University timetable scheduling

Bachelor Project Report

Aron Fiechter

2018

Advisors: Michele Lanza, Andrea Mocci, Luca Ponzanelli

Contents

Abstract

Creating and managing timetables of courses is an issue for many institutions because of the various constraints that need to be respected in the planning. This is also a problem for the Dean's office of the Faculty of Informatics at USI, which currently creates timetables manually. Automating this process by means of state of the art tools to solve planning problems can increase productivity. In this bachelor project report we describe the design and implementation of a web application that offers an automated timetable creator wrapped in a user friendly web interface.

1 Introduction

1.1 Motivation

Creating a timetable is an issue for many institutions because of the various constraints that might need to be respected, such as different room sizes, instructor availability, elective courses, different frequency of courses. For example, this is a problem for the Faculty of Informatics at USI as well: the Atelier courses need to be placed in the afternoons, there are three rooms that can contain 60 people and four that can contain 30 people, there are different elective courses, and not all professors are available everyday.

1.2 Goal

The goal of this project is to create a web application (a module of the MyUSI platform) that allows to automatically create timetables given a set of constraints, and also customize proposed solutions; the resulting calendars should also be available to single students in the form of semester schedules, or personalised schedules in the case of elective courses and courses from different years.

Additional Goals

Once the basic system is implemented, possible extensions include managing timetables of the other faculties, adding single non-lecture events to schedules (such as seminars, talks), visualising statistical information about room usage.

2 State of the art

2.1 Planning problem

2.2 OptaPlanner

3 Approach

3.1 Technologies

The core solver implementation makes use of a state of the art library for solving planning problems called OptaPlanner[1]. This tool requires the user

---

[1]https://www.optaplanner.org/

to define a problem domain model, choose algorithms to use in the solving and ???.
The server side of the web application is implemented in Scala.
The client side uses Polymer 2.4 with TypeScript.

3.2 Domain model

3.3 Constraints

3.4 Interface

3.5 Web application UI

4 Evaluation

4.1 Case study on real data

4.2 Examples results

5 Conclusion

5.1 Future work

6 Bibliography

7 Appendix