

## Design of the Scheduler

The Scheduler consists of the model and controller package. In matters of the MVC software architecture there will be no view packages, because the view is implemented by the web interface. The model package consists of the sub-packages strategy and data which will be described in the following sections.

### Strategy

The strategy package holds the core classes of the genetic algorithm implementation. The classes are accessed through an Abstract Factory class, as a matter of fact we implemented the Abstract Factory pattern. The Abstract Factory class provides methods to instantiate 7 classes:

- **Schedule:** Schedule is the chromosome of the genetic algorithm which is developed through several generations.
- **Crossover:** Crossover provides the crossover operation which takes two Schedules and creates a new Schedule based on those.
- **Mutation:** Mutation provides the mutation operation which moves some courses to a new slot position thus altering a Schedule.
- **Calculate Hard Fitness:** Calculate Hard Fitness provides a method which measures the degree to which a Schedule fulfills the hard constraints.
- **Calculate Soft Fitness:** Calculate Soft Fitness provides a method which measures the degree to which a Schedule fulfills the soft constraints.
- **Setup:** Setup provides a method to initialize a new schedule.
- **Algorithm:** Algorithm provides a method to run the genetic algorithm using the other classes.

Each of these classes implements an interface which is associated to the Abstract Factory interface. Thus we can individually exchange the concrete implementations to provide a wide range of strategies and make the Scheduler configurable.

### Hard Constraints

- (1) No two CourseElementInstances shall overlap in TimeSlot and Room.
- (2) No two CourseElementInstances of the same Year shall overlap in TimeSlot if they are mandatory. A CourseElementInstances is mandatory if the sum of all CourseElementInstances duration equals the duration of the CourseElementInstances they are associated with.
- (3) No CourseElementInstance shall take place in a Room that does not provide all required Features.
- (4) No two CourseElementInstances held by the same Person shall overlap in TimeSlots.
- (5) No CourseElementInstance shall take place in a TimeSlot where the Person does not work(!personPrefersTimeSlot with priority 0).
- (6) The allocation of a CourseElementInstance to a Room shall respect the restriction of the !courseElementInstancePrefersRoom.
- (7) The allocation of a CourseElementInstance to a TimeSlot shall respect the restriction of the !courseElementInstancePrefersTimeSlot.
- (8) The allocation of a CourseElementInstance to a Room shall respect the restriction of the RoomPrefersTimeSlot.

### Soft Constraints

- (1) A CourseElementInstance should take place in a Room that does provide all preferred Features.

- (2) A CourseElementInstance should take place in a TimeSlot where the Person would like to hold the CourseElementInstance.
- (3) The allocation of a CourseElementInstance to a Room should respect the preferences of the CourseElementInstancePreferesRoom.
- (4) The allocation of a CourseElementInstance to a TimeSlot should respect the preferences of the CourseElementInstancePreferesTimeSlot.
- (5) The allocation of a CourseElementInstance to a Room should respect the preferenced of the RoomPreferesTimeSlot.

## **Data**

The data package holds the classes for the model which is mapped by the actual design of the relational database scheme.

## **► Attachments**