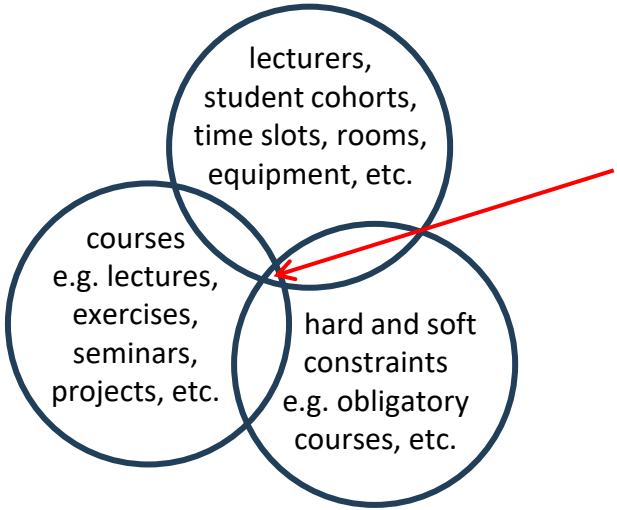
Universität Potsdam

# AI-Based Tool for Curriculum-Based Course Timetabling at the University of Potsdam

Christian Dohrmann, Ulrike Lucke, Torsten Schaub, **Sebastian Schellhorn**

# Motivation

What is a Curriculum-Based Course Timetabling (CB-CTT) problem?




Goal: seeking for a conflict-free timetable for teaching from the beginning, instead of resolving conflicts on demand

## Motivation

Traditional approach at UP (Faculty of Science)

- Coordinate/schedule obligatory lectures/courses for large lecture halls (on faculty level)
  - Involves a lot of human efforts, agreements and time
  - May contain/cause redundancies
  - May result in suboptimal timetables and/or conflicts
- Plan other courses after obligatory lectures/courses are set (on institutes level)
  - May ignore needs of players of other institutes/faculties
  - May run into conflicts at a later stage

## AI-Based Approach

- Represent hard and soft constraints by a logic program
  - Using Answer Set Programming (ASP)
  - ➢ Human readable and elaboration tolerant
  - ➢ Get solutions by state-of-the-art solvers like *clingo*

# AI-Based Approach

- Represent hard and soft constraints by a logic program
  - Using Answer Set Programming (ASP)
  - ➢ Human readable and elaboration tolerant
  - ➢ Get solutions by state-of-the-art solvers like *clingo*

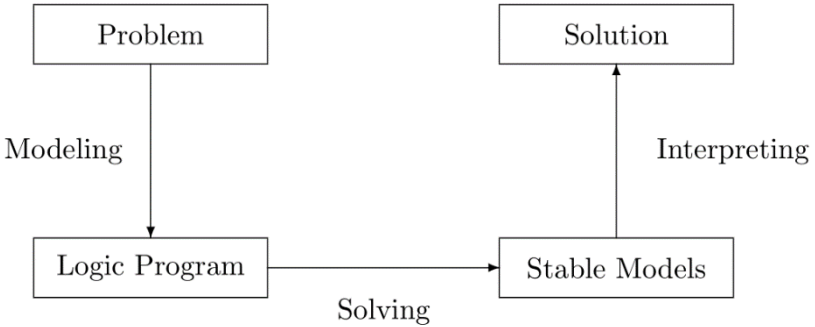



Figure 3: Schematic representation of declarative problem solving process.

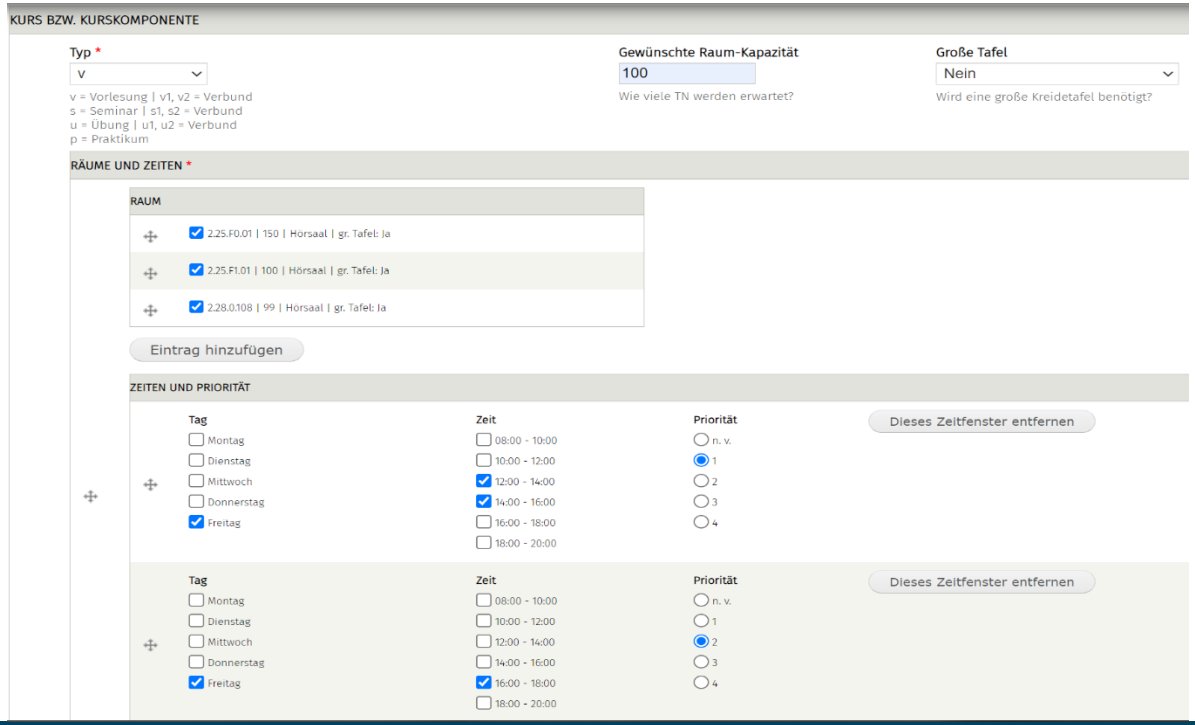
## AI-Based Approach

- Represent hard and soft constraints by a logic program
  - Using Answer Set Programming (ASP)
  - ➢ Human readable and elaboration tolerant
  - ➢ Get solutions by state-of-the-art solvers like *clingo*

- Develop a web interface
  - ➢ Minimizing communication among planners and lecturers
  - ➢ Reduce (some) redundancies of creating a course, linking to modules and booking a room

# Prototypical Web Interface (1)

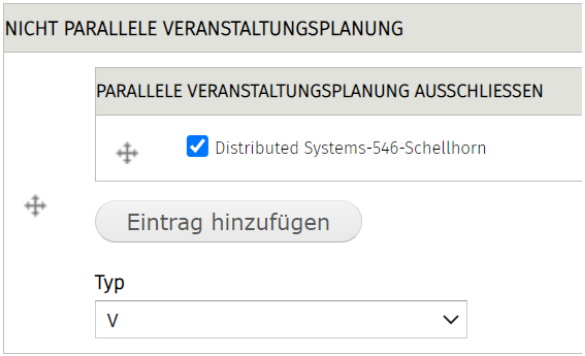
# Prototypical Web Interface (2)



# Prototypical Web Interface (3)

Relating courses

- Not parallel
- Parallel
- Simultaneous
- Consecutive



## Modeled Hard Constraints

H0. No course component can take place in parallel to its lecture.

H1. Each course must be scheduled.

H2. Obligatory lectures of a cohort must be scheduled in different time slots.

H3. Two courses cannot take place in the same room and time slot.

H4. A lecturer cannot be scheduled in parallel.

H5-8. Respect relations parallel, not parallel, simultaneous and consecutive.

Logic representation of H1.

```
1{ book(ID, Course) : availability_course(ID, Course) }1 :- course(Course).
```

# Modeled Soft Constraints

S0. Try to serve prioritized availability.

S1. Try to fit students that are expected to attend into a selected room.

S2. Courses of a cohort should be scheduled in different time slots.

S3. Reduce gaps of courses regarding cohorts and lecturers.

S4. Try to schedule consecutive courses into the same room.

S5. Cohorts and lecturers should not exceed 4 courses per day.

S6. Reduce travel time for cohorts and lecturers.

S7-13. Try to satisfy particular equipment (e.g. large board, etc.).

# Output

```
Optimization: 34 1 292
Answer: 92

Optimization: 34 0 291
OPTIMUM FOUND

Models       : 92
  Optimum    : yes
Optimization : 34 0 291
Calls        : 1
Time         : 170.129s (Solving: 169.69s 1st Model: 0.02s Unsat: 154.04s)
CPU Time     : 0.000s
```

## Results and Future Work

- Optimal and conflict-free timetable in a couple of seconds
- Fair, transparent and elaboration tolerant approach
- Reduces involved time, human efforts and communication overhead

- Involve more data
  - Implement interfaces to access data
  - Reuse previous course catalogues and availabilities
  - Previous linked modules
- Implement conflict handling
- Automated integration of resulting timetable to course catalogue

## Questions?

- Is this idea new?
  - No
- Why not using existing approaches then?
  - Elaboration tolerance and flexible
  - Human readable modeling language
  - Easy to handle
  - Addresses particular needs (of UP)
  - (free) state-of-the-art solver

## More questions?

**Christian Dohrmann**

Institute of Mathematics, University of Potsdam
Leibniz Institute for Science and Mathematics Education Kiel
christian.dohrmann@uni-potsdam.de

**Torsten Schaub**

Institute of Computer Science,
University of Potsdam
torsten.schaub@uni-potsdam.de

**Ulrike Lucke**

Institute of Computer Science,
University of Potsdam
ulrike.lucke@uni-potsdam.de

**Sebastian Schellhorn**

Institute of Computer Science,
University of Potsdam
sebastian.schellhorn@uni-potsdam.de