Meeting Minutes

Dependency analysis for Kactus2

Client meeting
Date: 1.11.2012, 12:05 – 13:05
Number of meeting: client meeting 1/2012
Place: Tampere University of Technology, Korkeakoulunkatu 1, room TG309

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

Meeting

1. Selected chairman for the meeting, Joni-Matti Määttä. Selected secretary for the meeting, Mikko Honkonen.

2. Meeting started at 12:05.

3. Reviewed the current status of project. Project plan is well underway, no problems with it at this point.

4. The client was asked his objectives for the project:
- The new features should have at least as good usability as the rest of Kactus2.
- The new features should be something that is useful for the user, "easing their life".
- The new features should clearly be a part of Kactus2, so it should be well integrated.

5. The client stated that dependency analysis, especially with manual dependencies, seems to be something that is wanted in the industry.

6. New feature proposed by client: Dependencies changed from previous run.

7. Checked tool status (Visual Studio, Redmine for project organization) for group members. Everything OK.

8. Discussion about how the file status (changed/unchanged) should be decided:
- Hash is probably the best option.
- Where would the hash be stored? What if multiple components have the same file, do they all have their separate hash? This seems to be useful.
- Do we run the dependency check even if hash hasn't changed?
- Is a more coarse way to check file changes (such as timestamp + size) needed to check if hash needs to be calculated.
- What about comments? Comment changes shouldn't break dependencies. However, this is quite sophisticated, and might not be relevant at this point.

8. Discussion about tool UI:

- There should always be room to create a manual dependency between files. A separate column dedicated for new manual dependencies was proposed, and was agreed to be a good idea.
- When a manual dependency is being created, the current connected rows should be highlighted.
- There should be filters to show/hide dependencies based on certain criteria: external files, dependency status, automatic/manual dependencies etc..

10. Should it be possible to remove automatic dependencies? Agreed that it might be more harmful than useful. Only possible to add to automatic dependencies.

11. Discussion about external files (such as libraries):
- Should be shown separately from other files. User should be able to tell where the files can be found: because they can be in different locations for different users, this should be a user-level option.
- Some sort of ignore list for external files, “don't care about this file”. Ignore list could be defined on both global level for entire Kactus2 and component level for certain component.

12. It should be possible to determine file extensions used for file types.

13. The analysis algorithms should be DLL files. This allows anyone to create their own algorithm for other file types.

14. Next meeting most likely after two weeks, precise time not yet decided.

15. Meeting ended at 13:05.

Decisions made

1. New feature to be planned: changes in dependencies.
2. File changes should be detected based on hash calculation, other methods aren't as reliable.
3. There should be a dedicated column for creating manual connections in the dependency graph view.
4. The dependency graph should have filters to show/hide items based on various criteria.
5. It will not be possible to manually remove automatic dependencies.
6. External file dependencies should be presented separately from other dependencies. User should be able to point their location.
7. External dependencies should also be ignorable, with user-lever configuration.
8. File extensions for file types should be specifiable.
9. Each supported file type should be a separate DLL, and it should be possible to easily add additional file type support.
10. Next meeting with the client when the project plan is getting ready.

Actions to follow

1. With client objectives clear, no more information is required from client for the project plan. Work will continue on writing the project plan.
2. New features to be added as tasks for the project plan.
3. Weekly team meetings will continue, next client meeting will be arranged after project plan is done and work can start on the functional specification.