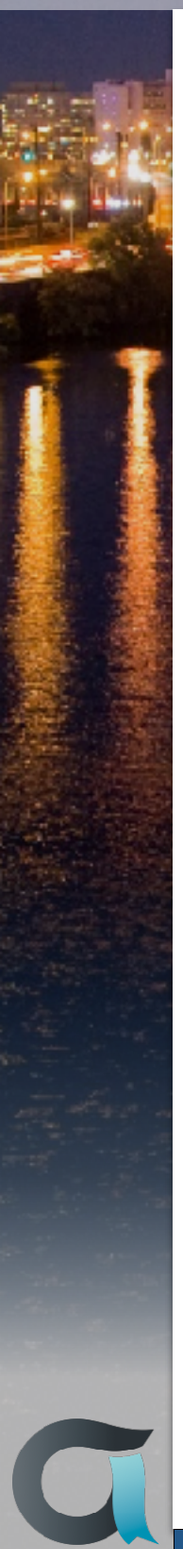
UniTime
Best Practices

UniTime 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

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

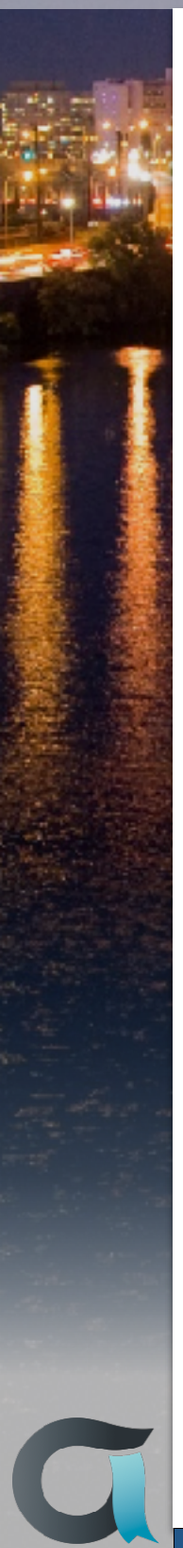
UniTime Demo Instance

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

demo.unitime.org/workshop



User	Department	Courses	Classes	Instructors
20, 26, 48	Art	57	154	43
38, 40	Biology	33	111	41
14, 49	Civics	58	95	21
17, 18, 28, 42	Czech	114	225	32
15, 30, 36	English	157	250	50
1, 22	French	56	81	18
24, 33	Geography	25	43	19
8, 12, 34	German	78	133	20
27, 47	Health Ed	21	39	17
6, 32	History	39	93	49
4, 45	IT	49	95	20
9, 10	Language	23	89	14
23, 25, 29	Mathematics	53	104	27
41, 51	Music	59	196	17
37, 46	Pedagogy	17	76	28
2, 7, 31, 35, 43	Physics	170	416	84
5, 19	Prime Ped	34	99	16
16	Psychology	40	109	14
21, 39	Physical Ed	24	64	16
11, 50	Russian	83	156	18
13	Social Ed	89	136	75
3, 44	Special Ed	135	231	74

Username:
user001

Password:
pwd001

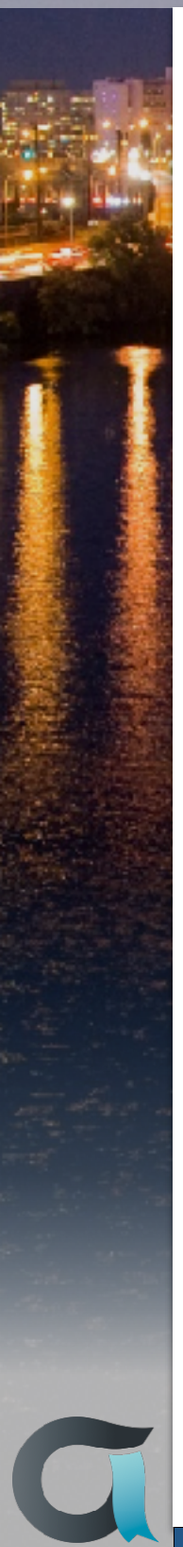
•
•
•

Username:
user051

Password:
pwd051

Administration

UniTime Setup



Installation

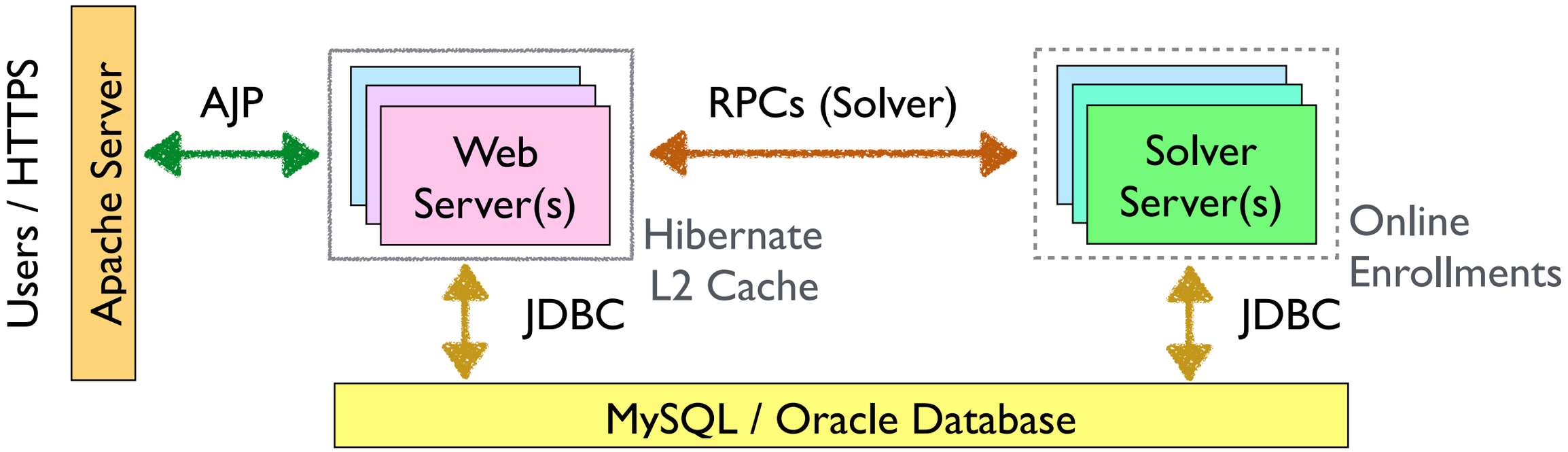
- UniTime can be downloaded from http://builds.unitime.org
- Installation Instructions: help.unitime.org/Timetabling_Installation
  - *See Windows / Linux specific notes at the bottom of the page*
- Hardware Requirement
  - Any system capable of running Java and MySQL/Oracle
  - Linux is recommended, 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!

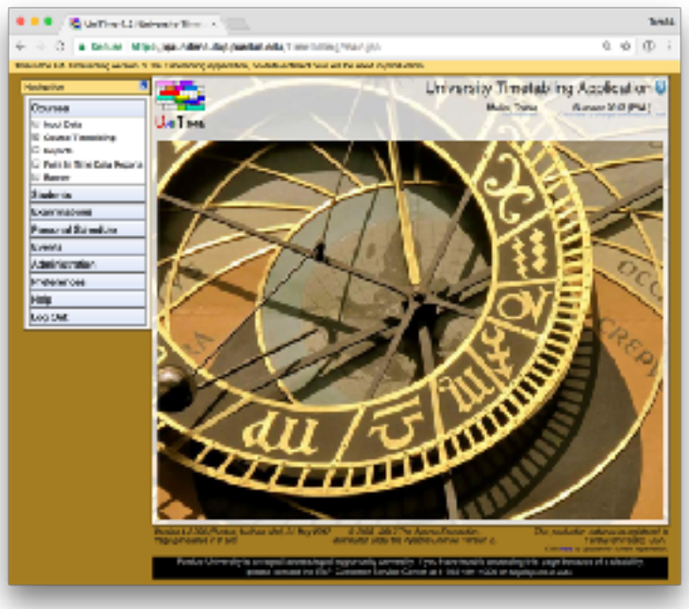
UniTime Setup

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)



UniTime Setup



Customization

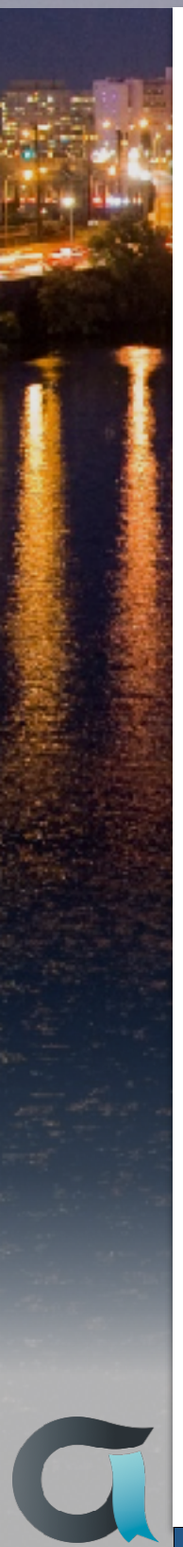
- Custom properties
  - Application Configuration page
  - Custom properties file
- Custom CSS, welcome message, disclaimer, menu content & style
- Much more, see the Application Configuration page for the list

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.

UniTime Setup



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.

See https://demo.unitime.org/UniTime?locale=cs for UniTime in Czech.

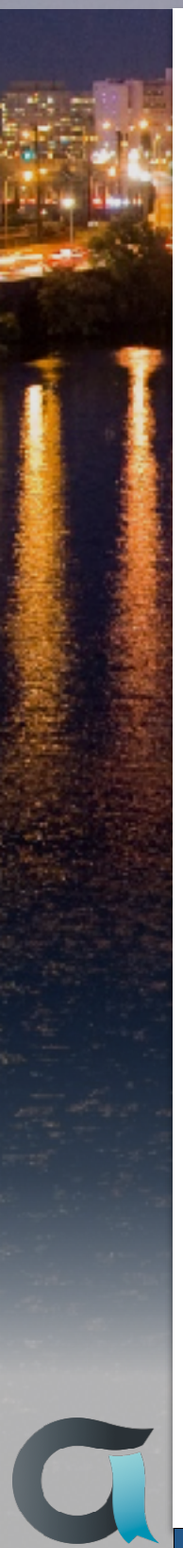
UniTime Setup

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.

UniTime Setup



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.

UniTime Setup

Date Patterns

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

March 2015

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
10	1	2	3	4	5	6	7
11	8	9	10	11	12	13	14
12	15	16	17	18	19	20	21
13	22	23	24	25	26	27	28
14	29	30	31				

April 2015

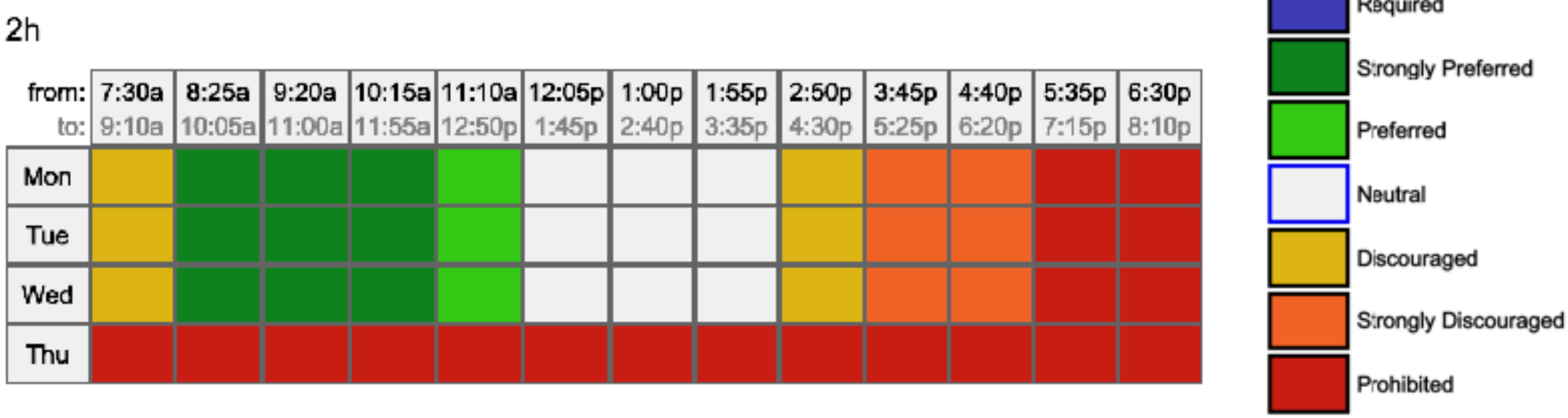
	Sun	Mon	Tue	Wed	Thu	Fri	Sat
14				1	2	3	4
15	5	6	7	8	9	10	11
16	12	13	14	15	16	17	18
17	19	20	21	22	23	24	25
18	26	27	28	29	30		

May 2015

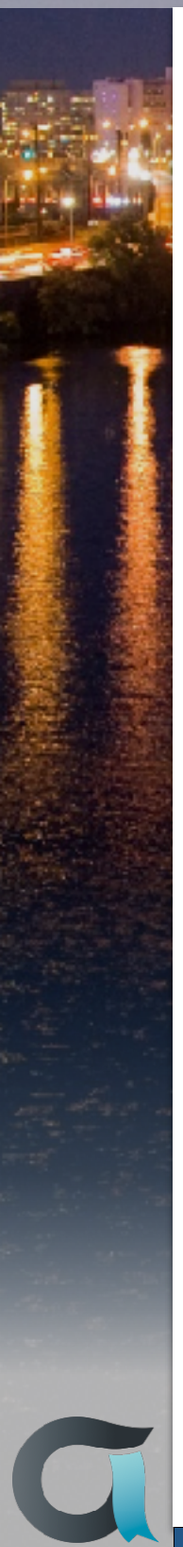
	Sun	Mon	Tue	Wed	Thu	Fri	Sat
18						1	2
19	3	4	5	6	7	8	9
20	10	11	12	13	14	15	16
21	17	18	19	20	21	22	23
22	24	25	26	27	28	29	30
23	31						

Time Patterns

- Possible time slots within a week



UniTime Setup



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

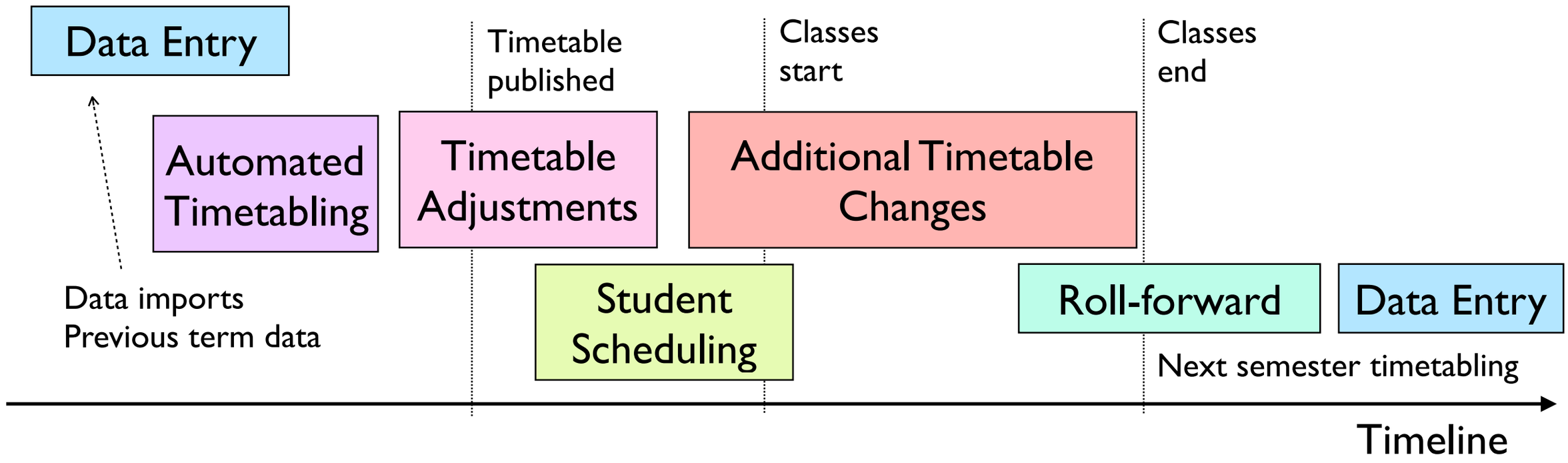
See http://www.unitime.org/uct_interfaces.php for the list of XML interfaces.

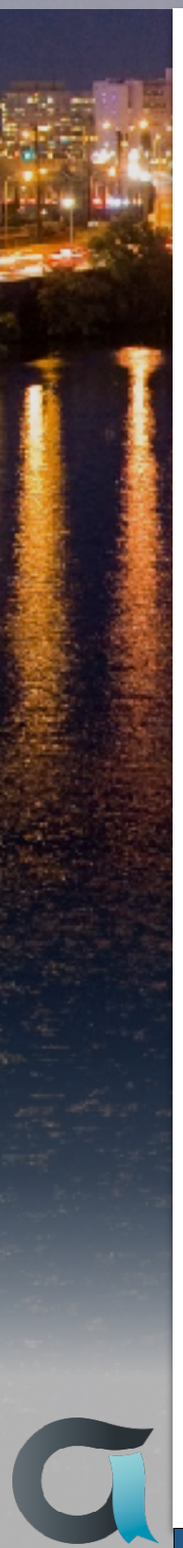
See https://goo.gl/L1sEVN for UniTime 4.2 APIs.

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





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)

Data Entry

Data Entry: Rooms

Rooms

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

K 73

Workdays × Daytime

from: / to:	7:30a 8:00a	8:00a 8:30a	8:30a 9:00a	9:00a 9:30a	9:30a 10:00a	10:00a 10:30a	10:30a 11:00a	11:00a 11:30a	11:30a 12:00p	12:00p 12:30p	12:30p 1:00p	1:00p 1:30p	1:30p 2:00p	2:00p 2:30p	2:30p 3:00p	3:00p 3:30p	3:30p 4:00p	4:00p 4:30p	4:30p 5:00p	5:00p 5:30p	5:30p 6:00p	6:00p 6:30p
Mon																						
Tue	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL
Wed	CIVC	CIVC	CIVC	CIVC	CIVC	CIVC	CIVC	CIVC	CIVC													
Thu	CIVC	CIVC	CIVC	CIVC	CIVC	CIVC	CIVC	CIVC	CIVC													
Fri	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

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

	A 50	D 20	K 73	140A	JAMU	
A 50		5	5	19	22	A 50
D 20	5		0	17	20	D 20
K 73	5	0		17	22	K 73
140A	19	17	17		10	140A
JAMU	22	20	22	10		JAMU
	A 50	D 20	K 73	140A	JAMU	

Data Entry: Rooms

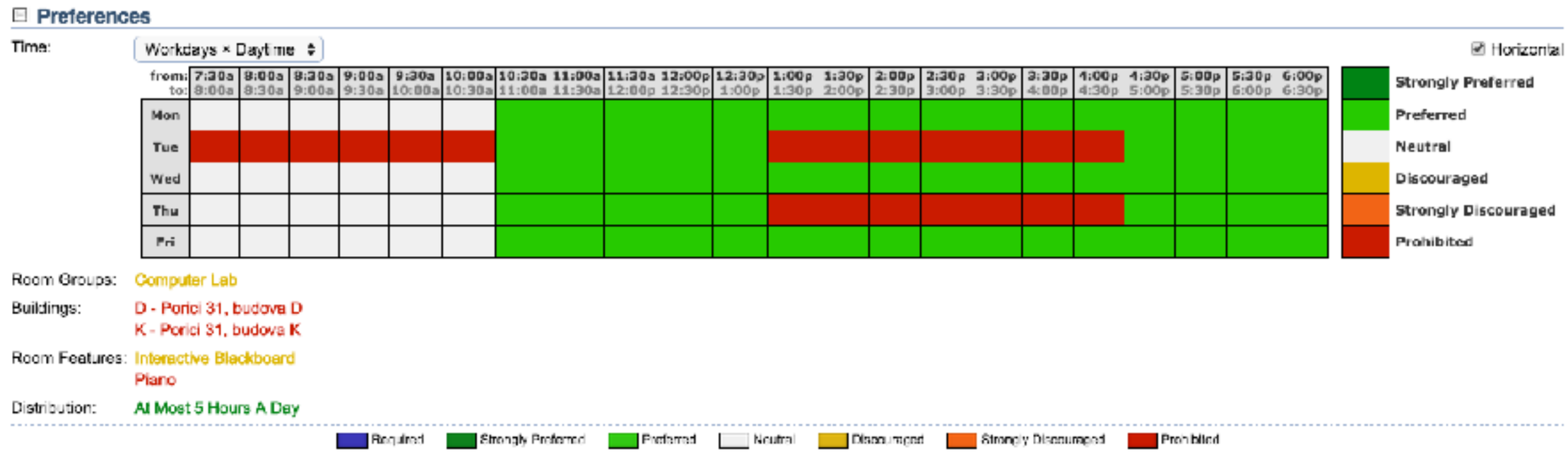
Best Practices: Rooms

- 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)

Data Entry: Instructors

Instructors

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



Data Entry: Instructors

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.

Data Entry: Courses

Instructional Offering

	Limit	Date Pattern	Minutes Per Week	Time Pattern	Time	----Preferences---- Room	Distribution	Instructor
MA 170	40	**Statistics I**						
STAT 170		Introductory statistics						
Lecture	40	Full Term	50	1 x 50		Classroom		
Laboratory	40	Full Term	150	3 x 50		EDUC CompPr	Same Room	
Lec 1	40	Full Term	50	1 x 50		ThtrSeat Classroom		G. Newman
Lab 1	20	Full Term	150	3 x 50		EDUC CompPr	Same Room	J. Smith
Lab 2	20	Full Term	150	3 x 50		EDUC CompPr	Same Room	J. Smith

Data Entry: Courses

Instructional Offering

Course Offerings

	Limit	Date Pattern	Minutes Per Week	Time Pattern	Time	----Preferences---- Room	Distribution	Instructor
MA 170 STAT 170	40	**Statistics I** Introductory statistics						
Lecture	40	Full Term	50	1 x 50		Classroom		
Laboratory	40	Full Term	150	3 x 50		EDUC CompPr	Same Room	
Lec 1	40	Full Term	50	1 x 50		ThtrSeat Classroom		G. Newman
Lab 1	20	Full Term	150	3 x 50		EDUC CompPr	Same Room	J. Smith
Lab 2	20	Full Term	150	3 x 50		EDUC CompPr	Same Room	J. Smith

Data Entry: Courses

Instructional Offering

Course Offerings

Scheduling Subparts

	Limit	Date Pattern	Minutes Per Week	Time Pattern	Time	Room	Distribution	Instructor
						----Preferences----		
MA 170 STAT 170	40	**Statistics I** Introductory statistics						
Lecture	40	Full Term	50	1 x 50		Classroom		
Laboratory	40	Full Term	150	3 x 50		EDUC CompPr	Same Room	
Lec 1	40	Full Term	50	1 x 50		ThtrSeat Classroom		G. Newman
Lab 1	20	Full Term	150	3 x 50		EDUC CompPr	Same Room	J. Smith
Lab 2	20	Full Term	150	3 x 50		EDUC CompPr	Same Room	J. Smith

Data Entry: Courses

Instructional Offering

Course Offerings

Scheduling Subparts

Classes

	Limit	Date Pattern	Minutes Per Week	Time Pattern	Time	----Preferences---- Room	Distribution	Instructor
MA 170	40	**Statistics I**						
STAT 170		Introductory statistics						
Lecture	40	Full Term	50	1 x 50		Classroom		
Laboratory	40	Full Term	150	3 x 50		EDUC CompPr	Same Room	
Lec 1	40	Full Term	50	1 x 50		ThtrSeat Classroom		G. Newman
Lab 1	20	Full Term	150	3 x 50		EDUC CompPr	Same Room	J. Smith
Lab 2	20	Full Term	150	3 x 50		EDUC CompPr	Same Room	J. Smith

Data Entry: Preferences

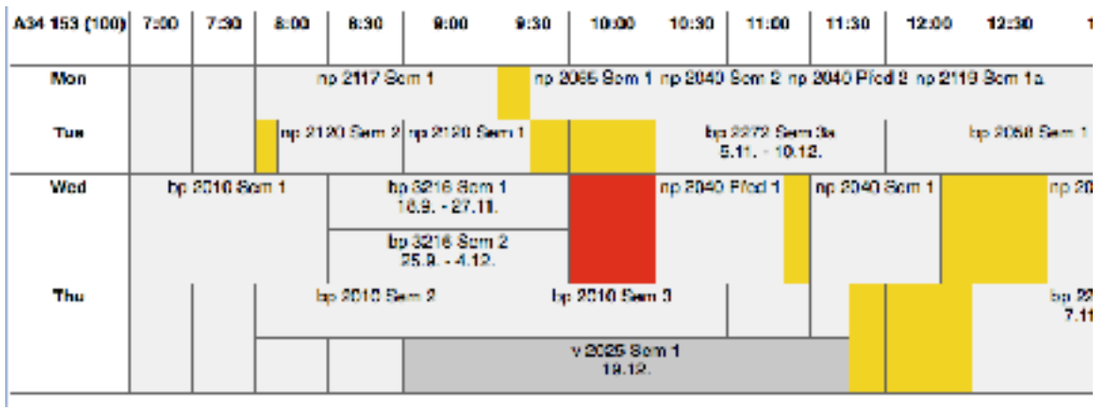
Best Practices: Courses

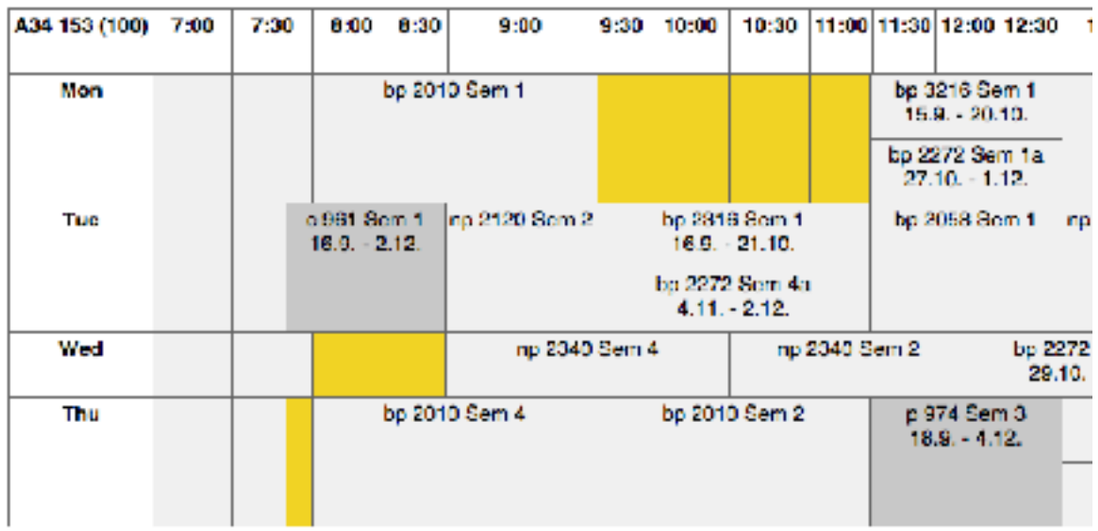
- There can be multiple configurations
  (with different instructional method, e.g., traditional x online)
- If a class does not follow a standard time pattern, it could be split
- Reservations can be used to direct students to the appropriate configurations / classes
- Use cross-lists whenever a course 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

Data Entry: Preferences

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





Data Entry: Preferences

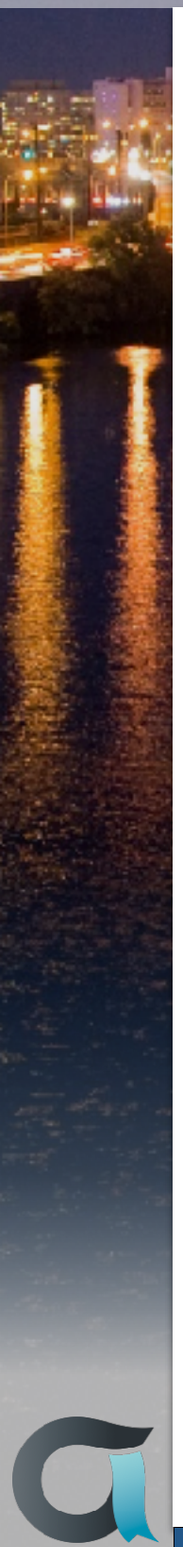
Best Practices: Preferences

- 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

unitime.preferences.highlightClassPrefs

	Limit	Date Pattern	Minutes Per Week	Time Pattern	Time	Room	Distribution	Instructor
						----Preferences----		
MA 170 Statistics I STAT 170 Introductory statistics	40							
Lecture	40	Full Term	50	1 x 50		Classroom		
Laboratory	40	Full Term	150	3 x 50		EDUC CompPr	Same Room	
Lec 1	40	Full Term	50	1 x 50		ThtrSeat Classroom		G. Newman
Lab 1	20	Full Term	150	3 x 50		EDUC CompPr	Same Room	J. Smith
Lab 2	20	Full Term	150	3 x 50		EDUC CompPr	Same Room	J. Smith

Data Entry: Input Data



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

Solver

Timetabling: Solver

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

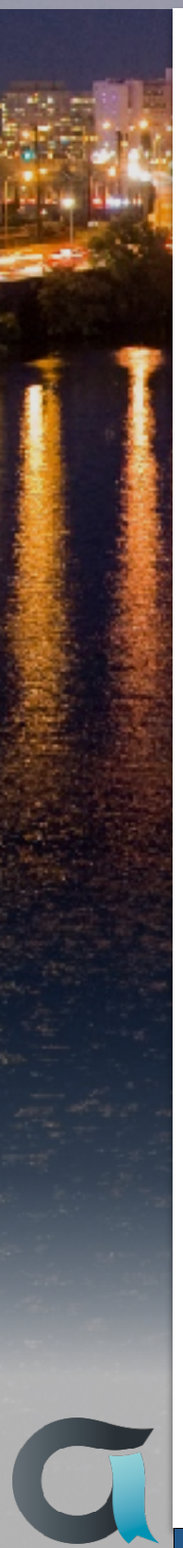
Suggestions

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

(all 1571 possibilities up to 3 changes were considered, top 4 of 17 suggestions displayed)

Search Deeper

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, …)

Timetabling: Solver

Using the Solver

1. Make sure the problem has a solution
   - All classes are assigned
   - Using **check** configuration
   - Conflict-statists 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

Type	Course Timetabling Solver
Solver	Solving problem ...
Phase	Improving found solution ...
Progress	5 of 100 (5.0%)
Owner	A. Root as ART & BIOL & CIVC & CZ & ENG & FRN &...
Host	local
Session	Spring 2015 (ED)
Version	4.0.16
Assigned variables	100.00% (1613/1613)
Overall solution value	-17554.24
Time preferences	91.26% (-36722.00)
Student conflicts	807 [committed:0, distance:1, hard:177]
Room preferences	93.31% (-1385)
Distribution preferences	96.37% (-525.00)
Back-to-back instructor preferences	99.98% (1)
Too big rooms	19.84% (1280)
Useless half-hours	0.63% (0 + 1316)
Same subpart balancing penalty	36.58
Room Size Penalty	17.36
Perturbation variables	9.60% (154 + 8)
Perturbations: Total penalty	330.10
Time	0.06 min
Iteration	1940
Memory usage	1791.38M
Speed	520.45 /s
Block Constraints	100% (0)
Important student conflicts	495 [hard: 34]

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

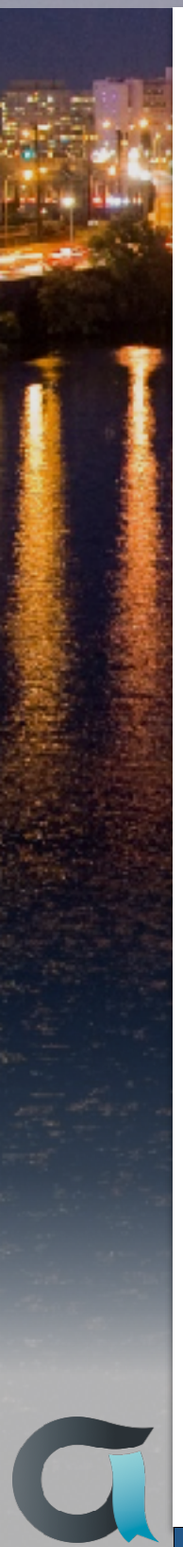
Suggestions

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

(all 1571 possibilities up to 3 changes were considered, top 4 of 17 suggestions displayed) Search Deeper

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

Timetabling: Making Changes



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

Timetabling: Cooperation

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

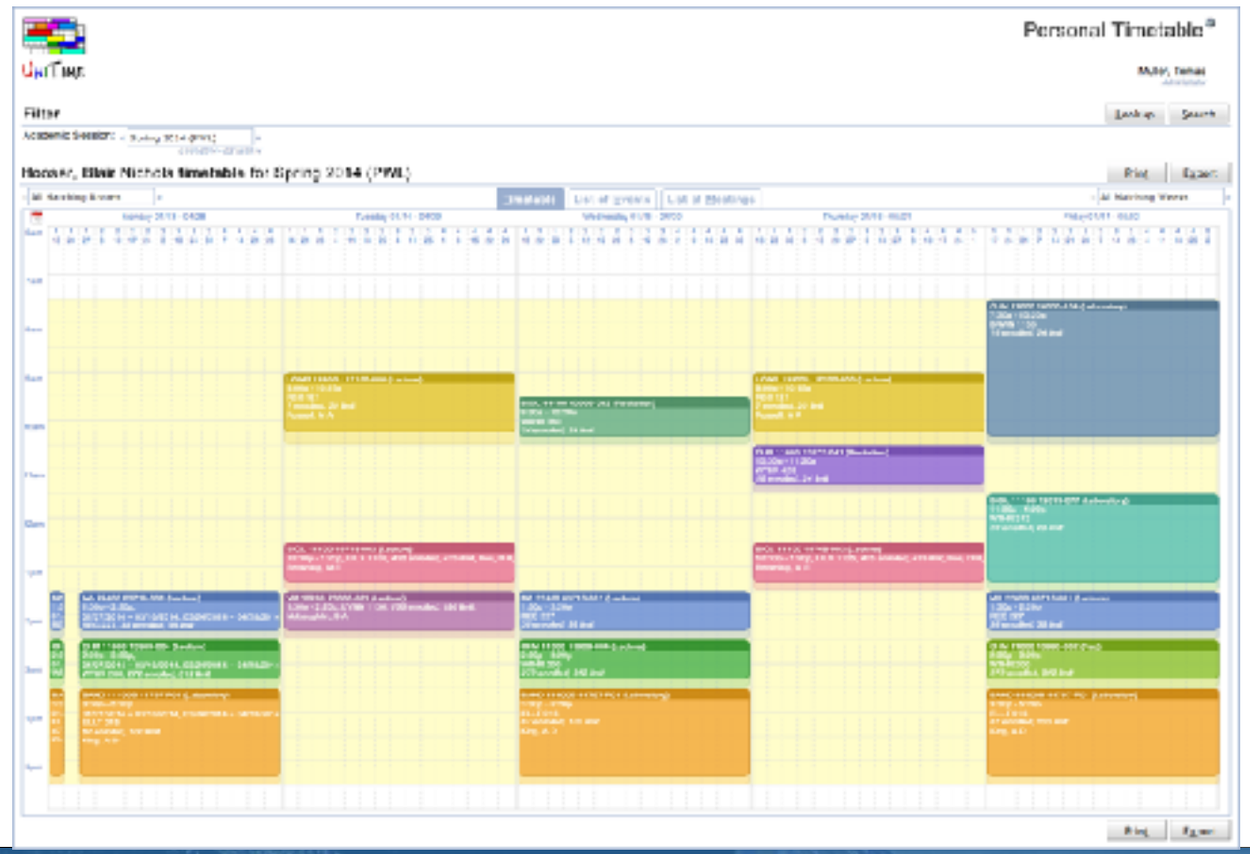
Externally Managed Classes

- 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

Timetabling: Publication

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



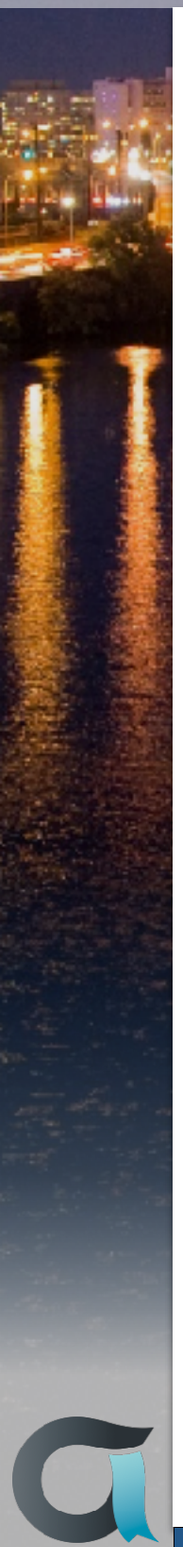
Best Practices: Timetabling

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

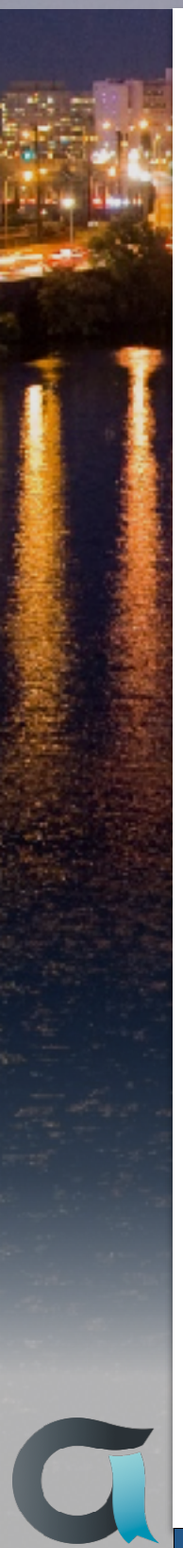
Best Practices: Timetabling



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: Timetabling



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

Other Features

Other: Reporting

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

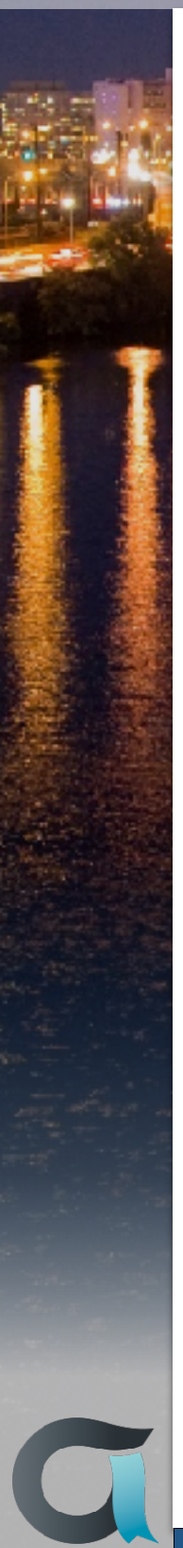
Point In Time Reports

New in UniTime 4.2

- 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 utilizations, etc.)
- Roll forward

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

Other: Scripts



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.

Other: What-If Scenarios

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
  - …

Conclusion

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