Отчёт по планнировщику HEART

Подоляко Илья Александрович БПМИ-176

29.01.19

Данный файл прилагается к диаграмме для уточнения того, что делает каждый файл в данном планнировщике.

Содержание:

Color.java

Color в данном планнировщике является main файлом, и всё остальное запускается из него. Особых действий нет кроме вывода состояния и выполнения других функций. Такая последовательность важных действий:
1) Создание объекта класса World на основе таски(в нашем случае это сразу и домен и таска)
2) Создание объекта класса WorldControl на основе world.flow(поле класса world)
3) Создание объекта класса Problem
4) Создание объекта класса Planner которому после присваивается один из объектов класса Heart, POP или HiPOP, т.е. по сути алгоритма планнирования.(Выбираем прямо в файле вручную)

World.class

Данный класс находится в библиотеке World.jar. Вероятнее всего там и находится парсер, который разделяют task'у на компоненты. Проследить дальнейшие связи трудно, все задача сначала переносится в класс flow, потом ещё в другие менее важные, но для осознания это не важно, так взаимодействие уже с задачей происходит в другом классе.

WorldControl.java

В данном файле происходит уже непосредственно помещение всей task'и по различным категориям(предикаты, эффекты и тд). Класс имеет следующие поля:
pre - предикаты
eff - эффекты
constr - конструкции
method - методы
: - обозначение "такой что"
? - используется для превращения предмета в переменную (для использования действий типа take)

Самый большой интерес представляет создание класса Problem из подготовленной таски(это 4-ый пункт файла Color.java), тут происходит разделение на начальное состояние, цель и домен. Происходит поиск целевого действия(action) и начального.(Вопрос почему только они называются тут действиями, если это скорее состояния). Работает это всё так:
1) Во-первых с помощью функции domain данного класса, создаётся объект класса Domain. Сам класс Domain состоит из некой пользовательской коллекции из библиотеки, в своб очередь функция разделяет наш benchmark на группы:
1) Берутся объявленные сущности(Entity statement), берутся сущности с "::"(объявлению любых сущностей в benchmark'e), и fluent'ы.(что странно все флюенты закоменчены, хотя это не точно, но я видел комментарий с двумя слешами, а флюенты именно так и записаны). После этого они записываются в лист сущностей.
2) После этого берётся лист сущностей и с помощью цикла созадются флюенты, из отдельного класса WorldFluent. Конструктор определяет там свойства и их негативность, также операторы,параметры и привязывет флюенту к данным сущностям(видимо чтобы потом обращаться обратно). После этого запихиватся кванторы через цикл, которые были определены и их значения.(Для этого есть отдельное поле в WorldFluent сигнатура)
3) После этого всё возвращается к классу WorldControl, в котором мы ищем наще целевое и начальное состояние, предварительно удалив его из домена, и создаётся объект класса Problem в котором лежат только пока наши целевые, начальные значения, и готовый домен.
Итого: Это одна из важнейших частей программы, так как именно тут идёт подготовка для работы с планнировщиком.

Planner.java

В данном классе определяется цель для планнировщика, задаётся эвристика и агенда. В файле color.java одновременно происходит создание этого объекта на основе объекта класса Problem и выбор алгоритма, с

помощью которого происходит решение проблемы. Есть функция solve, которая уже запускает работу алгоритмов и выкидывает ошибки если нет решения или ключевых вещей(цели).