JPass Model Testing

Verificação e Validação de Software 2020–2021

Manuel Tomás 51054
Tiago Varela 51017

2021/03/22

Contents

Selected Use Cases

We initially experimented using the GUI and understanding its behaviour. By pretending to be normal users and discovering all its abilities we could then derive possible Use Cases for a normal user. In particular, we thought of scenarios that would exercise the GUI's different screens and states. Once we knew what functions the tool possesses and what diffrent screens existed we chose the following three use cases:

Use Case 1

Description. The user is looking to open a file to see their entries. Therefore, the user opens a stored JPass file, inputing the corresponding file password that is necessary to open it.
Selection explanation. Use case 1 was selected because it depicts a branch of the model tree that covers three state changes, and for representing a likely task a user may perform.

Use Case 2

Description. The user is looking to store their credentials. Therefore, the user creates a new entry, fills it with their data and uses the tool's "Generate Password" function in the process.
Selection explanation. Use case 2 was selected for the same reason as use case 1. In particular we have found that the JPass tool performs a lot of actions that change the content but relatively few which transition state. It is possible that some of these actions do in fact represent a transition to a new state, rather than the return to an old one. For example, when you open a file we have modelled it such that it returns to the Idle state, but it's possible that this is a whole new state with a similar UI. (See [State Machine] for details on the Idle > OpenFileMenu > Authenticating > Idle transitions we mention here.)

Use Case 3

Description. The user is looking to edit one of their entries. Therefore, they use the search function to search the entry in question by name. Afterward, they edit the entry and finalise their changes.
Selection explanation. Use case 3 was selected for the same reason as the two previous cases but with a difference: Fot this case, we join two seperate tree branches into a single continuous path. (See State Machine for details on the Idle > FindEntryMenu > Idle > EditEntryMenu > Idle transitions we mention here.)

QF-Tool Feedback

The tool offers a level of depth and complexity much beyond what we required to create the simple test cases we needed. Nevertheless, it remains very much possible to interact and make use of the tool for simple tasks and tests. Rather than just depth,

convinience is also an important facet of any tool being developed for regular use, and the convinience of recording the actions directly through the GUI of a Java program is unmatched, meaning the the tool is excellent even for usage by beginners.
Speaking of which, the available tutorials for beginners are extremely helpful. The existence of videos with subtitles ensures the maximum accessibility for any user.
Our experiences with the tool have been wholly positive, though it should be noted we have only used it to execute basic tasks and that all our feedback comes from the perspective of beginners with little experience.
The only issue we had was with understanding how to ensure tests could be run from varying machines. Even if a folder with all the necessary files and locations exists, we never found out how to indicate locations relative to a directory.

Use Case 1 - Open File

State Machine

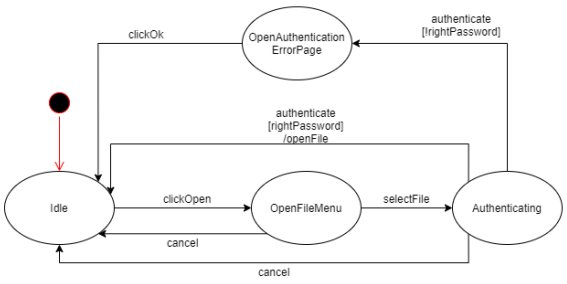

Figure 1: UC1 State Machine

Use Case 1 requires a few states in order to model it.

- Idle - As in all other models, this represents the main JPass screen where the open file's entries are displayed.
- OpenFileMenu - The menu where the user selects a file to open.
- Authenticating - A box where the user must input the file's password.
- OpenAuthenticationErrorPage - A warning screen that appears when an incorrect password is input.

Furthermore, the following transitions, which represent clickable UI elements in the JPass GUI, exist:

- clickOpen - Clicks a "Open File" button.
- selectFile - Selects a file in the menu (double click or ok).
- authenticate - Inputs a password and selects ok.
- clickOk - Click ok on the warning screen.
- cancel - Click on the X or cancel button.

Transition Tree

The following branches can be extracted.

- Idle > OpenFileMenu > Idle
  Opens the file menu but cancels.

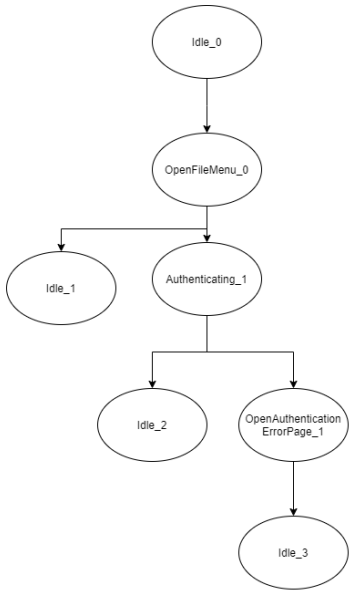

Figure 2: UC1 Transition Tree

- Idle > OpenFileMenu > Authenticating > Idle
  Opens the file and authenticates correctly.
- Idle > OpenFileMenu > Authenticating > OpenAuthenticationErrorPage > Idle
  Opens the file and authenticates incorrectly. Then closes the error page.

Transition Table

	clickOpen	selectFile	authenticate[rightPassword]	authenticate[!rightPassword]	clickOk	cancel
Idle	OpenFileMenu	x	x	x	x	x
OpenFileMenu	x	Authenticating	x	x	x	Idle
Authenticating	x	x	Idle	OpenAuthenticationErrorPage	x	Idle
OpenAuthenticationErrorPage	x	x	x	x	Idle	x

QF-Tool Test Cases

SuccessfulAuthentication	
What is being done?	1. Launches JPass 2. Clicks the Open File Button 3. Selects a .jpass file 4. Inputs the given file's password 5. Checks that Idle has the Entry from the opened file 6. Quits JPass
What is being tested?	Runs through the scenario where a user successfully opens a JPass file.
Result	Pass

FailedAuthentication	
What is being done?	1. Launches JPass 2. Clicks the Open File Button 3. Selects a .jpass file 4. Inputs an incorrect password for the given file 5. Confirms the incorrect password warning 6. Checks that Idle has no entries loaded to verify that the file was not loaded 7. Quits JPass
What is being tested?	Runs through the scenario where a user fails to open a JPass file due to an incorrect password.
Result	Pass

CancelAuthentication	
What is being done?	1. Launches JPass 2. Clicks the Open File Button 3. Selects a .jpass file 4. Cancels the password input process 5. Checks that Idle has no entries loaded to verify that the file was not loaded 6. Quits JPass
What is being tested?	Runs through the scenario where a user cancels out of opening a JPass file.
Result	Pass

FullUseCase	
What is being done?	1. Launches JPass 2. Clicks the Open File Button 3. Selects a .jpass file 4. Inputs an incorrect password for the given file 5. Confirms the incorrect password warning 6. Checks that Idle has no entries loaded to verify that the file was not loaded 7. Clicks the Open File Button 8. Selects a .jpass file 9. Inputs the given file's password 10. Checks that Idle has the Entry from the opened file 11. Quits JPass
What is being tested?	A likely user scenario as described by the Use Case: Mixes the FailedAuthentication scenario with the SuccessfulAuthentication one. The user fails the password input once and then inputs the correct one.
Result	Pass

Use Case 2 - Create Entry

State Machine

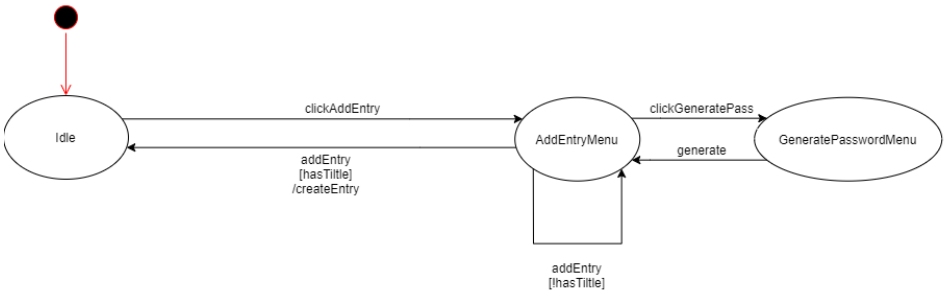

Figure 3: UC2 State Machine

Use Case 2 requires a few states in order to model it.

- Idle - As in all other models, this represents the main JPass screen where the open file's entries are displayed.
- AddEntryMenu - The menu where the user inputs all the information to create a new entry.
- GeneratePasswordMenu - The menu with the options about the password to be generated and the option to generate it.

Furthermore, the following transitions, which represent clickable UI elements in the JPass GUI, exist:

- clickAddEntry - Click the "Add Entry" button.
- clickGeneratePass - Click the "Generate" button.
- generate - Click the "Accept" button in the GeneratePasswordMenu.
- addEntry - Click on the ok button in the AddEntryMenu.

Transition Tree

The following branches can be extracted.

- Idle > AddEntryMenu > Idle
  Opens the add entry menu but cancels. Or inputs the entry data and adds it.
- Idle > AddEntryMenu > AddEntryMenu
  Opens the add entry menu and inputs the entry data except a title and adds the entry.

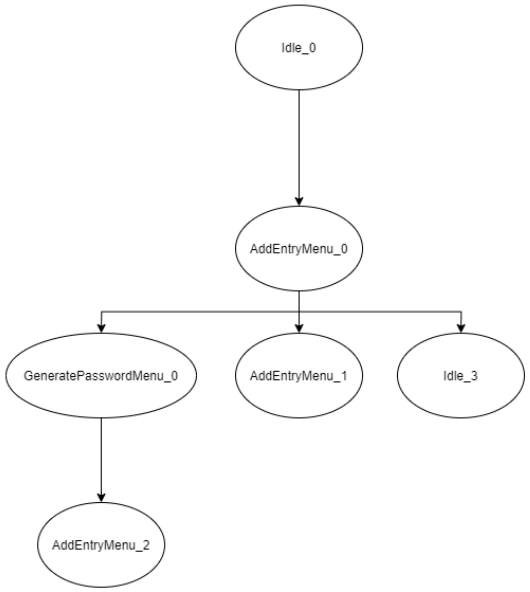

Figure 4: UC2 Transition Tree

- Idle > AddEntryMenu > GeneratePassword > AddEntryMenu
  Opens the add entry menu and inputs the entry data except a title. Then opens the generate password menu and generates it.

Transition Table

	clickAddEntry	clickGeneratePass	generate	addEntry[hasTitle]	addEntry[!hasTitle]
Idle	AddEntryMenu	x	x	x	x
AddEntryMenu	x	GeneratePasswordMenu	x	Idle	AddEntryMenu
GeneratePasswordMenu	x	x	AddEntryMenu	x	x

QF-Tool Test Cases

GeneratePassword	
What is being done?	1. Launches JPass 2. Clicks the Add Entry Button 3. Fills the entry with information except for the password 4. Clicks the generate password button, generates a password and exits the generate password menu 5. Checks if password was generated 6. Exits add new entry menu 7. Quits JPass
What is being tested?	Runs through the scenario where a user successfully generates a password for their entry.
Result	Pass

NoTitle	
What is being done?	1. Launches JPass 2. Clicks the Add Entry Button 3. Fills the entry with information except for the title 4. Confirms entry creation 5. Confirms the input title warning 6. Exits add new entry menu 7. Quits JPass
What is being tested?	Runs through the scenario where a user does not input a title for their entry.
Result	Pass

CreateEntryNoPasswordGeneration	
What is being done?	1. Launches JPass 2. Clicks the Add Entry Button 3. Fills the entry with information including a password 4. Confirms add new entry menu 5. Checks the given entry was created there 6. Quits JPass 7. Confirms nothing needs to be saved.
What is being tested?	Runs through the scenario where a user successfully creates a new entry.
Result	Pass

FullUseCase	
What is being done?	1. Launches JPass 2. Clicks the Add Entry Button 3. Fills the entry with information except for the password 4. Clicks the generate password button, generates a password and exits the generate password menu 5. Checks if password was generated 6. Confirms add new entry menu 7. Checks the given entry was created there 8. Quits JPass 9. Confirms nothing needs to be saved.
What is being tested?	Runs through the likely user scenario where a user generates a password for their entry and creates it.
Result	Pass

Use Case 3 - Find and Edit Entry

State Machine

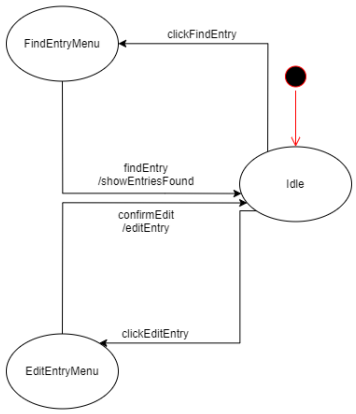

Figure 5: UC3 State Machine

Use Case 3 requires a few states in order to model it.

- Idle - As in all other models, this represents the main JPass screen where the open file's entries are displayed.
- FindEntryMenu - The menu where the user inputs the string to be found among the entries in the system.
- EditEntryMenu - The menu where the user can change any information contained in the entry.

Furthermore, the following transitions, which represent clickable UI elements in the JPass GUI, exist:

- clickFindEntry - Click the "Find Entry" button.
- findEntry - Insert the string to be found among the entries in the system.
- clickEditEntry - Click the "Edit Entry" button.
- confirmEdit - Click on the ok button in the EditEntryMenu to confirm the changes.

Transition Tree

The following branches can be extracted.

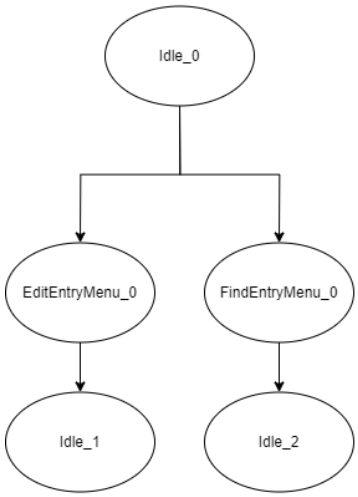

Figure 6: UC3 Transition Tree

- Idle > FindEntryMenu > Idle
  Opens the add find entry menu and inserts the string to be found. In the idle state the correspondent entries are shown.
- Idle > EditEntryMenu > Idle
  Opens the edit entry menu and inputs the entry data to be setted and confirms it, coming back to idle.

Transition Table

	clickFindEntry	findEntry	clickEditEntry	confirmEdit
Idle	FindEntryMenu	x	EditEntryMenu	x
FindEntryMenu	x	Idle	x	x
EditEntryMenu	x	x	x	Idle

QF-Tool Test Cases

FindEntry	
What is being done?	1. Launches JPass 2. Clicks the Open File button 3. Selects the file to open [1] 4. Clicks the open button 5. Inserts the password 6. Clicks the ok button 7. Clicks the Edit Menu button 8. Clicks the find entry button 9. Inserts a name in the find box 10. Checks if the entry at top is the right one 11. Quits JPass
What is being tested?	Runs through the scenario where a user successfully finds an entry.
Result	Pass

NoTitle	
What is being done?	1. Launches JPass 2. Clicks the Open File button 3. Selects the file to open [1] 4. Clicks the open button 5. Inserts the password 6. Clicks the ok button 7. Clicks the first entry of the list 8. Clicks the edit button 9. Inserts a username in the username box 10. Clicks the ok button 11. Clicks the first entry of the list 12. Clicks the edit button 13. Checks if the value in the username box is the inserted one 14. Clicks the close button 15. Quits JPass 16. Clicks in the no button of the "Save Changes Before Closing" window
What is being tested?	Runs through the scenario where a user edits an entry.
Result	Pass

[1] Note that the file opened was created beforehand with more than one entry

CreateEntryNoPasswordGeneration	
What is being done?	1. Launches JPass 2. Clicks the Open File button 3. Selects the file to open [1] 4. Clicks the open button 5. Inserts the password 6. Clicks the ok button 7. Clicks the Edit Menu button 8. Clicks the find entry button 9. Inserts a name in the find box 10. Clicks the first entry of the list 11. Clicks the edit button 12. Inserts a username in the username box 13. Clicks the ok button 14. Clicks the first entry of the list 15. Clicks the edit button 16. Checks if the value in the username box is the inserted one 17. Clicks the close button 18. Quits JPass 19. Clicks in the no button of the "Save Changes Before Closing" window
What is being tested?	Runs through the scenario where a user successfully finds and edits an entry.
Result	Pass

Sneak Paths

Sneak paths correspond to paths which the system is unprepared to deal with and is not meant to do so.
In the transition tables, all such paths are marked with an x. These paths should not exist because no action exists that should transition from the current state to the new state.
In order to attempt to find these sneak paths we closely studied the UI in each state and experimented with the various buttons. Most buttons are disabled when the corresponding transition should not be possible, and therefore we found no viable sneak paths during testing. The only available transitions from all states we tested were those that are depicted in the model and transition tree.