A Generic Approach to Interactive University Timetabling

Michael Zeising, Stefan Jablonski
Chair for Applied Computer Science IV
University of Bayreuth
Bayreuth, Germany
{michael.zeising, stefan.jablonski}@uni-bayreuth.de

Abstract—**Room utilization is a problem for universities most notably when students do not have to enroll for the courses they want to attend. In this case, the only information available is the examination regulation of their respective course of study. In this paper, it will be illustrated how student constraints can be inferred from these regulations and how they can be used to reduce the task to a well understood problem. Furthermore, an innovative user interface based on these constraints which enables highly interactive university course scheduling will be presented. In order to support a wide range of environments, the approach rests upon a very general domain model and does not depend on a specific solver technology.**

Keywords-interactive timetabling, decision support systems, examination regulations, planning user interfaces

I. Introduction

Universities are forced to optimize their room utilization while satisfying time constraints for students as well as for teachers. The task becomes even more complex if the university decides to not force their students to enroll for courses before attending them. The only information from which time constraints for the students may then be derived is the curriculum to which they committed themselves. The actual scheduling problem is well understood and numerous mathematical approaches have been published so far [1-3]. In the scope of these approaches, a curriculum is understood as set of courses that a student must attend [4]. However, for German universities this abstraction is oversimplified. There are many paths through a course of study and only a few courses could be identified as a mandatory core. The possible paths that lead to a degree are constrained by often complex regulations.

This paper demonstrates how curriculum-based scheduling can be significantly improved by incorporating these regulations into the planning problem. Furthermore, it will be shown how an innovative user interface allows for an interactive scheduling. By the use of simple technology-agnostic models, the approach imposes only very few restrictions on the surrounding landscape and can therefore be easily integrated into existing management solutions. The suggested approach is also not bound to a specific optimization technique.

II. Problem Description and Requirements

A university maintains hundreds to thousands of rooms shared for teaching activities. To make the best possible use of these rooms, they are managed in a centralist manner by at least one room authority. A room authority is responsible for positioning a teaching activity in time and space so that no hard constraints and as few as possible soft constraints are violated. An authority, however, is not permitted to alter an activity's properties like its duration. It is argued that this planning procedure must be of interactive nature. So the challenge consists in providing the best possible support for human planners at the user interface level.

The actual constraints may vary between universities and therefore must be specified individually. It is important for the solution to be independent of the respectively used optimization technology. So the second challenge is to make sure that the solution exposes a generic and yet efficient interface to solver engines and allows for as many kinds of constraints as possible to be implemented.

The most important hard constraints usually encompass persons and rooms being impartible. While the violation of most of them may be easily discovered, there is one very complex constraint: time overlaps for students. In this context, a university offers courses of study that lead to certain degrees. The structures and rules of such a course of study are specified by regulations (also called examination regulations). For easier naming, it is assumed that *one* regulation describing one course of study leading to one degree for a student. Regulations are usually subject to change so they are versioned but it is expected that one student is always registered for only one regulation (version) at a time. The third task is to exploit the information on students for inferring potential timetabling conflicts.

III. Contribution

Within this paper, insight into how university course timetabling can be improved by interpreting basic regulation rules will be delivered. A generic approach to an interactive regulation-based planning tool will be introduced. By the term "generic" it is meant that the models used are simple and generally applicable to universities. "Interactive" means that the goal is a decision support system for planning authorities. By means of an innovative user interface the solution will support the authorities with the intelligence of a

 ISBN: 978-1-61208-177-9

semi-automatic optimization. The tool presented will not depend on a specific optimization technology.

In Section IV, the domain model of the application for adequately capturing regulations, the teaching offer and planning information will be introduced. In Section V, it will be demonstrated how these information can be used to infer conflicts for the students. In Section VI, a model used for communicating with the solver for automatic optimization will be proposed. Finally, in Section VII, the planning process supported by the approach and a user interface for interactive planning will be presented.

IV. DOMAIN MODEL

The essential information on which the approach is based have to be adequately modeled. In the following, a brief overview of this data model will be given.

A. Regulations and Teaching Offer

The first task consists in modeling regulations. The authors have gained the experience that they may be modeled as trees of teaching units where the leaves are courses and the inner nodes may declare constraints on their children. The left side of Figure 1 shows that a regulation consists of units, which can be "phases", "areas", "sections", "stages" and so on. Everything that is used to structure courses of study is considered a unit. The module is a special kind of unit which holds a number of credit points gained according to the European Credit Transfer System (ECTS) if the module is passed [5]. A module may contain courses, which again are special units. The course holds an amount of workload in hours per week which is interpreted as the time of attendance for students and teachers. It is assumed that a course always lasts for one semester which, up to the knowledge of the authors, holds for every German university. If a course spanned more than one semester it would be split up into several continuously numbered parts like "Basic Mathematics I" and "II" and offered in different semesters.

The model allows for two very simple rules: prerequisites for a unit and choices. Each unit may reference others that must be attended before the unit because e.g., it requires knowledge gained in the other units. Generally, the hierarchy of units is interpreted as connected neither by "and" nor by "or" operators. That means that for a unit containing two or more others *some* of these child units are expected to be attended. For modeling explicit choices instead, the `nOutOf`-attribute may be set, which means that exactly *n* out of all child units need to be attended by a student.

It is assumed that every student is registered for exactly one regulation. Students registered for multiple regulations (and therefor enrolled in multiple courses of study) at the same time do not raise a problem but they cannot be supported. However, this should be an acceptable restriction as studying multiple independent courses of study is a rare case.

The actual teaching offer per semester is modeled by the right half of Figure 1. It states that a course mentioned in a regulation may be offered in a semester. A course offer in turn consists of at least one session which involves at least one person usually in the role of a teacher.

This model is, at the best of the authors' knowledge and experience, in line with the respective models of major commercial university management solutions. Therefore, the approach can be simply integrated with these systems.

B. Time, Space and Planning

There is only one valid planning at a time. A planning consists of allocations while an allocation assigns a session to a certain room and a certain time. The solution is not bound to a specific optimization model and time granularity.

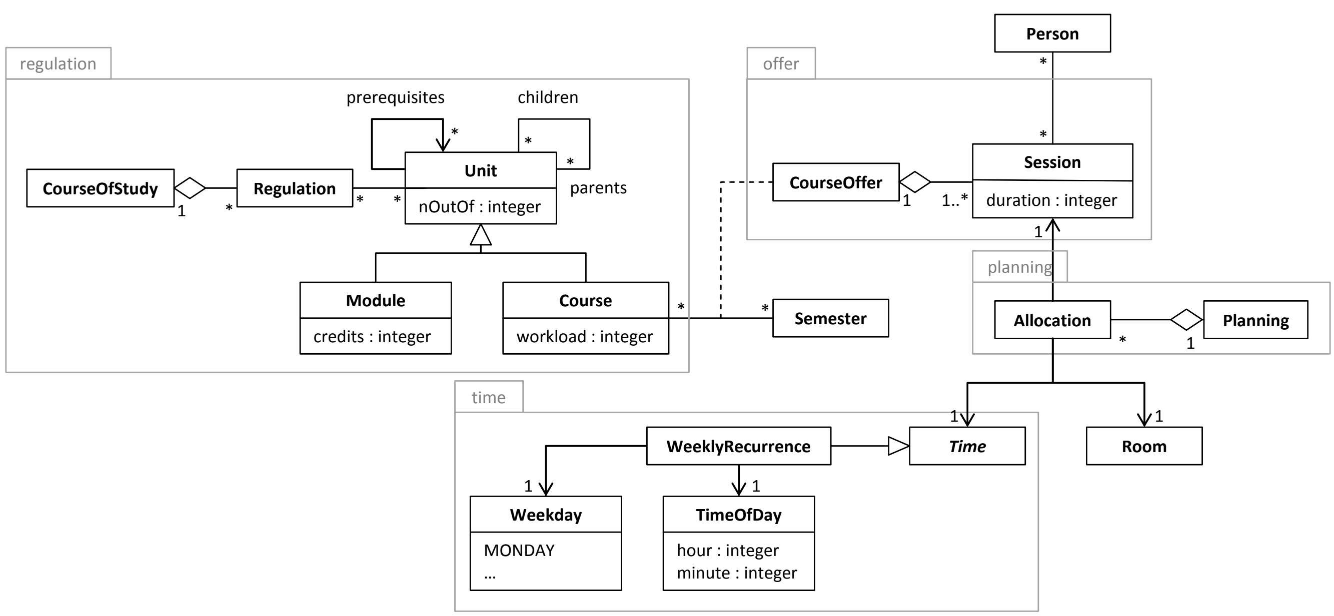

Figure 1. Domain model for capturing regulations, teaching offers and plannings

As a result, time is represented absolutely and technology-agnostic within the data model. Time may be a single event (not shown) or a recurrence rule. In the most frequent case, sessions will occur weekly. Simple weekly recurrence rules may be modeled as shown in the lower half of Figure 1. There is an event with a certain duration recurring every certain weekday at a certain time of day. Of course, these rules quickly become complex as an event may only occur every second or third week and there may also be gaps within the rule but we concentrate on the simple case for now.

V. Deriving Session Conflicts

A simple perception of collisions and conflicts is used: two sessions "conflict" if they must not overlap in time. If two sessions conflict and overlap, they "collide" and render a timetable either unfeasible or worse than another.

There are the usual trivial cases of two sessions conflicting that most approaches support and that may easily be determined. Therefor two sessions are conflicting if

- they take place in the same room or
- they involve the same teacher.

Determining whether two sessions are conflicting from the students' point of view, however, is nontrivial and, up to the authors' knowledge, not supported by any existing solution (if not provided to the system *a priori*). From the students' point of view, two sessions are potentially conflicting if

- they are offered in the same semester and
- there is a regulation containing both of them and
- there is at least one "path" allowing for attending both of them.

In short, they conflict if there is potentially at least one student attending both of them while following his or her regulation.

For illustrating the principles of deriving conflicts from a regulation, imagine two exemplary regulations shown in Figure 2.

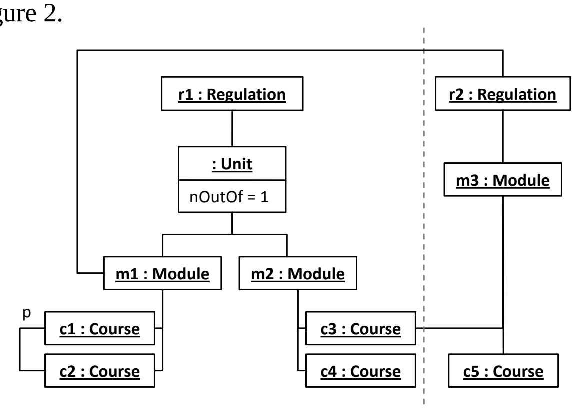

Figure2. Object model for two exemplary regulations

Regulation r_1 consists of a section containing two modules from which one has to be chosen. Each of the modules contains two courses. Course c_2 has c_1 as a prerequisite which means that c_1 has to be attended before c_2. Regulation r_2 reuses module m_1 and contains another module m_3. Module m_3 reuses course c_3 and contains a further course c_5. This degree of reuse between regulations is typical for modular interdisciplinary courses of study at the university. Figure 2 shows the resulting object graph for the example in the form of an object diagram.

A. Semester-specific Representation of Regulations

Now, assume that during the considered semester there is one session s_i offered for each course c_i. For being able to operate directly on sessions, the object graph depicted in Figure 2 is reduced to a specialized representation of the regulations shown in Figure 3. This representation may be efficiently computed when using adequate representations of hierarchical data like *nested sets* [6] on the storage level.

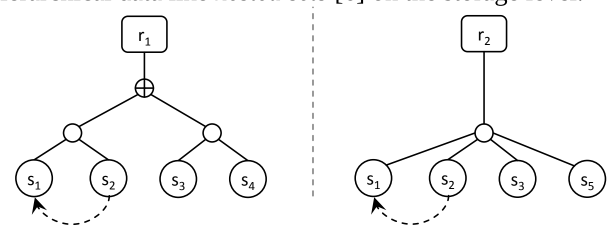

Figure 3. Alternative representation of the exemplary regulations

This form reveals the interpretation of the regulations. For regulation r_1 the two paths are connected by an *exclusive or* while for r_2 all sessions are just grouped. The empty node is interpreted neither as *or* nor as *and* because this part of the regulation is considered "under-specified" meaning that potentially all of the sessions are attended by the same student.

B. Local Conflict Handling

Initially, it is assumed that all sessions within a regulation are conflicting. This results in the two conflict graphs in Figure 4**a** for the two regulations.

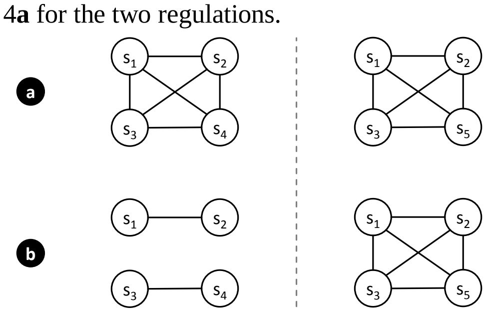

Figure 4. Local conflict graphs resulting from the regulations

Other student-aware approaches would be finished at this point. But now the *exclusive or*-connector between the two session groups in regulation r_1 is interpreted. It can be safely assumed that if a student has to choose between the two directions there will never be a student (registered for this regulation) attending both of them. So the according conflict edges may be removed which leads to the graphs shown in Figure 4**b**. At this point, it becomes clear why this step has to be performed regulation-locally. There could be another regulation where the modules or courses are reused and no exclusive choice has to be made between them. Actually,

 ISBN: 978-1-61208-177-9

regulation r_2 also contains s_1, s_2 and s_3 but without a choice so the conflicts must not be removed there. The meaning of the two graphs in Figure 4**b** now reads as follows:

- there may be at least one student enrolled in r_1 who attends s_1 and s_2 in this semester and at least one who attends s_3 and s_4 and
- there may be at least one student enrolled in r_2 who attends s_1, s_2, s_3 and s_5 in the respective semester.

C. *Global Conflict Handling*

The individual conflict graphs of the regulations can now be merged to one graph for further steps. The set of vertices of the global graph is the union of the local sets. It contains an edge if any of the local graphs contains it. The result for our example is shown in Figure 5**a**. Note that some of the edges removed due to the *exclusive or* are restored after the merge.

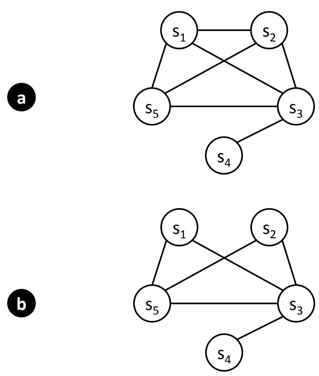

Figure 5. Global conflict graph resulting from the regulations

Further edges may be removed by interpreting the prerequisites of a unit. Figure 1 shows that a unit may declare other units as its prerequisites which means that these units must be attended by a student before the unit. This is a global relation which does not depend on regulations and may not be altered by them. As it is assumed that units span at least one semester, it may be inferred that if a unit is a prerequisite of another one then there will never be a student attending both of them in the same semester and there is no conflict between them. In the example, course c_1 is a prerequisite for c_2. This has been projected to s_1 and s_2 so the conflict between these two sessions is removed leading to the final conflict graph shown in Figure 5**b**.

Please note that by interpreting only very simple properties of the regulations, the number of conflicts has been reduced by almost 40%. Due to fewer conflicts, the optimization procedure now has more degrees of freedom and a better planning becomes possible.

VI. INTEGRATING AN OPTIMIZATION ENGINE

After the potentially conflicting sessions have been identified, the task can be reduced to a general optimization problem. This general problem known as *university course timetabling* may now be solved by one of the numerous approaches mentioned before [7]. For the solution to be independent of the chosen technology, the information necessary for optimization is translated to a planning model. The model constitutes a (not necessarily feasible) solution of the problem and contains information needed for implementing constrains like, e.g., [4]:

- *impartible teachers*: a person may only be involved into one session at a time
- *impartible rooms*: there must be only one session taking place in a room at a time
- *room capacity*: the expected audience of a session must not exceed a room's seating capacity
- *timetable compactness*: there should be no time gaps within a planning
- *room stability*: when assigning a session to another room, the distance to the original one should be minimal
- *session conflicts*: conflicting sessions must not overlap in time

Figure 6 shows the core of the planning model. The underlined properties of an allocation (`room` and `slot`) are to be changed by the solver during optimization.

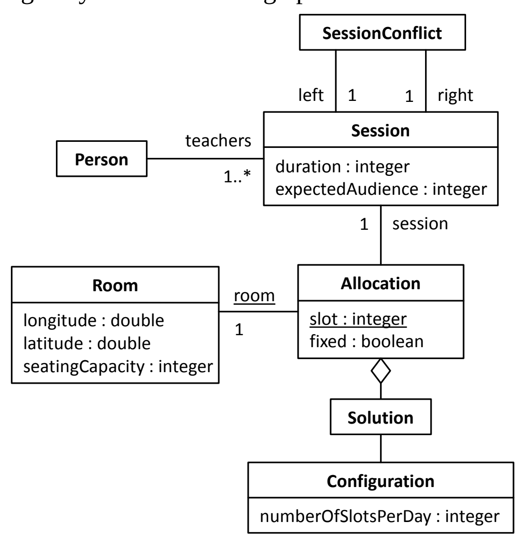

Figure 6. Planning model for communicating with a solver

For the current prototype, a solver implemented on the basis of a linear programming method is employed but it is important to note that the solution is not bound to any particular optimization algorithm or engine. The solver backend is completely exchangeable by design. This is a crucial point because high performance solvers for large real-world problems and the accompanying hardware usually impose significant costs. With an independent solution the decision for an appropriate licensing model can be made by the university individually and even custom developed solutions may be easily integrated.

VII. SUPPORTING INTERACTIVE PLANNING

It has been illustrated how to infer conflicts for students between teaching sessions. In the following, it will be shown how these information serve as the basis for a "student dimension" and enable a comprehensive planning user interface.

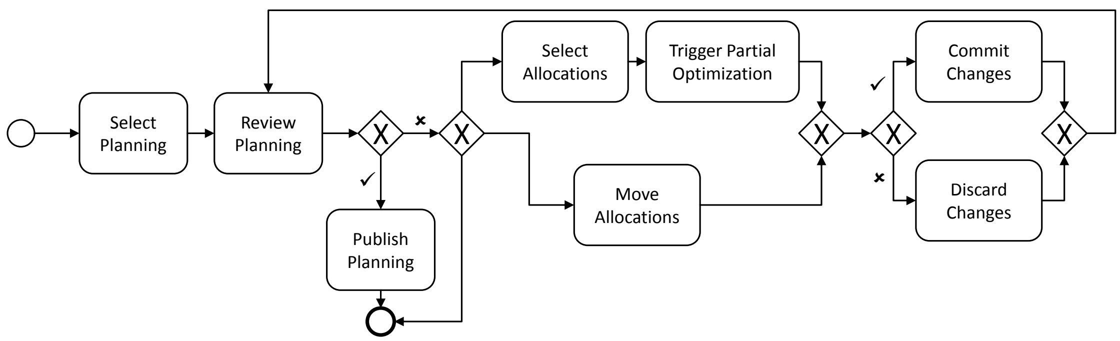

Figure 7. Planning process supported by the user interface

Course scheduling is a highly political topic not only in Germany [8] involving a lot more aspects than the ones currently modeled in university management software [9]. In [3] the authors actually complain that "practical course timetabling is 10% graph theory, and 90% politics" [3]. Therefore, we argue that course scheduling must be a semi-automatic process that can only be successful when combining the users' knowledge with a decision support system. Figure 7 shows a BPMN representation of the planning process skeleton as our interface supports it [10].

First of all, a planning is selected for editing. This may be an already valid and published planning or an intermediate sketch. The step that surely makes the highest demands on the user interface is the review of the edited planning. The human planning authority needs to capture information on all the eventual collisions like:

- a room is occupied multiple times
- a room's capacity is exceeded
- sessions involving the same teacher(s) overlap
- sessions potentially attended by the same students overlap

In order to support these aspects, we have designed an innovative resource-time-view. It arranges session allocations on a resource-time-grid. Allocations are colored according to whether they do not, potentially or certainly collide with others. Colliding sets can be quickly identified by hovering over them. The view is implemented as a JavaServer Faces component so that it can be used within a Java EE environment, a standard platform for business applications. In addition to that, it is fully amenable to the top three recent browser technologies. Figure 8 shows the detail of a screenshot taken from the view in action.

If the planning turns out to be feasible it may be published and made the valid planning, if not the process may just be aborted and deferred or the semi-automatic editing phase may be entered. For manual editing, the planner may move allocations directly in time and space per drag and drop. Of course, this manual planning is only intended for small easily manageable changes of the schedule. If greater numbers of sessions should be moved the planner may resort to the automatic planning of partial timetables. For this to accomplish, he/she selects the sessions to be moved by clicking on them and triggers the optimization. The sessions are moved and the resulting difference is visualized by arrows and can be applied to the edited planning afterwards.

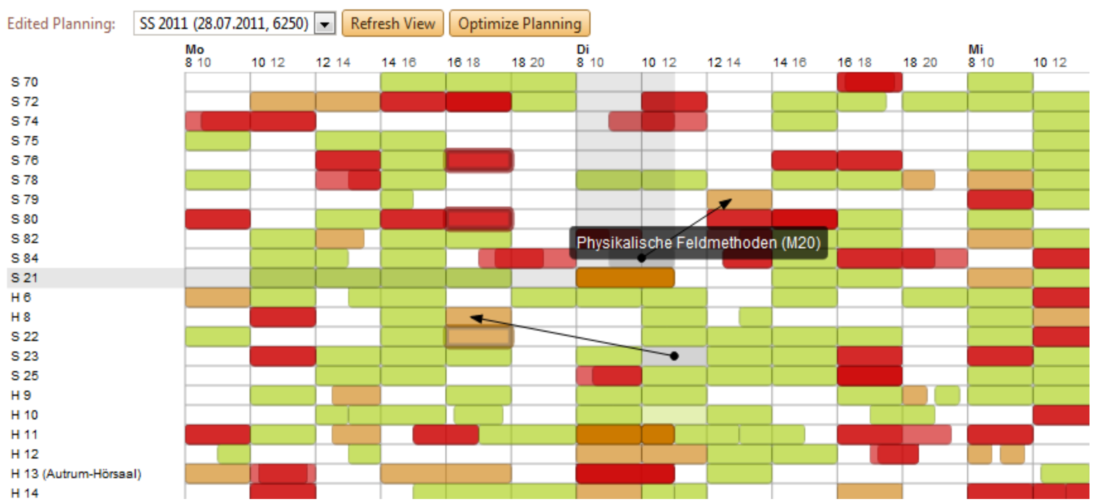

Figure 8. Screenshot of the interactive planning view

VIII. EVALUATION

A prototype of the solution has been integrated with the system landscape of the University of Bayreuth. Regulations are provided by the *FlexNow* examination management system [11]. Organizational and facility information as well as course information are provided by the *HIS LSF* system [12]. The prototype is currently being evaluated by the central room planning bureau of the university. The next step consists in making the results available to other universities.

IX. RELATED WORK

Most research effort is spent on the autonomous optimization of generally acknowledged “standard” problems like, e.g., *post enrolment based course timetabling*, *curriculum based course timetabling* or *examination timetabling* [4]. The assumption is that real world problems of universities may all be reduced to one of these models. However, to the best of the authors’ knowledge, there is no approach that examines whether the optimization can be simplified or improved if further knowledge like regulations is provided.

There are only very few approaches to interactive university timetabling. In [13], Piechowiak and Kolski present an interactive system supporting a resource-time-view adaptable to the needs of different kinds of users (namely the “designer”, the “analyzer” and the “consultor”). Like others [14], the approach is based on a model that relates students by fixed groups. This grouping must be made available to the system *a priori* in the form of syllabi which are usually not available in universities. Besides this, the system is bound to a specific solver technology.

Though the solution presented in [15] supports interactive timetabling, it is based on an inappropriate school model of students arranged in classes. Moreover, it does not provide a comprehensive view on time and the resources involved. It is also bound to a specific solver implementation.

X. CONCLUSION AND FUTURE WORK

In this paper, it has been demonstrated how information gained from examination regulations facilitate comprehensive interactive university course timetabling. It has been explained how to model these regulations in a way so that the information can be gathered from adjacent university management solutions. By the use of a lifelike example, it has been showed how to prepare the information for a generally acknowledged and well examined timetabling problem. The obtained constraints were used to integrate an optimization engine in a technology-agnostic way and to enable a highly productive user interface supporting human planners.

The authors are currently investigating how further rules within regulations can be employed for reducing student-related conflicts. They also concentrate on the weighting of these conflicts based on the expected number of students affected. Furthermore, the solution is currently extended on the planning of non-weekly events like block courses and conferences held at the university. These events must be planned on the basis of the weekly teaching events because they must not collide with it. In parallel to these aspects, the aim is to evaluate the approach at other universities.

ACKNOWLEDGMENT

This work is kindly funded by the Oberfrankenstiftung grant for project number 02803.

REFERENCES

[1] O. Rossi-Doria, M. Sampels, M. Birattari, M. Chiarandini, M. Dorigo, L. M. Gambardella, "A Comparison of the Performance of Different Metaheuristics on the Timetabling Problem," in 4th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2002), 2003, pp. 329-351.

[2] S. Abdullah, E. Burke, and B. McCollum, "An Investigation of Variable Neighbourhood Search for University Course Timetabling," in 2nd Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA), 2005.

[3] M. W. Carter, "A Comprehensive Course Timetabling and Student Scheduling System at the University of Waterloo," in 3rd International Conference on the Practice and Theory of Automated Timetabling (PATAT 2000), 2001, pp. 64-84.

[4] B. McCollum, P. McMullan, B. Paechter, R. Lewis, A. Schaerf, L. Di Gaspero, "Setting the Research Agenda in Automated Timetabling: The Second International Timetabling Competition," INFORMS Journal on Computing, vol. 22, 2010.

[5] E. Commission, ECTS Users’ Guide. Luxembourg: Office for Official Publications of the European Communities, 2009.

[6] M. J. Kamfonas, "Recursive Hierarchies: The Relational Taboo," The Relational Journal, 1992.

[7] D. de Werra, "An introduction to timetabling," European Journal of Operational Research, vol. 19, pp. 151-162, 1985.

[8] B. McCollum, "The implementation of a central timetabling system in a large British civic University," in Practice and Theory of Automated Timetabling, 1998, pp. 237-253.

[9] A. Bode and R. Borgeest, Informationsmanagement in Hochschulen: Springer Berlin Heidelberg, 2010.

[10] I. Object Management Group, "Business Process Model and Notation (BPMN) Version 2.0," ed, 2011.

[11] W. I. f. H. d. U. Bamberg. (12.08.2011). FlexNow. Available: http://flexnow.uni-bamberg.de/

[12] H.-I.-S. GmbH. (12.08.2011). Lehre, Studium, Forschung. Available: http://www.his.de/abt1/ab10

[13] S. Piechowiak and C. Kolski, "Towards a generic object oriented decision support system for university timetabling: an interactive approach," International Journal of Information Technology and Decision Making, vol. 3, pp. 179-208, 2004.

[14] J. P. Pedroso, N. Moreira, and R. Reis, "A Web-Based System for Multi-Agent Interactive Timetabling," in International Conference on Knowledge Engineering and Decision Support (ICKEDS'04), 2004, pp. 1-6.

[15] T. Müller and R. Barták, "Interactive Timetabling: Concepts, Techniques, and Practical Results," in 4th International Conference on the Practice And Theory of Automated Timetabling (PATAT 2002), 2002, pp. 58-72.

 ISBN: 978-1-61208-177-9