

ANALYSIS OF JMVA

Analysis: Analyzed dependency matrix in IntelliJ Pro

Analysis: Found cyclical dependencies, can be eliminated with some patterns.

Analysis: ExactWizard is gui component

Analysis: ExactModel is model object, passed to SolverClient which then sends to SolverDispatcher;

Suggestion: No abstract layer between SolverClient and SolverDispatcher (use interface)

Analysis: SolverDispatcher: solves problem by delegating problem to other classes depending on type of problem.

Analysis: SolverAlgorithm looks very bad, too much dependency with SolverDispatcher

Suggestion: Some classes not used at all by JMVA in jmt.analytical, so should be moved out to reduce dependency

Analysis: Write a table of responsibilities of main classes and which packages they're part of

Cycle gui.exact.ExactModel cycle with analytical package (Exact model depends 136 times on analytical and analytical depends 68 times on ExactModel)

Design gui package should be top layer of application rather than analytical; gui should be top of the layer and analytical should depend on it. recommend splitting up gui and taking out ExactModel into a separate package

- gui contains ExactModel which depends on SolverClient (also in gui), which depends on SolverDispatcher (in analytical); thus if we remove ExactModel out

Code issues:

Naming Classes are poorly names (ExactWizard is a gui window), main method is obscured,

Comments Sparse comments and sometimes written in Italian, making difficult to figure out what's going on; encoding issues in comments (UTF8 vs ISO-8859-1).

-