

Finding Classes

Read through the requirements looking for nouns and noun phrases. Make a list of these, changing the plurals to singulars, then get rid of all words and phrases that are obvious nonsense. Also, convert all the passive sentences to active sentences and all pronouns (I, it, he, they, etc.) to the noun it stands for. Make sure that the following has been included:

- * Physical objects, such as printers, robots, filing cabinets.
- * Non-physical entities such as accounts, files, windows.
- * Interfaces to the outside world, such as user interfaces and other programs.

Be careful with adjectives - they may be irrelevant, or they may indicate different behaviours, and therefore potentially different subclasses.

In the final system, these classes will usually outnumber the "real-world" objects considerably.

Finding Responsibilities

Responsibilities of an object include both the what it knows and the actions it can perform. When considering responsibilities, remember that they are the publicly available services, and that they specify what the object does, not how it does it.

Read through the requirements looking for verbs and use these as indicators of responsibilities. Assign the responsibilities to the classes they logically belong to, keeping in mind the following points.

- * State the responsibilities as generally as possible.
- * If you want to assign the same responsibility to several classes, consider how you might abstract out the common behaviour from these classes into a superclass.
- * Evenly distribute system intelligence so that you have a model of cooperating peers rather than one with a few super-intelligent objects and a lot of dumb objects.
- * Keep behaviour with related information (the principle of encapsulation)
- * Keep information about one thing in one place.
- * As you assign responsibilities, you will almost certainly discover that you need to add new classes.

Finding Collaborators

As you assign responsibilities, consider if the class is capable of fulfilling the responsibility by itself. If

not, what help does it need from other classes? These other classes will become collaborators.