs.

The File menu appears underneath the File button on the main menu bar when the File button is selected and clicked. As long as the File button remains selected, users can navigate through the File menu. Menu item selection is accomplished by pointing to an item in the menu and then releasing the mouse button.

3.2: Editing operations

Users will have available to them four editing operations which allow them to modify the currently displayed work space. These are:

1. Undo – allows users to undo the most recent CASRE operation.
2. Change data type – allows users to convert failure data in the form of interfailure times to failure counts and test interval lengths, and vice versa.
3. External application – allows users to invoke a preferred text editor, word processor, or other application from within CASRE. The application is selected from a user-configurable menu.
4. Escape to DOS – allows users to temporarily escape to DOS and execute DOS commands. Users re-enter CASRE by typing "exit" at the DOS prompt.

3.3: Transformation and smoothing operations

Filtering operations allow users to perform global modifications to the failure data displayed in the work space. Since failure data is frequently noisy, and may be noisy to the point that it is difficult or impossible to estimate model parameters, users may wish to apply a smoothing operation to the data to remove some of the noise. Users may also wish to apply other transformations which will allow them to change the shape and position of the failure data. For instance, one of the transformations can change exponential data to a straight line.

There are three types of filtering operations; these are transformation, smoothing, and data subsetting. Multiple transformation and smoothing filters may be applied to the data in a pipeline fashion. There is also a "Undo all filters" facility allowing users to remove all of the filters that have been applied to the data. The filtering capabilities include: (1) Affine transformations (scaling, power, logarithmic, exponential); (2) Smoothing (to use "Hann window"); (3) Subset data (to select severity); and (4) Remove all filters.

3.4: Model selection and application

CASRE allows users to select and apply existing software reliability models to the data displayed in the work space. These models include: (1) Brooks and Motley Model; (2) Geometric Model; (3) Goel-Okumoto; (4) Jelinski-Moranda; (5) Littlewood Model; (6) Littlewood-Verrall; (7) Musa-Okumoto; (8) Generalized Poisson Model; (9) Schneidewind Model; and (10) Yamada Delayed S-Shape Model.

Users are also allowed to define combinations of existing models, edit specifications of user-defined models, and remove user-defined models from the menu of available models. Users may also specify the parameter estimation method, the confidence bounds that will be reported for model parameters, and the amount of time into the future for which reliability predictions will be made. The Model menu items are:

1. Model selection – allows users to select and apply one or more software reliability models to the failure data displayed in the work space.
2. Model definition – allows users to define combination models to supplement models already provided with the tool. User-defined models remain available during the current and subsequent sessions.
3. Model editing/model removal – allows users to change or remove descriptions of combinations that were previously created using the "Model definition" capability. Only user-defined combinations of models can be changed or removed.
4. Parameter estimation – allows users to select the method of parameter estimation that will be used. The choices are maximum likelihood (default) and least squares.

5. Predictions − allows users to specify an interval of time over which predictions about future reliability behavior will be made.

3.5: Program configuration

Two operations, "Add application" and "Remove application," are available from this manual. They allow users to add or remove the name of an application to the "External application" submenu from which external editors can be invoked.

3.6: Help system

The help system provides context-sensitive on-line assistance to users by allowing them to search for and read descriptions of the major CASRE functional areas.

3.7: Displays

The graphics display window is invoked when the "Open" button under main window's "File" menu operation is selected. The graphics display window plots the failure data displayed in the work space and the results of applying models to that data. A separate menu bar is associated with this window, allowing users to control the contents and appearance of the display or send the contents of the window to an output device (disk file or printer). The menu items on this menu bar are:

1. Plot − There are 5 plot operations ("Save plot", "Save plot as", "Draw from file", "Setup printer", " Print plot") that can be selected for plot manipulation and drawing.
2. Model result selection − Multiple sets of model results can be displayed in the graphics display window. This capability allows users to specify the models whose results should be plotted in that window.
3. Display type − Modeling results and evaluations of models can be displayed in a variety of ways, including: (1) Interfailure times; (2) Number of failures per test interval; (3) Test interval lengths; (4) Cumulative number of failures; and (5) Reliability function.

   In addition to estimating reliability functions, users may wish to determine the applicability of the model(s) being run to the current set of failure data. The following types of model evaluation results can be drawn in the graphics window: (1) Goodness-of-fit tests; (2) Relative accuracy ("prequential likelihood"); (3) Model bias ("u-plot"); (4) Model bias trend ("y-plot"); (5) Model bias scatter plot; (6) Model noise; and (7) Model rankings.
4. Settings − allows users to scale the x and y axes of the graphics display window and to shift its origin. Also allows users to redraw data and/or model results.
5. Table − allows users to view the modeling results in the graphics display window in a tabular form. This table shows the detailed data on which plotted model results, evaluations, and rankings are based.

4: CASRE on-screen appearances

Figures 2-7 show a series of screen dumps for the CASRE tool. The application of models to failure data is straightforward. Users are also given a considerable amount of choice in the models to be applied. This combination of simple operation and variety in the available models makes it easy for users to identify an appropriate model for a particular development effort or to investigate a family of models.

4.1: Failure data display

The initial data display screen is shown in Figure 2. After opening a failure history file, the contents of the file are displayed in tabular and graphical form. If the data represents times between successive failures, the tabular display has three fields − the error number, the time since the last error, and the severity of the error on a scale of 1 to 9. Otherwise, the tabular display has 5 fields − test interval number, number of errors observed in the test interval, test interval length, fraction of the program tested during the interval, and severity of the errors in the test interval. Note that there can be multiple records for a test interval if errors of more than one severity category are encountered during a test interval.

The large graphics window displays the same data as the tabular display. If the failure data set is interfailure times, the initial graphic display is also interfailure times. If, as in this example, the data set is test interval lengths, the initial graphic display is the total number of failures per test interval. The display type can be changed by selecting one of the items from the "Display" menu associated with the graphics window. For example, the display can be changed to show the cumulative number of errors.

4.2: Selecting failure data range

Users may choose only a portion of the data set to estimate the current reliability of the software. This is because testing methods may change during the testing effort, or different portions of the data set may represent failures in different portions of the software. To use only a subset of the selected data set, users choose the

"Select data range..." item of the main window's "Model" menu. Choosing this item causes the dialog box shown in Figure 3 to appear, in which users select the starting and ending points, as well as the range of points that will be used in making the initial parameter estimates. By default, the entire data set is used, and the first half of the data set is used to make the initial parameter estimates.

Once a data range has been selected, all future modeling operations will be applied to that range, until either the range is changed or the size of the data set is changed. Any change in the size of the data set will cause the data range to revert to its default condition.

4.3: Applying software reliability models

After opening a data file, selecting a data range, and doing any filtering operations, one or more software reliability models can be run on the data. By choosing the "Select models..." item in the "Model" menu, users display the dialog box shown in Figure 4. The "Available Models" list shows the models that can be run on the data. Note that the list includes individual models as well as model combinations that have been defined by the user (e.g. SLC_13). To select a model, users highlight the name of a model in the "Available Models" list and then click on the "Add" button. The name of the model is added to the "Models to Run" list and removed from the list of available models. To run the selected models, users click on the "OK" button. This causes a dialog box to appear in which the progress of each model is displayed. This dialog box also allows users to cancel the modeling operation if they decide that it is taking too much time.

Prior to selecting models, users can also select the maximum likelihood or least squares parameter estimation methods. They may also specify the number of failures (or test intervals) after the last observation in the selected data range for which predictions of future behavior should be made.

4.4: Display of model results

After running one or more models, the modeling results can be displayed in the graphic display window. This is done by selecting the single item in the graphic display window's "Result" menu. Selecting this item brings up the dialog box shown in Figure 5. The leftmost list in the dialog box lists those models for which valid predictions could not be made, usually because parameter estimates would not converge. The middle list in this dialog box identifies the models that have been run, while the rightmost list names those models whose results will be displayed.

The "Add" and "Remove" buttons in Figure 5 are used to add the names of models to and remove them from the list of results to be displayed. Clicking the "OK" button produces a display of model results superimposed on the raw data, as shown in Figure 6. In this case, SLC_13 (a user-defined combination) and Yamada S-Shaped model results, as well as the raw data, are displayed.

In addition to seeing model estimates and predictions, users may also view the results of several types of statistical analysis supported by CASRE to help determine a model's applicability to a set of failure data. For instance, Figure 7 shows a u-plot to illustrate the selected models' bias.

5: Conclusions

Toward this end, we have demonstrated a modeling tool, called CASRE, to provide a systematic, comprehensive, and user-friendly approach for software reliability estimation. CASRE provides extensive automated support for software reliability modeling and measurement with graphical user interface and a high-resolution display of model results in a tool that is easy to learn and to use. Moreover, it features a set of linear-combination models for accurate estimations of software reliability. Some preliminary applications of this tool by real-world project data have shown promising results.

Acknowledgement

The work described in this paper was partly performed at the Jet Propulsion Laboratory, California Institute of Technology, under a NASA contract. The implementation of CASRE is being supported by the Air Force Operational Test and Evaluation Center under Task Order RE-182.

References

1. M.R. Lyu and A. Nikora, ''CASRE – A Computer-Aided Software Reliability Estimation Tool,'' *CASE-92 Proceedings*, pp. 264-275, Montreal, Canada, July 1992.
2. G.E. Stark, ''A Survey of Software Reliability Measurement Tools,'' *ISSRE'92 Proceedings*, pp. 90-97, Raleigh, North Carolina, October 1992.
3. M.R. Lyu and A. Nikora, ''Using Software Reliability Models More Effectively,'' *IEEE Software*, pp. 43-52, July 1992.