UniTime
Best Practices

June 2017

Tomáš Müller, Zuzana Müllerová
Agenda

• Short introduction of UniTime & workshop instance
• Administration (installation, setup, data exchange, modeling, …)
• Data Entry (rooms, instructors, courses, …)
• Timetabling
• Other Features (reporting, scripts, …)
• Conclusions

Presentation available at www.unitime.org/present/apereo17-workshop.pdf
What is UniTime?

• Comprehensive academic scheduling solution
• Four components
  • Course timetabling
  • Examination timetabling
  • Student scheduling
  • Event management
• Open source, web-based, written in Java using modern technologies
• Using state-of-the-art optimization algorithms
• Distributed data entry and timetabling in multi-user environments
• First used at Purdue University in 2005
• Apereo project since 2015
Workshop Demo Instance

- A college with about 6,000 students
- 24 departments entering the data
- Distributed data entry, centralized timetabling
  - Distance learning timetabled separately
  - For this workshop, the timetabling has been decentralized
- Shared resources (especially rooms)
- Student demands based on curricula
- Loosely based on the College of Education, Masaryk University
- Web: demo.unitime.org/workshop
- Accounts: user001/pwd001 … user051/pwd051
<table>
<thead>
<tr>
<th>User</th>
<th>Department</th>
<th>Courses</th>
<th>Classes</th>
<th>Instructors</th>
</tr>
</thead>
<tbody>
<tr>
<td>20, 26, 48</td>
<td>Art</td>
<td>57</td>
<td>154</td>
<td>43</td>
</tr>
<tr>
<td>38, 40</td>
<td>Biology</td>
<td>33</td>
<td>111</td>
<td>41</td>
</tr>
<tr>
<td>14, 49</td>
<td>Civics</td>
<td>58</td>
<td>95</td>
<td>21</td>
</tr>
<tr>
<td>17, 18, 28, 42</td>
<td>Czech</td>
<td>114</td>
<td>225</td>
<td>32</td>
</tr>
<tr>
<td>15, 30, 36</td>
<td>English</td>
<td>157</td>
<td>250</td>
<td>50</td>
</tr>
<tr>
<td>1, 22</td>
<td>French</td>
<td>56</td>
<td>81</td>
<td>18</td>
</tr>
<tr>
<td>24, 33</td>
<td>Geography</td>
<td>25</td>
<td>43</td>
<td>19</td>
</tr>
<tr>
<td>8, 12, 34</td>
<td>German</td>
<td>78</td>
<td>133</td>
<td>20</td>
</tr>
<tr>
<td>27, 47</td>
<td>Health Ed</td>
<td>21</td>
<td>39</td>
<td>17</td>
</tr>
<tr>
<td>6, 32</td>
<td>History</td>
<td>39</td>
<td>93</td>
<td>49</td>
</tr>
<tr>
<td>4, 45</td>
<td>IT</td>
<td>49</td>
<td>95</td>
<td>20</td>
</tr>
<tr>
<td>9, 10</td>
<td>Language</td>
<td>23</td>
<td>89</td>
<td>14</td>
</tr>
<tr>
<td>23, 25, 29</td>
<td>Mathematics</td>
<td>53</td>
<td>104</td>
<td>27</td>
</tr>
<tr>
<td>41, 51</td>
<td>Music</td>
<td>59</td>
<td>196</td>
<td>17</td>
</tr>
<tr>
<td>37, 46</td>
<td>Pedagogy</td>
<td>17</td>
<td>76</td>
<td>28</td>
</tr>
<tr>
<td>2, 7, 31, 35, 43</td>
<td>Physics</td>
<td>170</td>
<td>416</td>
<td>84</td>
</tr>
<tr>
<td>5, 19</td>
<td>Prime Ped</td>
<td>34</td>
<td>99</td>
<td>16</td>
</tr>
<tr>
<td>16</td>
<td>Psychology</td>
<td>40</td>
<td>109</td>
<td>14</td>
</tr>
<tr>
<td>21, 39</td>
<td>Physical Ed</td>
<td>24</td>
<td>64</td>
<td>16</td>
</tr>
<tr>
<td>11, 50</td>
<td>Russian</td>
<td>83</td>
<td>156</td>
<td>18</td>
</tr>
<tr>
<td>13</td>
<td>Social Ed</td>
<td>89</td>
<td>136</td>
<td>75</td>
</tr>
<tr>
<td>3, 44</td>
<td>Special Ed</td>
<td>135</td>
<td>231</td>
<td>74</td>
</tr>
</tbody>
</table>
Administration
Installation

• UniTime can be downloaded from http://builds.unitime.org
• Installation Instructions: help.unitime.org/Timetabling_Installation
  • See Windows / Linux specific notes at the bottom of the page

• Hardware Requirement
  • Any system capable of running Java and MySQL/Oracle
  • Linux is recommended, should have enough memory, could be a VM
  • E.g.: 8 cores, 12 GB RAM, 100 GB drive
  • Oracle database is recommended for production environments

• Prerequisites
  • Java, MySQL or Oracle Database, Apache Tomcat
  • For larger institutions (and especially when students can access)
    • Cluster containing web servers and remove solver serves

Do not forget the -Xmx parameter and the MySQL/Oracle JDBC driver!
Cluster

- One or more web servers (Apache Tomcat / UniTime.war)
- One or more remote solver servers (Java)
- JGroups Clusters
  - Hibernate L2 Cache (web servers only)
  - Solver Cluster (RPCs)
  - Online Student Scheduling Server replications (optional)
Customization

- Custom properties
  - Application Configuration page
  - Custom properties file
- Custom CSS, welcome message, disclaimer, menu content & style
- Much more, see the Application Configuration page for the list

Authentication

- By default, the Users page is used
- CAS or LDAP can be configured (or anything else using Spring Security)
- We need an external ID of an authenticated user
  - Students, Instructors, Timetable Managers
  - No match: No Role or Anonymous (can be disabled)

See http://help.unitime.org/Customizations for more details.
Localization

- Current locales: en, en_UK, cs
- Use en_UK to switch UniTime to use 24h times and dd.mm.yyyy dates
- Default can be set using unitime.locale property
  - Can be changed per user (User Settings),
  - or for HTTP session with the locale parameter
- Other translations exist but are mostly incomplete and/or have not been contributed back to UniTime

Translations

- Translations are provided in property files
- Zanata can be used to provide translations

See http://help.unitime.org/Localization for more details.
Initial Configuration

• User Roles & Permissions
  • Each permission contains a check (e.g., a schedule manager can only edit classes of his/her department when allowed by session status)

• Statuses *(Initial Data Load, Data Entry, Timetabling, Published, Closed)*

• Instructional Types *(Lecture, Lab, Recitation, …)*

• Room Types *(Classroom, Computing Lab, Outside Location, …)*

• Room Feature Types *(Seating Type, Room Configuration, A/V, …)*

• Many more (course types, instructional methods, position types, …)
  • See items under Administration > Other menu

• Solver Configuration (could be done much later, based on the data)

UniTime contains good default data for these.
Academic Session

- Dates
  - Session start date
  - Examination start date
  - Holidays, ...
- Date Patterns
- Time Patterns
- Examination Periods
- Departments
- Subject Areas
- Solver Groups
- Timetabling Managers

Department

- Most of the UniTime data are related to a particular department
- Instructors, room sharing, managers (permissions), solver groups, etc.
- External manager department for classes that are to be timetabled outside of the subject area (e.g., computing labs, large lecture rooms)

See the online demo http://demo.unitime.org for some examples.
Date Patterns

- Weeks of instructions (All weeks, Even/Odd weeks, Week 5, …)

Time Patterns

- Possible time slots within a week
Data Exchange

- A lot of the data can be imported via XML
- Departments, subject areas, rooms, staff, …
- Beware: rooms and staff do not get imported directly
  - Rooms: use Update Data on the Buildings page
  - Staff: use Manage Instructor List on the Instructors page
- Course Offerings XML can be used to import just courses, the whole structure, or anything in between

APIs

- Mostly to get data out of UniTime in real time
- Can be extended as needed
- Can be also used to import/export XMLs programmatically

See http://www.unitime.org/uct_interfaces.php for the list of XML interfaces.
See https://goo.gl/L1sEVN for UniTime 4.2 APIs.
UniTime Setup

Academic session roll-forward

• When there already are academic sessions in UniTime
• Roll-forward most of a session’s data
• Possible to combine data from different sessions
• After roll-forward, it is possible to still use XMLs to update the data

Timeline

Data Entry

Automated Timetabling

Timetable published

Classes start

Additional Timetable Changes

Roll-forward

Data Entry

Next semester timetabling

Student Scheduling
Best Practices: UniTime Setup

• Make sure UniTime has enough memory, especially for the solver
• Departments & subject areas need to be carefully defined
  • Instructors, room sharing, data entry / access
• Distributed or centralized data entry and/or timetabling
  • Most often: distributed data entry, centralized timetabling
• Student Course Demands
  • Last-like demands are the easiest to get, but may not be as good
  • Student course requests allows for individual students to be considered
  • Curricula are good, when available
    (can be combined with last-likes for optional course estimates)
## Rooms

- Each department may have a different set of rooms
- Some times may be unavailable or given to a different department

### Room coordinates (GPS), travel times (in minutes)
- Room groups and room features
- Rooms / non-university locations
Best Practices: Rooms

• Room features can be categorized by feature types (seating type, desk arrangements, audio/video, …)

• Having good room groups and room features helps with preferences
  • Think about the faculty preferences you may get (E.g., I want a room with a white board and a data projector, which could be used both at the same time)

• Approved events can be used to block certain times in a room.

• There can be pseudo rooms that do not check for overlaps (E.g., off-campus, instructor’s office, hospital)

• Dept. room preferences are useful to minimize use of a room
  • Prohibited … cannot be used (for what-if scenarios)
  • Strongly Discouraged … only when there is a direct preference
  • Discouraged … minimize use of the room (avoid if possible)
Instructors

- Each department has a list of instructors
  - Connection between departments through external id
- Instructor availability (prohibited times)
- Instructor preferences & requirements
  - Time, room, distribution
Best Practices: Instructors

• Use instructor preferences in combination with subpart preferences
  • Especially time availability and preferences
• Useful Distribution Preferences *
  • Max N Hours
  • N Hour Work Day
  • Max Blocks
  • Max Breaks
  • N Days a Week

*) Some need to be registered first, see https://goo.gl/ufqW1t for the scripts.
# Instructional Offering

<table>
<thead>
<tr>
<th>Course</th>
<th>Limit</th>
<th>Type</th>
<th>Minutes</th>
<th>Per Week</th>
<th>Time</th>
<th>Room</th>
<th>Distribution</th>
<th>Instructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA 170</td>
<td>40</td>
<td>Lecture</td>
<td>50</td>
<td>1 x 50</td>
<td></td>
<td>Classroom</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STAT 170</td>
<td></td>
<td>Laboratory</td>
<td>150</td>
<td>3 x 50</td>
<td></td>
<td>EDUC CompPr</td>
<td>Same Room</td>
<td></td>
</tr>
<tr>
<td>Lec 1</td>
<td>40</td>
<td>Lecture</td>
<td>50</td>
<td>1 x 50</td>
<td></td>
<td>ThtrSeat Classroom</td>
<td>G. Newman</td>
<td></td>
</tr>
<tr>
<td>Lab 1</td>
<td>20</td>
<td>Laboratory</td>
<td>150</td>
<td>3 x 50</td>
<td></td>
<td>EDUC CompPr</td>
<td>Same Room J. Smith</td>
<td></td>
</tr>
<tr>
<td>Lab 2</td>
<td>20</td>
<td>Laboratory</td>
<td>150</td>
<td>3 x 50</td>
<td></td>
<td>EDUC CompPr</td>
<td>Same Room J. Smith</td>
<td></td>
</tr>
</tbody>
</table>
## Instructional Offering

### Course Offerings

<table>
<thead>
<tr>
<th>Course</th>
<th>Limit</th>
<th>Type</th>
<th>Minutes Per Week</th>
<th>Time Pattern</th>
<th>Room</th>
<th>Distribution</th>
<th>Instructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA 170</td>
<td>40</td>
<td>Lecture</td>
<td>50</td>
<td>1 x 50</td>
<td>Classroom</td>
<td>Same Room</td>
<td>G. Newman</td>
</tr>
<tr>
<td>STAT 170</td>
<td></td>
<td>Laboratory</td>
<td>150</td>
<td>3 x 50</td>
<td>EDUC</td>
<td>Same Room</td>
<td>J. Smith</td>
</tr>
<tr>
<td>Lec 1</td>
<td>40</td>
<td>Lecture</td>
<td>50</td>
<td>1 x 50</td>
<td>ThtrSeat</td>
<td>Classroom</td>
<td></td>
</tr>
<tr>
<td>Lab 1</td>
<td>20</td>
<td>Laboratory</td>
<td>150</td>
<td>3 x 50</td>
<td>EDUC</td>
<td>Same Room</td>
<td>J. Smith</td>
</tr>
<tr>
<td>Lab 2</td>
<td>20</td>
<td>Laboratory</td>
<td>150</td>
<td>3 x 50</td>
<td>EDUC</td>
<td>Same Room</td>
<td>J. Smith</td>
</tr>
</tbody>
</table>
### Instructional Offering

#### Course Offerings

#### Scheduling Subparts

<table>
<thead>
<tr>
<th>Course</th>
<th>Limit</th>
<th>Date</th>
<th>Pattern</th>
<th>Minutes Per Week</th>
<th>Time Pattern</th>
<th>Time</th>
<th>Room</th>
<th>Distribution</th>
<th>Instructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA 170</td>
<td>40</td>
<td>Full Term</td>
<td>50</td>
<td>1 x 50</td>
<td>Classroom</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STAT 170</td>
<td>Introductory statistics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Laboratory</td>
<td>40</td>
<td>Full Term</td>
<td>150</td>
<td>3 x 50</td>
<td>EDUC CompPr</td>
<td>Same Room</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lec 1</td>
<td>40</td>
<td>Full Term</td>
<td>50</td>
<td>1 x 50</td>
<td>ThtrSeat Classroom</td>
<td></td>
<td>G. Newman</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lab 1</td>
<td>20</td>
<td>Full Term</td>
<td>150</td>
<td>3 x 50</td>
<td>EDUC CompPr</td>
<td>Same Room</td>
<td>J. Smith</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lab 2</td>
<td>20</td>
<td>Full Term</td>
<td>150</td>
<td>3 x 50</td>
<td>EDUC CompPr</td>
<td>Same Room</td>
<td>J. Smith</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Instructional Offering

### Course Offerings

### Scheduling Subparts

### Classes

<table>
<thead>
<tr>
<th>Course</th>
<th>Section</th>
<th>Type</th>
<th>Limit</th>
<th>Date Pattern</th>
<th>Minutes Per Week</th>
<th>Time Pattern</th>
<th>Room</th>
<th>Distribution</th>
<th>Instructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA 170</td>
<td>Lec 1</td>
<td>Lecture</td>
<td>40</td>
<td>Full Term</td>
<td>50</td>
<td>1 x 50</td>
<td>Classroom</td>
<td>ThtrSeat Classroom</td>
<td>G. Newman</td>
</tr>
<tr>
<td></td>
<td>Lab 1</td>
<td>Laboratory</td>
<td>20</td>
<td>Full Term</td>
<td>150</td>
<td>3 x 50</td>
<td>EDUC CompPr</td>
<td>Same Room</td>
<td>J. Smith</td>
</tr>
<tr>
<td></td>
<td>Lab 2</td>
<td>Laboratory</td>
<td>20</td>
<td>Full Term</td>
<td>150</td>
<td>3 x 50</td>
<td>EDUC CompPr</td>
<td>Same Room</td>
<td>J. Smith</td>
</tr>
</tbody>
</table>
Best Practices: Courses

• There can be multiple configurations
  (with different instructional method, e.g., traditional x online)

• If a class does not follow a standard time pattern, it could be split

• Reservations can be used to direct students to the appropriate configurations / classes

• Use cross-lists whenever a course is offered under multiple names

• Meet together constraint can be useful, but use it wisely

• Externally managed departments can be used to timetable some classes as a different problem (large lecture rooms, computing labs)
  • It is possible to move control of such classes from the department of the course to the external department with a status change
Best Practices: Subparts and Classes

- Minimal room size: room ratio times class limit
- Classes of a scheduling subpart are spread in time (can be disabled)
- Only matching time patterns are visible
  - E.g., minutes per week = number of meetings × minutes per meeting
- Too many start times result in a bad timetable
  - Too many small holes, hard to swap rooms
Best Practices: Preferences

- Preferences can be set on scheduling subpart, class, or instructor
- The end result is displayed on the class and used by the solver
- Put as many preferences as possible on instructors and subparts
  - Class overrides can be highlighted in yellow

---Preferences---

<table>
<thead>
<tr>
<th></th>
<th>Limit</th>
<th>Date Pattern</th>
<th>Minutes Per Week</th>
<th>Time Pattern</th>
<th>Time</th>
<th>Room</th>
<th>Distribution</th>
<th>Instructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA 170</td>
<td>40</td>
<td>Full Term</td>
<td>50</td>
<td>1 x 50</td>
<td></td>
<td>Classroom</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STAT 170</td>
<td>Introductory statistics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lecture</td>
<td>40</td>
<td>Full Term</td>
<td>150</td>
<td>3 x 50</td>
<td></td>
<td>EDUC CompPr</td>
<td>Same Room</td>
<td></td>
</tr>
<tr>
<td>Laboratory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lec 1</td>
<td>40</td>
<td>Full Term</td>
<td>50</td>
<td>1 x 50</td>
<td></td>
<td>ThtrSeat</td>
<td>G. Newman</td>
<td></td>
</tr>
<tr>
<td>Lab 1</td>
<td>20</td>
<td>Full Term</td>
<td>150</td>
<td>3 x 50</td>
<td></td>
<td>EDUC CompPr</td>
<td>Same Room</td>
<td>J. Smith</td>
</tr>
<tr>
<td>Lab 2</td>
<td>20</td>
<td>Full Term</td>
<td>150</td>
<td>3 x 50</td>
<td></td>
<td>EDUC CompPr</td>
<td>Same Room</td>
<td>J. Smith</td>
</tr>
</tbody>
</table>
Importance of having good input data

- The solution will only be as good as the input data
- No preferences
  - A class can end up anywhere (unpopular time, wrong room)
- Too many requirements
  - Impossible to find a complete timetable
  - Too many student conflicts
  - Difficult to make modifications
Constraint-based Solver

• Can be used in modes between manual and fully automated
• State of the art
  ◦ Work published a number of research papers
  ◦ Winner of the International Timetabling Competition 2007
• Easy to extend

Suggestions

<table>
<thead>
<tr>
<th>Score</th>
<th>Class</th>
<th>Date</th>
<th>Time</th>
<th>Room</th>
<th>Students</th>
</tr>
</thead>
<tbody>
<tr>
<td>+15.2</td>
<td>POL 101 Lec 3</td>
<td>Full Term</td>
<td>TTh 12:00p → TTh 7:30a</td>
<td>BRNG 2280</td>
<td>+11</td>
</tr>
<tr>
<td>+31.7</td>
<td>POL 101 Lec 3</td>
<td>Full Term</td>
<td>TTh 12:00p → TTh 10:30a</td>
<td>BRNG 2280</td>
<td>+36 (h+3)</td>
</tr>
<tr>
<td></td>
<td>HIST 342 Lec 1</td>
<td>Full Term</td>
<td>TTh 10:30a → TTh 1:30p</td>
<td>BRNG 2280 → BRNG 2290</td>
<td></td>
</tr>
<tr>
<td>+36.6</td>
<td>POL 101 Lec 3</td>
<td>Full Term</td>
<td>TTh 12:00p → TTh 10:30a</td>
<td>BRNG 2280</td>
<td>+36 (h+4)</td>
</tr>
<tr>
<td></td>
<td>HIST 342 Lec 1</td>
<td>Full Term</td>
<td>TTh 10:30a → TTh 7.30a</td>
<td>BRNG 2280</td>
<td></td>
</tr>
<tr>
<td>+44.1</td>
<td>POL 101 Lec 3</td>
<td>Full Term</td>
<td>TTh 12:00p → TTh 10:30a</td>
<td>BRNG 2280</td>
<td>+34 (h+2)</td>
</tr>
<tr>
<td></td>
<td>HIST 342 Lec 1</td>
<td>Full Term</td>
<td>TTh 10:30a → TTh 3:00p</td>
<td>BRNG 2280 → BRNG 2290</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OBHR 330 Lec 4</td>
<td>Full Term</td>
<td>TTh 3:00p</td>
<td>BRNG 2290 → LWSN B155</td>
<td></td>
</tr>
</tbody>
</table>

(all 15/1 possibilities up to 3 changes were considered, top 4 of 1/ suggestions displayed)
Timetabling: Problem

Model

• Variable: class
• Value: time and room placement

Hard Constraints

• Room size, sharing, availability
• No instructor / room can have two classes at the same time
• Required or prohibited preferences

Soft Constraint (Objectives)

• Time, room, and distribution preferences
• Student conflicts
• Additional criteria (too big rooms, back-to-back instructors, …)
Using the Solver

I. Make sure the problem has a solution
   • All classes are assigned
   • Using check configuration
   • Conflict-statistics can be used to discover issues
Timetabling: Solver

Using the Solver

1. Make sure the problem has a solution
2. Run the solver to produce a timetable
   - Using **default** configuration
   - It is possible to iterate (if needed), or start the solver from the previous timetable
Timetabling: Solver

Using the Solver

1. Make sure the problem has a solution
2. Run the solver to produce a timetable
3. Once there is a decent timetable
   • Make manual changes, using *interactive* configuration

<table>
<thead>
<tr>
<th>Suggestions</th>
<th>Score</th>
<th>Class</th>
<th>Date</th>
<th>Time</th>
<th>Room</th>
<th>Students</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>15.2</td>
<td>POL 101 Lec 3</td>
<td>Full Term</td>
<td>TTh 12:00p → TTh 7:30a</td>
<td>BRNG 2280</td>
<td>+11</td>
</tr>
<tr>
<td></td>
<td>31.7</td>
<td>POL 101 Lec 3</td>
<td>Full Term</td>
<td>TTh 12:00p → TTh 10:30a</td>
<td>BRNG 2280</td>
<td>+36 (h+3)</td>
</tr>
<tr>
<td></td>
<td>36.6</td>
<td>HIST 342 Lec 1</td>
<td>Full Term</td>
<td>TTh 10:30a → TTh 1:30p</td>
<td>BRNG 2280 → BRNG 2290</td>
<td>+36 (h+4)</td>
</tr>
<tr>
<td></td>
<td>44.1</td>
<td>POL 101 Lec 3</td>
<td>Full Term</td>
<td>TTh 12:00p → TTh 10:30a</td>
<td>BRNG 2280</td>
<td>+34 (h+2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HIST 342 Lec 1</td>
<td>Full Term</td>
<td>TTh 10:30a → TTh 3:30a</td>
<td>BRNG 2280 → BRNG 2290</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>OBHR 330 Lec 4</td>
<td>Full Term</td>
<td>TTh 3:00p</td>
<td>BRNG 2290 → LWSN B155</td>
<td></td>
</tr>
</tbody>
</table>

(all 15/1 possibilities up to 3 changes were considered, top 4 of 17 suggestions displayed)

Solver Configuration: it is possible to tweak solver parameters if needed
(there is a tradeoff between times, rooms, distributions, and student conflicts)
Making changes

1. Minimal Perturbation Mode (MPP)
   - When many changes are needed
   - Fully automated (default configuration with the mode set to MPP)
   - Additional criterion: changes from the initial solution
   - Different weights, e.g., time changes are usually more penalized

2. Once there is a timetable saved, use the interactive configuration
   - Can break some constraints
   - Solver provides suggestions, but does not make any decisions

3. When the timetable is published
   - Changes can be made without loading the data into the solver
Decentralized Timetabling

- Defined by solver groups
  - One or more departments that are to be solved together
- Committed solutions of other problems are used as basis
- Multiple problems can be solved together, manual changes can be made separately

Externally Managed Classes

- For instance, distance learning classes are solved separately
- Different set of rooms
- Timetabled before or after the departmental problems
- Other examples: large lecture rooms, computing labs, need room
Publication

• A committed timetable can be published by changing the status on the academic session
• Instructors and students can see the timetable
• Next steps
  • Export to an external system
  • Student scheduling
  • Examination timetabling
  • Event management
Best practices: Solver

- Multiple problems can be timetabled together
- Multiple solutions can be saved
- It is important to commit a solution when you wish the assignments to show in other problems
- Use distribution preference priority for problems that are solved before or after the departmental problems (see Departments page)
- Use Reload Input Data when there is a change in the inputs

- Use Chameleon if you want to run several solvers at once
- Create several timetables, then choose the best one
Best practices: Solver parameters

- Optimization can usually be achieved by setting up a combination of solver parameters
- Example: Hard conflict weights
- Example: No student conflicts
- Example: Times are way more important than rooms
- Distance conflict settings (student speed, distances between non back-to-back classes, …)
- Automatic distribution constraints
- …
- Try experiment with various solver settings
Best practices: Making Changes

- Use the Interactive solver (from the Timetables page) to be able to break some hard constraints
- MPP penalization can tell the solver what changes are hard
- Do not use the solver when students are already being enrolled, use Class Assignment page instead
Custom Reports

- Written in HQL (Hibernate Query Languages)
- Can have parameters (current session, department, subject area, …)
- Lines can be clickable
- Export to CSV
- Example reports are available in UniTime
- Requires knowledge of the UniTime data model

Point In Time Reports

- Snapshot of current state of students and their registration, class limits, etc.
- Using the Data Exchange page
- Multiple snapshots can be imported
- Full set of reports (weekly class hours, room utilisations, etc.)
- Roll forward

See http://help.unitime.org/Course_Reports for more details.
Scripts

• Using JSR 223: Scripting for the Java Platform
• JavaScript or Python, can call UniTime methods
  • For Python, put Jython Standalone JAR to Tomcat/libs
• Can have parameters (including a file)
• Can produce a file
• Convenient for additional administrative tasks, custom CSV imports and exports, etc.
• Some examples are available at https://goo.gl/ufqW1t
• Permission (users with the given permission can run the script)
• Requires knowledge of the UniTime code base

See http://help.unitime.org/Scripts for more details.
What-If Scenarios

- Use academic session export/import to copy a session to a test instance
- Test session status can be used for multiple copies of the same session
- XMLs exports/imports or Scripts can be used to quickly manipulate the data (there is a new XML for preferences in UniTime 4.2)
- Examples
  - Building or room should become unavailable
  - Change in time patterns
  - Going from semesters to trimesters
  - ...
UniTime

• Comprehensive system
• A lot to configure, customize, or otherwise to do
• But the defaults work well

For more details, please see us at the conference

• UniTime: Best Practices (Sunday, 1:30pm - 4:30pm in Flower)
• Case Study: UniTime at Masaryk University (Monday, Showcase Reception)
• UniTime 4.2: Instructor Scheduling (Tuesday, 10:15am - 11:00am in Flower)
• Course Timetabling Around the World (Tuesday, 2:30pm - 3:15pm in Flower)

• Or visit www.unitime.org

An online demo is available at https://demo.unitime.org