Study Exceptional Entities

We used CodeCity and InFusion for the study of exceptional entities. Loading the entire project into InFusion gave us some initial idea about the quality of the code.

There seemed to be 1 Godclass and 15 data classes. The class hierarchies tends to be shallow and of average width. The classes itself have an average number of methods and are organized in rather fine-grained packages. The methods are from an average length and have rather simple logic and call few mehods (low coupling). (See webpage 1).

We notice that there seems to be some cases (20) of shotgun surgery, but these are all located in the model package and 1 in the changerecorders. These won't affect our reengineering project. We will check the other design errors in depth further when we focus on the destiller.

We can export this to MSE by using the FAMIX2.1 MSE exporter, which can be imported in codecity. This gives us an overall overview of the #packages and the classes that they contain. (see image 1)

We used a color filter to indicate what the god class and data classes are (brown = god class, purple = data classes.) However, if we look at those classes, it's a rather obvious reason why they are like that. The ModelManager is the God Class. Documentation about the ModelManager says: "This class is the entity that holds and maintains the entire famix and change model. All famixEntities and all changes that act upon those famixEntities are stored in this ModelManager." Therefore it's rather obvious that it performs a lot of work!

The big data class is the class Change in the model package, it seems to have a lot of setters and getters.

Also notice the 'parking spot', the flat square, which is the constants class in the model package. It holds a lot of constant values.

image 2, shows the destiller package, which we will have to focus on for our project. It shows clearly that there are 3 different packages contained, the svnconnection, popup and cd (changedistiller). The SVNLogEntryHandler is a data class, with a lot of getters that return info about a SVNEntry.

If we now only load the Destiller package into InFusion (webpage2), then we notice that inside, there is one case of feature envy (in extractChangesFromJavaFile), 1 case of intensive coupling (also in extractChangesFromJavaFile) and 3 cases of brain methods (convertChanges, extractChangesFromJavaFiles and iterateRevisions).

If we use codecity on the destiller package and we check the class blueprint, then we can see where the connections with the SVN part are made, these we will have to change.

(image findconnections)

we can check this by right clicking the SVNConnector and SVNLogEntryHandler class -> model -> mondrian -> clients collaboration. This results in (image link1 en link2)

//TODO: show link between the classes that the SVNConnector uses.