

FUDA EMPIRICAL EVALUATION

Development Experience Questionnaire (Used Documentation)

NOTE: This is a fillable PDF file. If you are not using a PDF writer, please make sure to print this file to a PS or PDF file to not lose the information.

Name: S1 Date: Aug. 11, 2008

Concept: JFace – Context Menu JFace – Content Assist
 Eclipse – Navigate Eclipse – Table Viewer

Q.1: Were you able to implement the concept successfully? Yes No

Q.2: How much time did you spend on the concept's implementation? 25 minutes

Q.3: If not successful to implement the concept, what was the main reason in your opinion?

- Lack of experience.
- Not useful documentation.
- Not useful sample applications.
- Complexity of the concept.
- Other. Please specify: _____

Q.4: Did you refer to the example applications' source code to implement the concept?

- No. None of them. Yes. One of them. Please Yes. Both of them.
specify: KTreeMap

Q.4.1: If yes, for what program statements and what kind of information?

Referred to KTreeView

- Looked at the 'drillDownAdapter' field, then traced its use using the Java Editor markers (on left hand side).
- Added "drillDownAdapter = new DrillDownAdapter(viewer);" statement to sample application.
- Added "drillDownAdapter.addNavigationActions(manager);"
- Traced the manager parameter from the method using the Call Hierarchy View to the statements: "IActionBars bars = getViewSite().getActionBars()" and "bars.getToolBarManager()".

Q.5: How many documentations did you use in this experiment?

None. Only one. Two.
Please specify: Java Doc

Q.6: For each documentation, please specify which parts of the documentation did you read?

Doc1: None Only Relevant A Big Portion All of that Parts
Doc2: None Only Relevant A Big Portion All of that Parts

Q.7: Were you able to find all the required information for Yes No implementing the concept in the provided documentation?

Q.7.1: If not, what kinds of information were missing in the provided documentation?

In answering this question, I referred to documentation as the JavaDoc and Tutorial.

The JavaDoc provided some usage information in the class header. However, this information was much too coarse-grained for me to construct any useful implementation. I could not find any useful information relating to the Nagivate concept in the Tutorial.

As a result, I needed to use the sample applications to find the implementation details.

Q.8: Were you able to easily access the desired information in the Yes No provided documentation?

Q.8.1: If not, what were the difficulties?

The Tutorial has too much information on various other concepts that were not of interesting.

The JavaDoc provided implementation detail that was either far too abstract (in the class header) or far too detailed (at the individual API calls). It was hard to see how to use these API calls in implementing the concept in my application.

Q.9: In your opinion, was the documentation concise enough? Yes No

Q.10: Overall, in the range of 1-5, how do you rank the provided documentation in terms of usefulness to implement the concept?

1 = Not Useful 2 3 4 5 = Excellent

Q.11: Do you have any additional comments on this experiment?

- The Eclipse Call Hierarchy View was useful in identifying variable usage.
- The provided JavaDoc gave a hint as to the Class to search for when implementing the concept. In reality, we would need to do further searching in order to identify this class, since the Abstract Concepts generally do not map directly to an Object Type.
- The JavaDoc was very useful in explaining the purpose and function of the concept.
- Was unsure as to whether I needed to implement any additional actions to the Toolbar, other than the provided "goHome", "goForward" and "goBack" actions.