BenderRBT Test Case Design Tool – Release 8.0

Cause-Effect Graphing Test Case Design Component

- **Neoning:** Neoning is a powerful tool for analyzing your graphs. It allows you to select a variation or a test and have its states highlighted on the graph. When you highlight a variation you can have it extrapolate the states, forwards and backwards – i.e. the states in the variation imply that other nodes must be at specific states. For infeasible variations when the states are extrapolated the conflicting states are shown in gray. In neoning mode you can also manually set the states of one or more nodes and extrapolate the states of other nodes.

- **UI Type:** In the Node definition property sheet you can now specify what type of variable the node represents. Defining UI types is totally flexible, allowing the user to specify whatever they want. You might use a generic description such as GUI or data base. You could also be more specific such Customer Data Base. This information is then included in the outputs generated by RBT. This feature improves the integration with playback tools such as Quality Center and data base utilities such as Grid Tools Data Maker.

- **Global Language Support:** The user may enter the text for Node Names, Node Descriptions, Title, Test ID's, and Notes in any language they choose.

- **Improved Sequence Control for Test Scripts:** The sequence of the nodes in the test scripts as produced by RBT was not always in the sequence the tester wanted them in. A number of factors in the test design engine drove the sequencing. You can now control the sequence of the nodes and that will be reflected in all of the RBT outputs.

- **Limitations Removed on Node Names and Node Descriptions:** The input to the RBT test design engine had always been the CEG file. This was a Prolog syntax text file. Being a simple text file resulted in numerous limitations on what characters could be in the input. You could also not use terms that RBT used as keywords such as Nodes, Constraints, etc., as node names. In the new release the input to the engine is now via XML. Therefore we were able to eliminate all limitations on what is in a node name and node description. For practicality reasons we still limit Node names to 32 characters.

- **View/Edit Constraints:** In prior releases adding or editing a Constraint could be challenging for very complex graphs. You could get lost in all the line connections. In Release 8 you can now display just the Constraint you are working on while hiding all of the others. You can also display just a subset of Constraints. When you finish editing the Constraint you can then return to the full display mode.

- **Queue Manager:** In earlier releases you could create and run queues of CEG files from the engine component. In Release 8 you can build and run queues of RBT files from the front end drawing component. The ability to create and run queues is useful when you have a set of graphs that take a while to run. You can set up a queue and then run them over lunch or overnight.

- **Scrub:** On occasion we have had customers who needed us to review their graphs but were limited in what they could show us due to confidentiality or security reasons. The scrub feature translates all of the node names into the handles RBT uses internally for processing. These are just numbers. The node descriptions are eliminated. The new graph has the same name as the original one with the term "_Scrubbed" appended to it. Therefore the meaning of the nodes is hidden while the structure of the graph remains unchanged.

- **Export to Quality Center:** The export to HP-Mercury's test management tool has been updated to cover the latest release of QC.

- **Export to TOSCA:** We have added an export of the tests to Tricentis's TOSCA.

- **TestIF Support:** RBT is one of the first test tools to support the OMG's Test Interchange Facility format. This is the mechanism by which all test tools will exchange information.

- **Consolidated Data Base:** Release 7 and earlier had multiple files – the RBT, CEG, D_B, and CSV files. All of this has now been consolidated into the one XML based RBT file. This makes managing the RBT data much simpler. The consolidated file is also smaller.

- **Revise Descriptions is Done Automatically:** In prior releases if you change the True/False node description or changed a Node's name you had to either rerun the model or at least run the Revise Descriptions to update the RBT reports. With the consolidated data base any change to the Node names or Node descriptions is automatically reflected in all of the reports as soon as you Save the changes.

- **Improved Integration of the Drawing Component with the Engine:** In prior releases when you selected a Report, Option, or Utility from the front end you just went to the CEG file in the back end engine. You then had to select what you wanted from the back end's menus. When you make a selection now you are taken directly to the Report, etc., you requested.

- **Improved Test File Editor:** RBT allows you to save a set of tests it created or define a set that already exists. This is used to protect the investment made in implementing the tests from previous releases of your applications. Creating or editing these tests is now a lot easier. It is done using the format of the Test Definition Matrix. You edit or add tests to the Old Tests file by just setting the causes to true or false. RBT extrapolates the states of the effects as you go.

- **Run New and Run Old / Run Both Data All Available for Review:** In prior releases a Run Old or Run New replaced the data created by a Run New. In RBT 8 both sets of data are available to review. Therefore you can easily see what coverage existing tests have and compare that to what RBT would come up with if it had designed all of the tests. You do not have to rerun the model to access this information.

- **Flexible Test Naming:** RBT now allows you to set a default for test case names for a graph. Previously tests were just named Test 1, Test 2, etc. You can have different user defined test name defaults for new tests versus those in a Test File. You can even modify the name of an individual test.

- **Next Generation RBT Engine:** To accomplish many of the above features the entire RBT engine was re-architected. We also rewrote the engine to better position RBT for integration with Requirements Management tools in the future – e.g. the ability to automatically generate the graph from the requirements. The new engine is usually slightly faster than the old one.

- **File Conversion:** You can bring an old RBT file into Release 8. It will then convert the old Access data base structure into the new XML data base. The old file is saved off as an MDB file. You can also run the bulk conversion utility which will convert all of your RBT, CEG, D_B files into the new XML based RBT file.

- **New Cause-Effect Graphing Documentation:** The Cause-Effect Graphing User Guide and User Tutorial have been totally rewritten to reflect all of the new functionality and the Cause-Effect Graphing Process Tutorial has been updated.

- **New Help Support:** The embedded Help has been upgraded to reflect all of the new functionality. The Help feature is now also available in the front end drawing component.

- **Check Graph:** As you are building the graph you can check to see if there are any basic issues before compiling it. The Check Graph feature will identify such things as orphaned nodes and Constraints that have been improperly set up.

- **Check Overlap**: When you move objects around on the drawing grid you might accidently place one on top of another. This is especially true when you copy/paste objects from another model. The Check Overlap feature identifies where this has happened and automatically corrects it.

- **Scrollable Reports:** In Release 7 there were a number of reports that were not scrollable. All reports are now scrollable (obviously long overdue).

Quick Design Component

- **Export to Quality Center** - An export to HP-Mercury's test management tool has been updated to cover the latest release of QC.

- **Flexible Test Naming:** QD now allows you to set a default for test case names for a graph. Previously tests were just named Test 1, Test 2, etc. You can have different user defined test name defaults for new tests versus old tests. You can even modify the name of an individual test.