

Presupposition Based Exploratory Testing

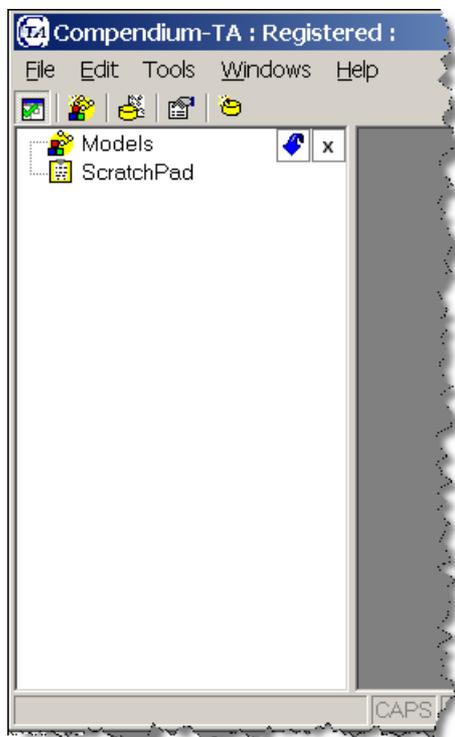
– a worked example

This is a draft. I should tidy the writing up, I should draw more conclusions, I might expand the analysis and show more paths that could be taken through the software belief statements (Requirements). And one day I might, but for now...

Exploratory testing involves building a model of (learning) the system as we use and test it.

This short example documents the use of NLP presupposition based techniques during exploratory testing.

A very simple area of a real system will be documented.



I know quite a bit about this system and the supposed functionality because I wrote it. But that doesn't invalidate this testing in fact it makes it quite interesting to see what a presupposition driven approach to testing leads me to do.

This is a side panel for a system and it is designed to show all models loaded into a system and various bits of information about them. The ScratchPad is a temporary storage area for model elements. I know that I can create folders under the scratchPad and that the ScratchPad can hold model elements from any of the loaded models.

We can delete some items from the sidebar by pressing delete, and move some around by using the mouse, e.g. when an item is in the scratchpad we can move it to a folder under the scratchpad etc.

Items in the tree can be double clicked to open for editing in the main system.

NLP analysis

UQ: *All* models are displayed under the models tab

MO: Delete key deletes *some* items from the tree

- Exploration reveals that structural branches can not be deleted ‘Models’, ‘ScratchPad’
- Windows, Shortcuts and folders can be deleted
- When the main window for the model is deleted the model closes
- ‘Model’ branches can not be deleted – should we be able to close the model by deleting it from the tree? (**raise issue**) (**action taken: close model on delete key of a Model branch**)

Presuppositions:

- Multiple models can be active in the system.
 - I’m focussing on the display function so I’m going to ignore the ‘What if they can’t be?’ path
 - New models are displayed in the side panel
 - Loaded models are displayed in the side panel
 - When a loaded model is closed it is removed from the side panel
 - Can the side panel trigger removal
 - Can’t do anything to the models branch when no models are loaded
- Top scratch pad is independent of the models
 - Can create folders and move folders when no models are loaded
 - Noted: Created folders all have name ‘NewFolder’
 - (P) multiple folders of same name is allowed
 - Rules for All Folders apply to folders of same name as well as named folders
 - (P) folders can be renamed -> Y, space bar, or click when selected
 - (P) names are valid (**raise issue**)
 - (P) a valid name is ascii only
 - (P) a valid name has length limits
 - >0 chars
 - <? chars
 - All folders can be moved around
 - UQ: *ALL*?
 - Move folder from scratchpad to models?
 - Parent -> child? Assume N
 - Sibling -> child? Assume Y
 - Parent.parent -> child? Assume N
 - Parent.sibling -> child? Assume Y
 - Parent.parent.sibling -> child? Assume Y
 - Child -> parent? Assume N
 - (P) items from any model can be added here
- Model items can be displayed in the top scratchPad
 - (P) Model has to be active to add model items to scratchpad
 - (P) model items in scratchpad are independent of the model so the model does not have to be loaded to use them

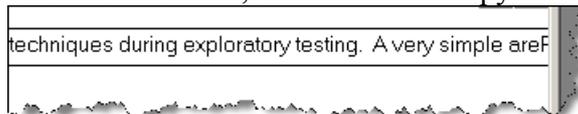
- (P) displayed element in scratchpad exists
 - what if it doesn't? (**raise issue**)
- (P) Displayed Elements in scratchpad can be dragged to models
- (P) Double clicking an element in the scratchpad opens it for editing
 - (Q) even if the model is not loaded? (**raise issue**)

Valid Names Explored



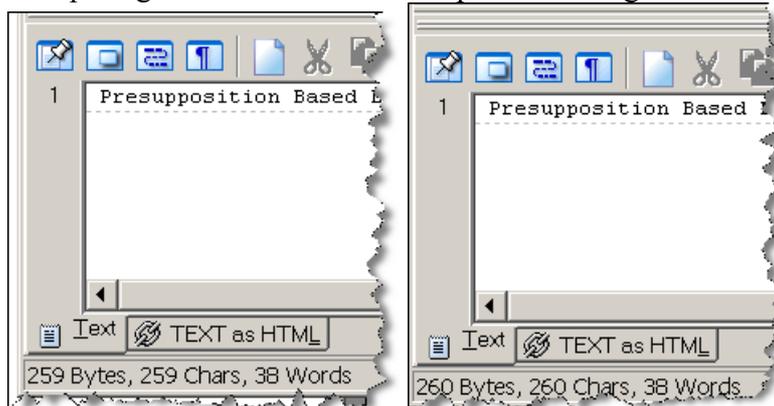
We can add very large names, cannot see the end of the edit, don't actually see the end to edit it. Looks like we can add another char but hard to tell.

Go into edit mode, select text and copy to clipboard.



Paste this to the end of the field and it looks like it has just added it based on the 'P' that we can just see. So (still in field edit mode) select and copy info to the clipboard manager.

Comparing the two states in the clipboard manager we see that:



The edit field can hold 260 chars but only 259 are saved into the system. So 259 is our upper limit.

We cannot add NULL names, but we can add names with only spaces

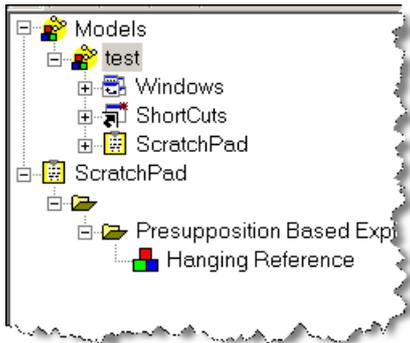


These are subject to the same 259 char limit.

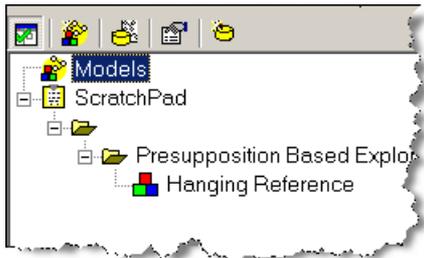
Raise issue (action taken: prevent blanks, trim spaces, restrict length to 50 chars)

Model has to be added to display elements at top level

- (P) ScratchPad items are independent of the model so the model does not have to be loaded to use them



Close model and reference is left.



Is this a problem? Don't know yet.

Reload model and try to use them

Can't add them back to the model. They are not independent.

Double clicking a model element when the model is not loaded has no effect.

(raise issue) (action taken: remove scratchpad items for closed models)

(P) displayed element in scratchpad exists

Creating a shortcut to a model element and then deleting the model element from the model, then double clicking on the scratchpad item opens a blank form because the model element is deleted.

Refreshing the display does not remove the deleted item from the display

(raise issue) (action taken: check for existence before opening and remove if not exists)

Summary

This was a very quick test session (< 15 mins). It actually took far longer to write this document (despite its obvious crudeness and lack of polish) than to test, identify and fix the defects.

I could have found the defects in other ways. I could have used boundary value analysis on the input fields. I could have looked through 'How to break software' and identified attacks. I could have used my experience.

I wanted to see how much information the system would give me as to how to proceed. And the system gives lots of information, as with many things, if we learn to observe, and then learn effective ways of questioning and modelling what we observe, we can achieve results quickly.

Based on that observed information I can apply my experience as a tester and developer. Much as a therapist, questions and observes and builds a model, and then applies their techniques and experiences to that model.

I can also take it at face value, take it literally and ask questions based on the Meta Model to guide my testing. This approach can be taught. Easily. This approach can be practised and remembered easily.

I do find it difficult to remember other peoples' models:

- James Whitacker's attacks
- James Bach's Heuristics
- Jerry Weinberg's lists of observations

I find it easier to have tools that lead me to a test based on first principles, and apply my experience too that. Then I can build my own models and lists. But I do cheat and have lists of other peoples' models nearby to refer to them. Is that cheating?

The Meta Model is a very simple way of exploring what we do.

Tools used for testing and seen in this document:

Compendium-TA the system under test

SnagIt the screen capture tool

ClipMate the clipboard manager