Andrew Rodriguez
10/17/12

UML Report

Allow me to verbally describe two classes: Algorithm and AlgorithmAndOptions. Algorithm has one method: run, which takes an input object, a map of configuration parameters to values, and outputs an output object by running the algorithm. The second class is the AlgorithmAndOptions class which contains an Algorithm object, a map of configuration parameters to a set of allowable values, and a map of current parameter values. Besides the standard get and set methods, AlgorithmAndOptions supplies setOption which is a method that sets the

IntelligentInformationSystem

An IntelligentInformationSystem consists of a list of phases. Each phase can have multiple algorithms that can be used on it. We represent this scenario with a list of lists of AlgorithmAndOptions.

IntelligentInformationSystem
- List<List<AlgorithmAndOptions>> possibleProcesses
- instantiateEngines : IntelligentInformationSystem -> List<AnalysisEngine>
- addAlgorithmAndOptions : AlgorithmAndOptions -> ()

AnalysisEngine

An AnalysisEngine is similar to an IntelligentInformationSystem but it does not have to keep track of the possible algorithms that we can run at each step. The AlgorithmAndOptions present in the AnalysisEngine are only read from: we no longer need to pick out configuration options. run takes an input object and passes it through the algorithms to yield a final output object

AnalysisEngine
- List<AlgorithmAndOptions> processes
- run : Object -> Object

Sequence Diagram

In order to produce the set of allowable AnalysisEngine's, we must go through the IntelligentInformationSystem's steps and at each step suggest that each algorithm may be chosen. For each algorithm we have to allow all settings of the configuration options. We can recursively go through each step and construct the corresponding concrete phase that could be generated. We then add this phase to the list of all AnalysisEngine's we're going to return.

A makeshift sequence diagram:

IntelligentInformationSystem --- AlgorithmAndOptions

---> instantiateEngines

for each step
    for each algorithm

        <--- get available configuration options ----------

    modify each AnalysisEngine in the result list to include this algorithm

Sorry about the text diagram…