Meeting Minutes

Dependency analysis for Kactus2

Client meeting
Date: 22.11.2012, 16:00 – 16:50
Number of meeting: client meeting 2/2012
Place: Tampere University of Technology, Korkeakoulunkatu 1, room TG309

Participants: Timo D. Hämäläinen (Client)
Joni-Matti Määttä (Project manager)
Mikko Honkonen (Project team member)
Tommi Korhonen (Project team member)

Meeting

1. Reviewed current status of the project. As the project plan is about finished, work has started on the functional specifications. The team has some questions for the client.
2. Question for client: What is the purpose of the "Used in" column shown in the UI mockup? It should list components that use that particular file. This should give some information about the criticality of a specific file. This information requires a separate scan, that scans all components in the library for files they use. The scan should be performed in the background, in a separate thread.
3. On the subject of threads, it would be wise to also perform the dependency analysis in a separate thread for the software to remain responsive. The "traffic lights" indicating status of the file would remain grey until all related files are scanned and then change to appropriate colour.
4. Question for client: Should the analysis be run every time the view is opened? Yes. The user could be shown the old result until the new scan is complete, and when it completes switch to the new result.
5. For creating manual connections, the team had the idea that the manual connection column heading could be used as a button to select between one- or two-way dependency. The heading would show the appropriate icon.
6. Two manual connections that connect the same files in different directions should be merged into one two-way connection, no merging with automatic connections.
7. What should be done if the user tries to create the same connection for the second time? Highlighting existing connections in grey seems like a good idea, but this might need checking during the development.
8. Should connections be created by dragging the mouse while mouse button is pressed or by clicking on the start file and then on the end file? The two-click

method is better. This would also allow pressing shift during the second click to create another connection with the same starting file.

9. It should be possible to document the connections, but at what scale? A short textual description should be enough. It's enough to allow descriptions only for manual connections, but it would be great if the automatic scan could also describe the dependency type (i.e. class instancing). This is likely out of scope of the project, but the possibility for later implementation should be left in.
10. How should a selected connection be highlighted? Doesn't matter as long as it's easy to notice, can be a border highlight or simple colour change. Not many colours are used, so the colour change should be fine.
11. Can a connection end and a new connection start on the same row in the same column? No, they need to be separated.
12. The team had the idea that folders could be collapsed and expanded from the very left side of the graph view. This should be fine, maybe a click on anywhere on the row should also expand/collapse, but there should at least be an indicator in the left side.
13. A clarification is needed for the dependency arrows: Does the arrow point from a dependant file to the file it depends on or the other way? From dependant file to file it depends on.
14. When should the dependency difference view be shown? The client thinks it would be best to show the normal dependency view after scan is completed, and allow the difference view to be shown separately. Manual dependencies are shown normally, but automatic ones are highlighted if they've been removed or added.
15. When the scan is run, we first check if any files are changed and only check for any dependency changes for the changed files – the dependencies can't have changed if the file hasn't changed.

Decisions made

1. “Used in” column now clear, this needs to be planned.
2. Dependency and component usage scans should be performed in a separate thread to not make main program unresponsive.
3. Analysis should be run every time the analysis view is opened.
4. Various clarifications on UI behavior for the functional specification.
5. Short textual descriptions to document the manual connections, with the possibility of later implementation of descriptions for automatic connections.
6. Dependency changes shown in a separate view that is shown only on demand
7. No need to check for dependency changes if the file has not changed.

Actions to follow

1. A lot of items were clarified in the meeting, so work can continue on the functional specification as planned.
2. Next meeting with the client likely in two weeks.