Project Description

What problems does the project solve

Course Timetabling

- A university can have different lecture halls with different capabilities, capacities and purposes.
- It can also have students that are taking different combination of courses.
- Therefore, an intelligent solution is required to optimize the usage of halls and period times.
- Course timetabling a course schedule that doesn't create any conflicts to the students attending the course.
- It can simulate a variety of situations like fewer classrooms or changes in course class requirements.

Course Management

- Manages timetable modifications.
- Searches for alternatives that have a minimal impact on the current timetable
- Communicates these changes to affected students and other systems.
- Each alternative produced by the solver provides information on any additional student conflicts created and any other preferences that may be violated.
- Changes can be customized to allow room-only, time-only swaps, or both.

Event Management

- A university can host more then just classes and lectures like Guest speakers, club meetings study sessions.
- UniTime handles the creation of class meetings and examinations in the events calender Automatically.
- Requesting and searching for events can be handled via the Web Interface that is available for all student and staff members.

Examination Timetabling

- Builds a complete exam schedule.
- Minimizes number of conflicting exam placements for students.
- Limits the number of exams per day for students.
- Creates schedules for midterm and final exams.

Student Scheduling

- UniTime supports 2 kinds of student scheduling.
- Batch Scheduling: UniTime automatically enrolls pre-registered students to their suitable classes.

- Online Scheduling: UniTime supports immediate real-time requests to change or apply to classes.
- UniTime's automatic enrollment is optimized to respect student preferences and course structure.

Techniques used to obtain information

We take an opportunistic approach (combinations of top-down and bottom-up approaches) to program comprehension, namely, the Integrated Metamodel[1] for Program Comprehension.

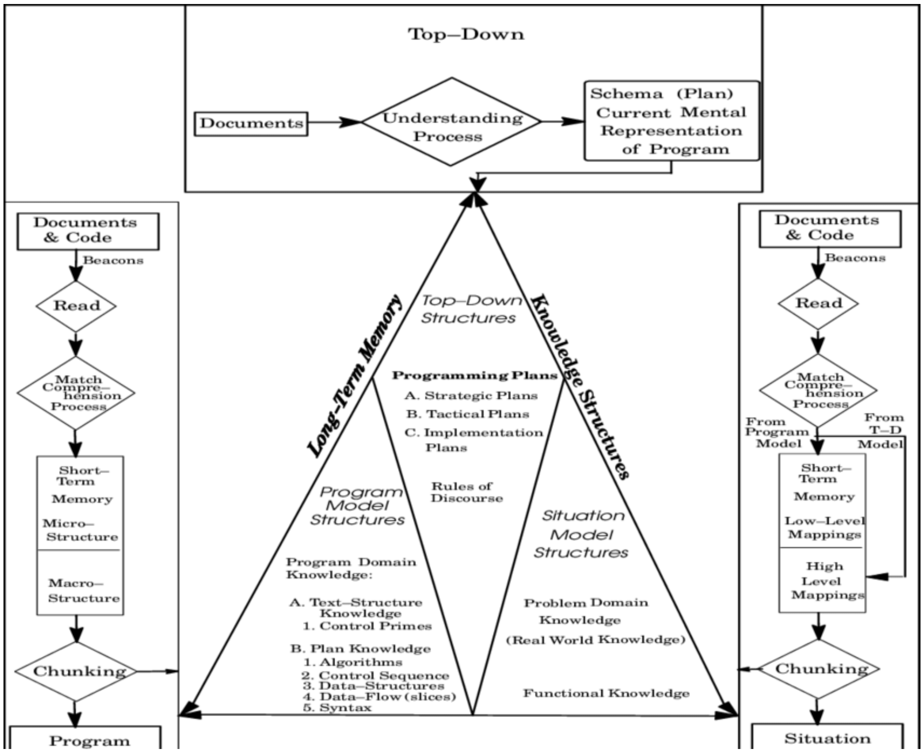

Figure 1: The integrated metamodel for program comprehension

Timetabling

- Establish a partial top-down model for the timetabling UniTime component using the provided documentation on the component.
- Refine the top-down model by running the application locally:

[1] Mayrhauser, Anneliese & Vans, A. Marie. (1995). Program comprehension during software maintenance and evolution. Computer. 28. 44 - 55. 10.1109/2.402076.

- Log in with an admin account, and test Courses > Course Timetabling > Timetable Grid.
- Log in with a manager account, and test Courses > Course Timetabling > Solver.