SOFTWARE REQUIREMENTS DESCRIPTION (SRD) FOR THE TOTAL-SYSTEM PERFORMANCE ASSESSMENT CODE GRAPHICAL USER INTERFACE (VERSION 1.0β)

Prepared by

Sitakanta Mohanty
Christopher Walker

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

July 2001

APPROVED:

Michael ... for G. Wittmeyer — 7/30/01

Gordon W. Wittmeyer, Manager
Performance Assessment

Date

CONTENTS

1.0 BACKGROUND

The U.S. Nuclear Regulatory Commission (NRC) and the Center for Nuclear Regulatory Analyses (CNWRA) have been developing their performance assessment tools to review license application (LA) for a proposed repository at Yucca Mountain (YM), Nevada. If the U.S. Department of Energy (DOE) submits an LA, the NRC will conduct a risk-informed review. The objective of risk-informing the review is to emphasize those aspects of the repository system and DOE's analyses that are most important to the protection of public safety and the environment.

NRC will use PA tools (Total-system Performance Assessment (TPA) computer code (Mohanty et al., 2000) and sensitivity analysis methods (Mohanty et al., 1999; Jarzemba et al., 1999) for conducting risk-informed and performance-based LA reviews. This is achieved partly by analyzing quantitative information used in performance assessment including parameter values, parameter uncertainties, parameter correlation, and parameter assumptions.

The TPA is a technically complex integrated multidisciplinary computer code for estimating long-term repository system behavior and provides a structure for examining couplings between phenomena that might not be adequately evaluated, within the limits of a single technical discipline. Integration of multiple disciplines has resulted in a code that has more than 930 input parameters and numerous input files representing spatial and temporal data variation as well as data uncertainty. The current approach for modifying input parameters requires the user to edit a very large file, which is very tedious, time consuming, and possibly error-prone. Additionally, for those who are not intimately familiar with the way the TPA code utilizes the input parameters, understanding the structure of input data is a challenge.

The purpose of this project is to develop a graphical user interface (GUI)-based preprocessor, which will allow a user to prepare input data for the TPA code with relative ease and in a short time period compared to the current approach. The GUI will provide transparency through logical groupings of the input parameters that will allow better understanding of the data used in the TPA code in two ways. First, parameters may be grouped by a particular subarea, or a repository component, such as the waste package, or a process, such as thermal reflux. Second, other group associations will enable the user to see if the changes in one parameter will affect other parameters to maintain consistency. For example, if a new subarea is added the user will be prompted to add all relevant parameters pertaining to the new subarea. This will assist users that may not be familiar with all of the disciplines represented in the complete list of parameters and their associated uncertainties and correlations. Overall, the pre-processor will increase flexibility in data entry and user interaction in preparing the input file for the TPA code.

A well developed pre-processor could help enhance staff utilization of the code, reduce data input errors, and decrease learning time for preparing a data set for the TPA code. The preprocessor will also help track the rationale of the user for changing input parameters from the base case supplied with the code.

2.0 SOFTWARE FUNCTION

The following is a list of functions proposed for the GUI interface.

- The preprocessor will allow the user to review default input data used in the TPA code in an intuitive manner.
- The preprocessor will display uncertain data using the specified range or the distribution function.

- The software will check the data provided by the user to ensure data consistency. That is, the pre-processor will have several error checks that will supplement error checks already in the TPA code. For example, checks for sub-area UTM coordinates will be performed to ensure that the subarea is within the confines of the repository.

- The pre-processor will default to the basecase parameters. The user may modify the basecase file or a previously saved TPA input file until the user decides to create a new file. The revised parameter set can be saved at any time.

- The user can select a distribution type from a list of distribution functions.

- The user can change the parameter range by entering new data and distribution function type by selecting from a list on a special screen. This special screen will graphically display the data using the selected distribution function and the specified data range.

- The software will create a *tpa.inp* file using the new set of data elements. Data and time stamp will indicate that the file is newly created.

- Input integration will be emphasized. If a parameter that has dependent parameters is changed, the user will be prompted for changes to related parameters. For example, if a nuclide is added or removed from the basecase data set, then several other variables need to be added or removed. A "reset" option with limited functionality will enable the user to recover the original default valve.

- Addition of features will be prioritized to match the budget and staff/contractor availability. For instance the specification of subareas could be changed to allow the user to drag the lines with the mouse or the graphing of distribution types could be improved.

3.0 TECHNICAL BASIS: PHYSICAL AND MATHEMATICAL MODEL

3.1 Software Description

The object-oriented approach will classify the TPA data set (consisting of data elements) into a series of logical sections, each containing a number of variables. These sections are each logically linked to a module in the TPA code. The data elements in these sections will be grouped together logically. Each of these logical groupings will be considered a module in the preprocessor.

The GUI will use a hierarchical tree approach. In this approach, the global parameters and main modules will be specified first. The complexity of the data set will be revealed as the user goes to lower and lower levels of the hierarchy. As shown in Figure 1, the parameters in the ASHPLUMO module are grouped into a few broad categories such as volcanic event, air properties, wind data, fuel particles, ash characteristics, and so on. As the user goes to the lowest level of the branch and clicks on a leaf, the special GUI page (Figure 2) will be available for data entry. Figure 1 also shows two text boxes to display parameter description and rationale for the choice of data. The user can enter revised rationale or provide rationale for changing a parameter or distribution type (see Figure 2).

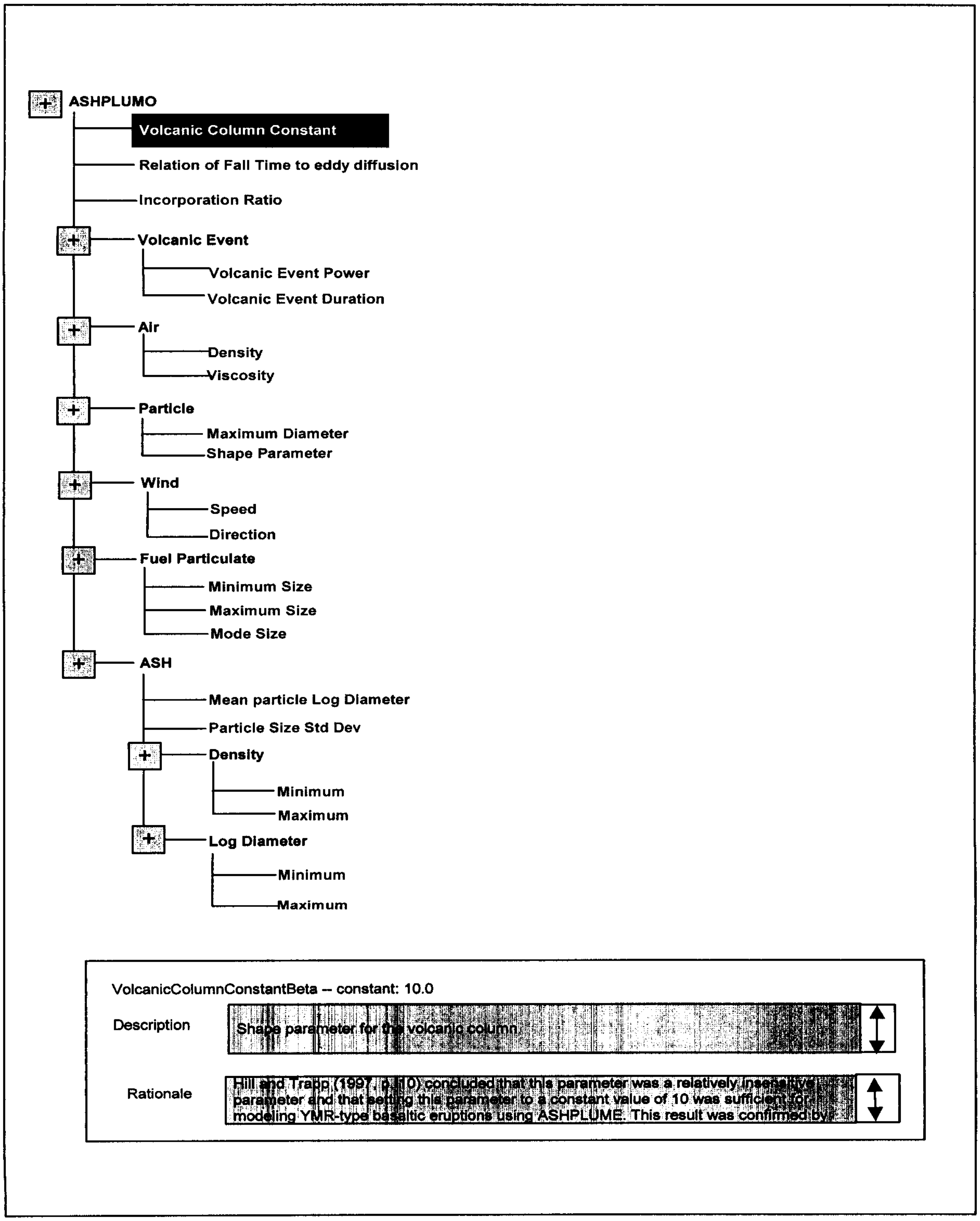

Figure 1. Proposed design for graphically displaying parameters in a Total-system Performance Assessment code module. Double clicking on a parameter name will bring up a screen that allows users to modify that variable.

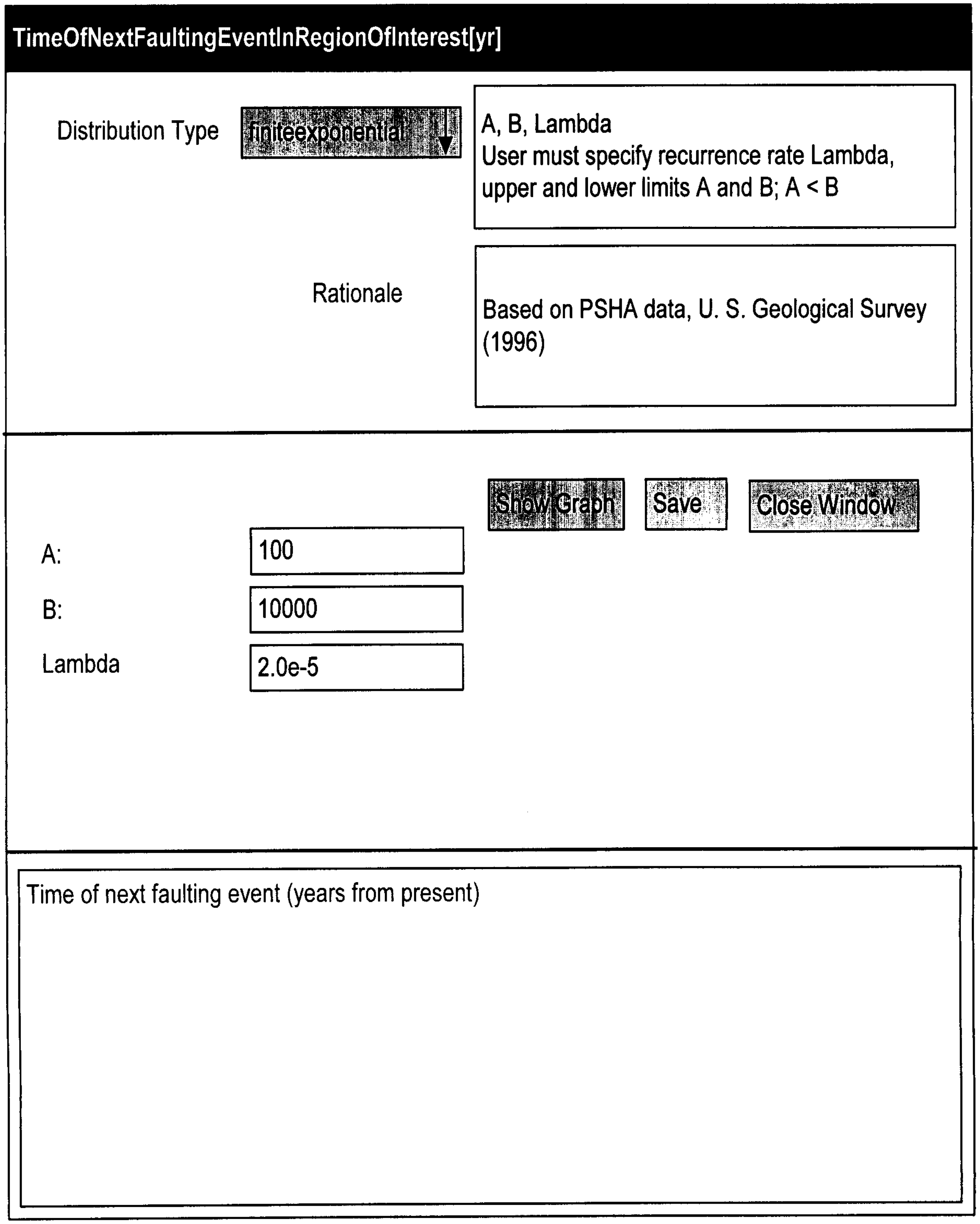

Figure 2. An example GUI page where TPA input parameters (distribution function, data range) can be revised and rationale for the revision can be documented.

The software will make use of any previously defined base-case *tpa.inp* files that exist on the user's computer. If the user has a previously created *tpa.inp* file that needs to be modified, the software will open that file, read in the data elements, display them to the screen, allow the user to modify the data elements, and write them back to the same file or a new file depending on the user's choice.

The software will group like-items together. For example, all distribution parameters for the EBS Release module can be grouped and displayed together. Additionally, data elements can be viewed in logical groupings. For example, all parameters for subarea 1 can be viewed together.

The software will provide the user the ability to add, remove, or modify subareas and radionuclide chains by prompting the user to modify the necessary parameters associated with addition of a subarea or radionuclide. Note that Java is platform-independent, implying that the software can run under Sun OS, Solaris, and/or windows operating systems.

Hardware and Software Requirements

This software will be written as a standalone application that will be independent of the installed operating system. Any machine that contains a Sun-Java runtime environment will be able to execute this application successfully.

Graphics Requirements

The preprocessor will be designed for a 17" VGA Color monitor with a minimum of 1024x768 resolution. The software will be able to run on different monitor sizes and with different resolutions but this will require the user to manually adjust screen sizes to view the screens properly. Monitors other than VGA or SVGA may not display the screens properly.

Graphing will be done utilizing the Java graphing objects. No specialized software or hardware is needed.

3.2 Mathematical Model

Only a limited mathematical model will be used in the preprocessor development. Mathematics will be primarily limited to equations presented in the appendix B and appendix D of the TPA 4.0 User's Guide (Mohanty et al., 2000). An example area where mathematical formulation will be required, is the graphical creation of the subarea coordinate system. The user will input the subarea data in the form of four Universal Transverse Mercator (UTM) coordinate pairs, which form the four vertices of the quadrilateral. The visual representations of these subareas will be displayed on a graphical plane and will be calculated using basic geometrical calculations as shown below.

- $SubArea_1$ Line would be written to the screen as DrawTheLine ((X_1, Y_1), (X_2, Y_2)).

- Consider the following set of (X, Y) coordinates that define the quadrilateral for Subarea 1 that would be written as follow:

  Subarea 1
  547514.88,4079310.61
  548069.20,4079136.50
  547847.30,4077816.20

547370.95,4077922.04
547514.88,4079310.61

- The coordinate pairs define the lines that bound the quadrilateral for Subarea 1. To draw the first line of this subarea that would extend from coordinate (547514.88,4079310.61) to coordinate (548069.2, 4079136.5), the code would execute the command

DrawTheLine((547514.88,4079310.61), (548069.20, 4079136.50)).

Mathematical formulation will also be used for displaying the distribution function selected by the user. On the special GUI screen, the user will select a probability distribution function from the predefined list, choose a set of related variables, and enter values. The probability density function will be computed using supplied mathematical formulae and displayed with the specified range and other related values.

4.0 COMPUTATIONAL APPROACH

None.

5.0 REFERENCES

Jarzemba, M.S., R.B. Codell, L. Deere, J.R. Firth, C. Lui, S. Mohanty, K. Poor, J Weldy, and V. Colten-Bradley. "NRC Sensitivity and Uncertainty Analyses for a Proposed HLW Repository at Yucca Mountain, Nevada, Using TPA 3.1. Volume II: Results and Conclusions." NUREG–1668. Volume 2. Washington, DC: U.S. Nuclear Regulatory Commission. 1999.

Mohanty, S. McCartin T, and Esh, D. "TPA Version 4.0 Code Module Descriptions and User's Guide." San Antonio, TX: Center for Nuclear Waste Regulatory Analyses. 2000.

Mohanty, S., R. Codell, R.W. Rice, J. Weldy, Y. Lu, R.M. Byrne, T.J. McCartin, M.S. Jarzemba, and G.W. Wittmeyer. "System-Level Repository Sensitivity Analyses Using TPA Version 3.2 Code." CNWRA 99-002. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses. 1999.

SOFTWARE REQUIREMENTS DESCRIPTION (SRD) FOR THE TOTAL-SYSTEM PERFORMANCE ASSESSMENT CODE GRAPHICAL USER INTERFACE (VERSION 1.0β)

Prepared by

Sitakanta Mohanty
Christopher Walker

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

July 2001

APPROVED:

Gordon Wittmeyer 7/10/2001

Gordon W. Wittmeyer, Manager **Date**
Performance Assessment

CONTENTS

1.0 BACKGROUND

The U.S. Nuclear Regulatory Commission (NRC) and the Center for Nuclear Regulatory Analyses (CNWRA) have been developing their performance assessment tools to review the license application (LA) for a proposed repository at Yucca Mountain (YM), Nevada. When the U.S. Department of Energy (DOE) submits an LA, the NRC will conduct a risk-informed review. The objective of risk-informing the review is to emphasize those aspects of the repository system and DOE's analyses that are most important to the protection of public safety and the environment.

NRC will use PA tools (Total-system Performance Assessment (TPA) computer code (Mohanty et al., 2000) and sensitivity analysis methods (Mohanty et al., 1999; Jarzemba et al., 1999) for conducting risk-informed and performance-based LA reviews. This is achieved partly by analyzing quantitative information used in performance assessment including parameter values, parameter uncertainties, parameter correlation, and parameter assumptions.

The TPA is a technically complex integrated multidisciplinary computer code for estimating long-term repository system behavior and provides a structure for examining couplings between phenomena that might not be adequately evaluated, within the limits of a single technical discipline. Integration of multiple disciplines has resulted in a code that has more than 930 input parameters and numerous input files representing spatial and temporal data variation as well as data uncertainty. The current approach for modifying input parameters requires the user to edit a very large file, which is very tedious, time consuming, and possibly error-prone. Additionally, for those who are not intimately familiar with the way the TPA code utilizes the input parameters, understanding the structure of input data is a challenge.

The purpose of this project is to develop a graphical user interface (GUI)-based preprocessor, which will allow a user to prepare input data for the TPA code with relative ease and in a short time period compared to the current approach. The GUI will provide transparency through logical groupings of the input parameters that will allow better understanding of the data used in the TPA code in two ways. First, parameters may be grouped by a particular subarea, or a repository component, such as the waste package, or a process, such as thermal reflux. Second, other group associations will enable the user to see if the changes in one parameter will affect other parameters to maintain consistency. For example, if a new subarea is added the user will be prompted to add all relevant parameters pertaining to the new subarea. This will assist users that may not be familiar with all of the disciplines represented in the complete list of parameters and their associated uncertainties and correlations. Overall, the pre-processor will increase flexibility in data entry and user interaction in preparing the input file for the TPA code.

A well developed pre-processor could help enhance staff utilization of the code, reduce data input errors, and decrease learning time for preparing a data set for the TPA code. The preprocessor will also help track the rationale of the user for changing input parameters from the base case supplied with the code.

2.0 SOFTWARE FUNCTION

The following is a list of functions proposed for the GUI interface.

- The preprocessor will allow the user to review default input data used in the TPA code in an intuitive manner.

- The preprocessor will display uncertain data using the specified range or the distribution function.

- The software will check the data provided by the user to ensure data consistency. That is, the pre-processor will have several error checks that will supplement error checks already in the TPA code. For example, checks for sub-area UTM coordinates will be performed to ensure that the subarea is within the confines of the repository.

- The pre-processor will default to the basecase parameters. The user may modify the basecase file or a previously saved TPA input file until the user decides to create a new file. The revised parameter set can be saved at any time.

- The user can select a distribution type from a list of distribution functions.

- The user can change the parameter range by entering new data and distribution function type by selecting from a list on a special screen. This special screen will graphically display the data using the selected distribution function and the specified data range.

- The software will create a *tpa.inp* file using the new set of data elements. Data and time stamp will indicate that the file is newly created.

- Input integration will be emphasized. If a parameter that has dependent parameters is changed, the user will be prompted for changes to related parameters. For example, if a nuclide is added or removed from the basecase data set, then several other variables need to be added or removed. A "reset" option with limited functionality will enable the user to recover the original default valve.

- Addition of features will be prioritized to match the budget and staff/contractor availability. For instance the specification of subareas could be changed to allow the user to drag the lines with the mouse or the graphing of distribution types could be improved.

3.0 TECHNICAL BASIS: PHYSICAL AND MATHEMATICAL MODEL

3.1 Software Description

The object-oriented approach will classify the TPA data set (consisting of data elements) into a series of logical sections, each containing a number of variables. These sections are each logically linked to a module in the TPA code. The data elements in these sections will be grouped together logically. Each of these logical groupings will be considered a module in the preprocessor.

The GUI will use a hierarchical tree approach. In this approach, the global parameters and main modules will be specified first. The complexity of the data set will be revealed as the user goes to lower and lower levels of the hierarchy. As shown in Figure 1, the parameters in the ASHPLUMO module are grouped into a few broad categories such as volcanic event, air properties, wind data, fuel particles, ash characteristics, and so on. As the user goes to the lowest level of the branch and clicks on a leaf, the special GUI page (Figure 2) will be available for data entry. Figure 1 also shows two text boxes to display parameter description and rationale for the choice of data. The user can enter revised rationale or provide rationale for changing a parameter or distribution type (see Figure 2).

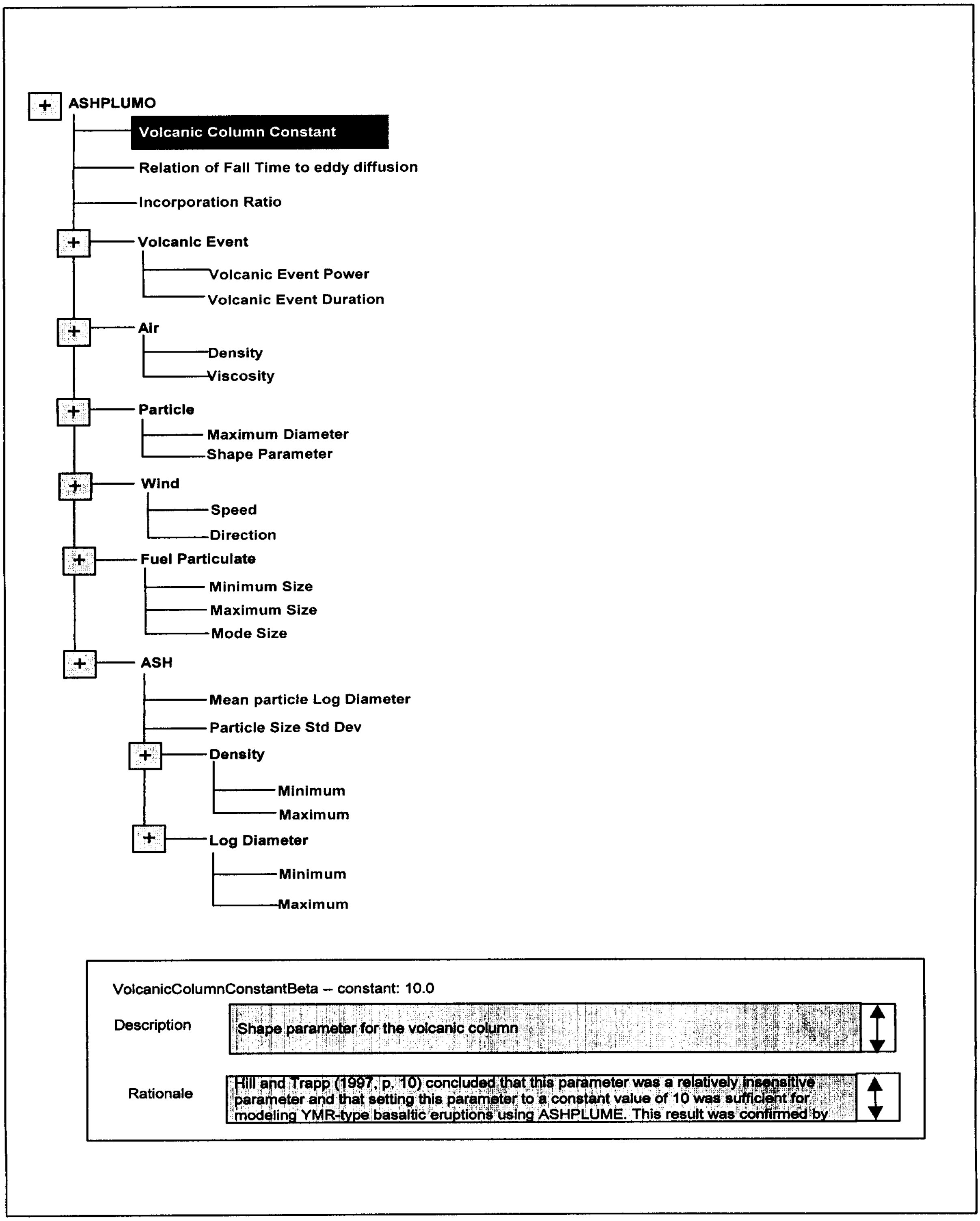

Figure 1. Proposed design for graphically displaying parameters in a Total-system Performance Assessment code module. Double clicking on a parameter name will bring up a screen that allows users to modify that variable.

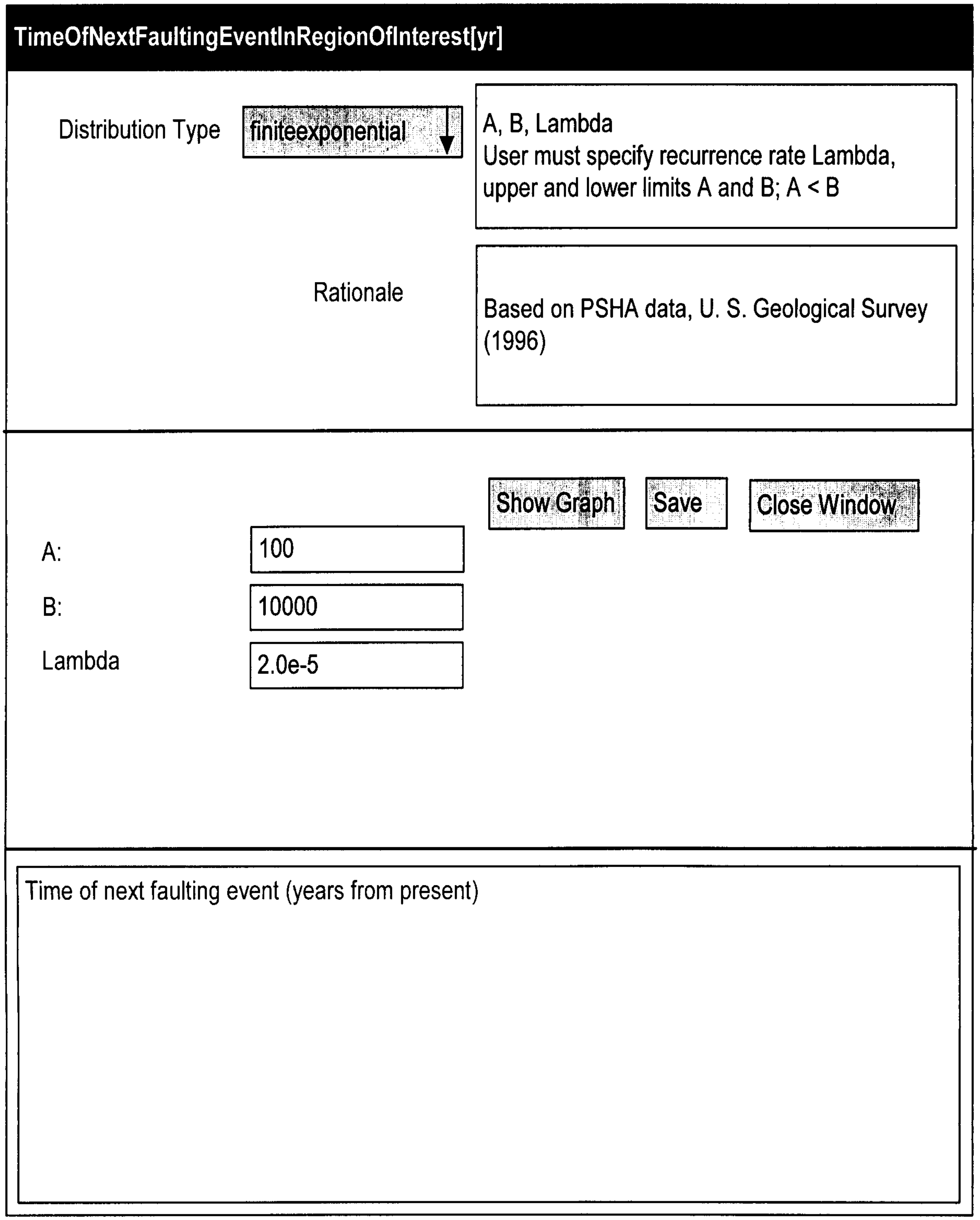

Figure 2. An example GUI page where TPA input parameters (distribution function, data range) can be revised and rationale for the revision can be documented.

The software will make use of any previously defined base-case *tpa.inp* files that exist on the user's computer. If the user has a previously created *tpa.inp* file that needs to be modified, the software will open that file, read in the data elements, display them to the screen, allow the user to modify the data elements, and write them back to the same file or a new file depending on the user's choice.

The software will group like-items together. For example, all distribution parameters for the EBS Release module can be grouped and displayed together. Additionally, data elements can be viewed in logical groupings. For example, all parameters for subarea 1 can be viewed together.

The software will provide the user the ability to add, remove, or modify subareas and radionuclide chains by prompting the user to modify the necessary parameters associated with addition of a subarea or radionuclide. Note that Java is platform-independent, implying that the software can run under Sun OS, Solaris, and/or windows operating systems.

Hardware and Software Requirements

This software will be written as a standalone application that will be independent of the installed operating system. Any machine that contains a Sun-Java runtime environment will be able to execute this application successfully.

Graphics Requirements

The preprocessor will be designed for a 17" VGA Color monitor with a minimum of 1024x768 resolution. The software will be able to run on different monitor sizes and with different resolutions but this will require the user to manually adjust screen sizes to view the screens properly. Monitors other than VGA or SVGA may not display the screens properly.

Graphing will be done utilizing the Java graphing objects. No specialized software or hardware is needed.

3.2 Mathematical Model

Only a limited mathematical model will be used in the preprocessor development. Mathematics will be primarily limited to equations presented in the appendix B and appendix D of the TPA 4.0 User's Guide (Mohanty et al., 2000). An example area where mathematical formulation will be required, is the graphical creation of the subarea coordinate system. The user will input the subarea data in the form of four Universal Transverse Mercator (UTM) coordinate pairs, which form the four vertices of the quadrilateral. The visual representations of these subareas will be displayed on a graphical plane and will be calculated using basic geometrical calculations as shown below.

- $SubArea_1$ Line would be written to the screen as DrawTheLine $((X_1, Y_1), (X_2, Y_2))$.
- Consider the following set of (X, Y) coordinates that define the quadrilateral for Subarea 1 that would be written as follow:

  Subarea 1
  547514.88,4079310.61
  548069.20,4079136.50
  547847.30,4077816.20

547370.95,4077922.04
547514.88,4079310.61

- The coordinate pairs define the lines that bound the quadrilateral for Subarea 1. To draw the first line of this subarea that would extend from coordinate (547514.88,4079310.61) to coordinate (548069.2, 4079136.5), the code would execute the command

DrawTheLine((547514.88,4079310.61), (548069.20, 4079136.50)).

Mathematical formulation will also be used for displaying the distribution function selected by the user. On the special GUI screen, the user will select a probability distribution function from the predefined list, choose a set of related variables, and enter values. The probability density function will be computed using supplied mathematical formulae and displayed with the specified range and other related values.

4.0 COMPUTATIONAL APPROACH

None.

5.0 REFERENCES

Jarzemba, M.S., R.B. Codell, L. Deere, J.R. Firth, C. Lui, S. Mohanty, K. Poor, J Weldy, and V. Colten-Bradley. "NRC Sensitivity and Uncertainty Analyses for a Proposed HLW Repository at Yucca Mountain, Nevada, Using TPA 3.1. Volume II: Results and Conclusions." NUREG–1668. Volume 2. Washington, DC: U.S. Nuclear Regulatory Commission. 1999.

Mohanty, S. McCartin T, and Esh, D. "TPA Version 4.0 Code Module Descriptions and User's Guide." San Antonio, TX: Center for Nuclear Waste Regulatory Analyses. 2000.

Mohanty, S., R. Codell, R.W. Rice, J. Weldy, Y. Lu, R.M. Byrne, T.J. McCartin, M.S. Jarzemba, and G.W. Wittmeyer. "System-Level Repository Sensitivity Analyses Using TPA Version 3.2 Code." CNWRA 99-002. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses. 1999.

SOFTWARE DEVELOPMENT PLAN (SDP) FOR THE TOTAL-SYSTEM PERFORMANCE ASSESSMENT CODE GRAPHICAL USER INTERFACE

Prepared by

Sitakanta Mohanty
Christopher Walker

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

July 12, 2001

Gordon Wittmeyer April 2, 2002

1.0 SCOPE

The scope of this project is the development of a graphical user interface (GUI)-based preprocessor, which will allow a user to prepare input data for the Total ~~Perfromance~~Performance Assessment (TPA) code with relative ease. The GUI-based preprocessor, ~~refered~~referred to as the TPA Editor will assist the user by providing logical groupings of the input parameters, which will allow better understanding of the data used in the TPA code. For example, parameters will be grouped according to subarea or ~~acceoring~~according to TPA module or according to Nuclide. Because the users are not necessarily versed in the subject matter associated with the input parameters along with their associated uncertainty and correlation with other parameters, the TPA Editor will prompt the user for additional parameters that may be associated with the entered parameter ~~set .~~set. For example, if a new subarea is added, the user will be prompted to add all relevant parameters pertaining to that new subarea.

The TPA Editor will be capable of maintaining all parameters that appear in the TPA input files for the version 4.0 TPA Code.

2.0 BASELINE ITEMS

2.1 SOFWARE REQUIREMENT DESCRIPTION (SRD)

The SRD, dated 3 JULY 2001, is the controlling document for this software development plan.

2.2 PROTOTYPE CODE

The prototype code baselined as version 0.1 will be the starting point for code development for the TPA Editor. This code is checked in to VSS in the TPA Editor project.

2.3 TPA.INP

This file contains all the parameters that will be used in the TPA Editor.

2.4 TPA VERSION 4.0 USERS GUIDE

The guide contains mathematical functions and lists of parameters as well as descriptions and rationale for the parameters. The information contained in the TPA Version 4.0 Users ~~Guide(~~Guide (Appendix A) will be utilized for the base case description and rationale for the code.

3.0 PROJECT MANAGEMENT

3.1 CONFIGURATION MANAGEMENT

Visual Source Safe (VSS) will be utilized for configuration management. Donna Jeffrey is monitoring the VSS server and has created a project for the TPA Editor. Chris Walker and Donna Jeffrey will be the only ones with access to checkout and modify the TPA Editor Code. Any other person will need permission from Sitikanta Mohanty to check out the code.

Software source code will need to be checked out from VSS at the beginning of each working day and then checked back in to VSS and the end of each working day. Each time that the code is checked back in to VSS a rational and description of the changes will be made for each source module that has been modified.

3.2 SCHEDULE AND WORK BREAKDOWN STRUCTURE

ACTIVITY	Task Name	Start Date	End Date	Estimated Task Hours
Coding				
	Scenario Selection	7/16	7/16	8
	Conceptual Selection	7/17	7/17	8
	Conceptual DataSets	7/18	7/18	8
	Opening and Saving Files	7/19	7/20	16
	Nuclide Modeling	7/21	7/22	16
	Generic Tree	7/23	7/24	16
	Generic DataSets	7/25	7/26	16
	Adding Panels	7/27	7/27	8
	Data Manipulation Functionality	7/28	8/16	40
	Correlation Parameters	8/17	8/18	16
	Graphing Distribution Functions	8/21	8/25	40
				192
TESTING				
	Conceptual Modeling	8/27	8/27	8
	Distribution Functions	8/28	8/28	8
	Keyboard	8/29	8/30	16
				32
MAINTENANCE				
	Fixing Bugs/Additional Functionality/Additional Testing	8/31	9/14	**80**

3.3 MILESTONES

The following milestones will be tracked:

- Conceptual Modeling Completed and Opening/Saving Files Completed 7/20
- Nuclide Modeling , Generic Structure Sets Completed and Panels added 7/27
- Data Manipulation screens completed 8/18
- All Coding completed 8/25
- Testing Completed 8/31

3.4 RISK MANAGEMENT

The Risks associated with this software development plan are minimal. Any change of the "Look and Feel" of the TPA Editor after 7/16/01 could delay the completion.

If the programming effort does not begin on 7/16/01 then the end date will be delayed accordingly.