Master Thesis

Integration of Resource Management Systems

For the University of Applied Sciences Central Hesse

James Antrim

November 2012

Central Hesse University of Applied Sciences
Department for Mathematics,
Natural and Information Sciences

Referees
Prof. Dr. Peter Kneisel
Oliver Christ

Date of the graduation
November 30, 2012

CHREMYLOS: There is too much of everything. Of love,
KARION: Bread,
CHREMYLOS: Songs,
KARION: And sweetmeats.
CHREMYLOS: Of honour,
KARION: Cakes,
CHREMYLOS: Courage,
KARION: And Figs.
CHREMYLOS: Of ambition,
KARION: Barley bread,
CHREMYLOS: High office,
KARION: And lentils.
-Aristophanes

Contents

Abstract

This work concerns resource management systems, their functions and the areas in which they are used. It discusses the resources necessary for scholastic scheduling and their modeling in the newly introduced Untis 2013 MultiUser software. The specific functions and workflows of the Untis and THM Organizer scheduling software are discussed including their security policies, concurrency management, their place in the THM system landscape, and the interface they share between these two systems.

1. Introduction

Over the last ten years the University of Applied Sciences Central Hesse (THM) has undergone vast and deep changes. The number of students has risen from under 10,000 to over 14,000. New THM campuses, installations, places, and methods of learning have come about, such as the Wetzlar Campus or the BDH Clinic in Braunfels, where students can get practical training in the usage and development in biological information systems.[fSA12] Research and further education of business or private students have gained increasingly important roles in addition to traditional education. At the Wetzlar Campus over 1,000 students take part in an educational concept which pairs traditional education with hands on experience in business for an all encompassing and cutting-edge experience. The number of available majors has increased to over 50, which is almost twice as large as it previously had been. This variety gives students more options as to what they study and a deeper more specialized knowledge of that field. The traditional German diplom degree program has been almost completely replaced with bachelor and master degree programs giving the students a more performance-oriented education and a degree which is standardized throughout Europe assuring THM Alumni better chances in a globalized job market.

Throughout all this modernization and adaptation of the education offered by the THM. The IT software infrastructure has been sorely neglected over the same time period. Multiple systems are in use with minimal transparency between them. Often even within the same organizational unit using the same system, stored data can have stark deviation from one data collection to the next.

The management of students takes place using HIS-SOS[1]. Grades in all departments, with the exception of department MNI, are managed using the HIS - POS. The management of modules and subjects is often undertaken using HIS-LSF, but just as often not managed at all. Facility Management (FM), library management, and document management each have internal software solutions.Scheduling takes place in multiple systems for the differing departments and campuses, among the most often used systems are Untis scheduling software and MS Excel spreadsheet software.

Interaction and collaboration between students has taken place using the collaboration platforms eStudy and Moodle. Interaction and collaboration between employees is decentrally managed, distributed differently across the vastly differing

[1]Hochschule Informations System a software offering various software solutions to educational institutions. `http://www.his.de/english/organisation`

subordinate organizational units. The THM intranet's system for the planning of appointments, eGroupware, is offered solely to the THM's employees. Students and student employees have no access to this system, which severely reduces both its use and usefulness. Internet is managed using various highly differing techniques by both centralized and decentralized organizations within the THM.

HIS is a large central system which, as previously mentioned, manages students, grades and modules. However, this system is antiquated and is scheduled to be replaced by a central campus management system in 2014. This campus management system will, however, never be able to completely cover all of the myriad demands and functions necessary. This necessitates the conception of a system landscape which covers the most urgent demands and bundles functionality sensibly into manageable units. Where this central solution lacks the functionality, clean software interfaces must be developed to enable its further development or the coupling with external software solutions.

This challenge has been countered over the last few years with the development of the iCampus project. This project has had this standardization as a goal. It has achieved a system landscape that although yet still heterogeneous, with multiple systems developed and used in parallel, which deliver much the same functionality. This produces a sense of stability and standards. The management of students and grades takes place as before within the HIS system. HIS-LSF was further developed and expanded to be the central repository for module description storage. As a further result of the iCampus project dialogues have been developed which are capable of formatted display of the module description contents. eStudy as a collaboration platform has been mothballed, making Moodle the sole collaboration platform for students. Untis is being developed into the central repository of scheduling information within the THM, with the exception of the Wetzlar Campus. The Joomla! 2.5.x content management system (CMS) is being developed to the sole internet platform for the THM's web presence across all central and decentral organizations, and with time interfaces are being developed from this platform to all other THM systems of any importance.

The THM has tested several campus management systems for a successor to HIS, such as Datenlotsen[2]. However, none of these had sufficient functionality in their scheduling software to cover its needs. Here, Untis which had already been used for a number of years decentrally by the Friedberg Campus and the departments MNI, KMUB, and BAU. Untis manages multiple schedule specific objects such as curricula, modules and subjects among many others. These resources have import and export interfaces to multiple systems, some of which are responsible for the management of other some of the same resources. The schedule planning in itself has strong dependencies with the content and temporal behavior of other university specific processes.

[2] A campus management software. For more information see `http://www.datenlotsen.de/index.php/en/`.

Building on the gaining predominance of Joomla! 2.5.x and Untis within the THM, the interface between these two systems has gained much on importance over the course of the last year. This interface, THM Organizer, was developed as software for the Joomla! CMS which is capable of importing Untis XML export files. This import then serves as the basis for a multitude of university specific functions.

These systems, the objects they manage, how they are managed, as well as their interfaces and dependencies in other systems will be discussed in this thesis.[Kne12]

1.1. Resource Management

This work is about integrating resource management systems. The systems which will be dealt with have, for the most part, previously been mentioned. This brings us to the next item for discussion: resource management.

Resource management can broadly be defined in terms similar to the four basic functions of persistent data storage: creation, reading, updating, and deletion of resources (CRUD)[Hel07]. These are used to build a layer of persistent data for the application layer where propagation and business logic are performed. Business logic is the extrapolation and synthesis of data based upon logical connections and process models.

These activities take on different meanings dependent upon the resource from whose perspective the activity is viewed. As applied to schools or, more specifically, colleges or universities, resource management has a few general perspective aggregates based upon the responsible organization.

Facility Management (FM)	Grounds, Buildings, Rooms, and Equipment
Human Resources Management (HRM)	Teachers and other Employees
Document and Library Management	Documents and Literature
Student Management	Students
Quality Management (QM)	Curricula and Modules
Schedule Management	Schedules, Lessons and Events

Creation, in terms of material objects, personnel and students, is the initial creation of the objects representing the resources in their respective information systems as part of the material acquisition, hiring and immatriculation processes. Creation in terms of abstract objects is not only, as with their concrete counterparts, the creation of representational objects in the respective information systems, but also the creation of the documentation dealing with these abstract objects.

Reading can be taken literally in terms of documentation, but in terms of other resources it could be seen as the browsing of a telephone book for the school, the retrieval of information in a spreadsheet program, the direct questioning of personnel. As we can see from the example of the telephone book, reading often implies that business logic has already been performed to filter the information about the

resources of personnel and sub-institutional organizational units down to their respective names and telephone numbers. This could, however, imply more extensive business logic such as quantifiable room usage statistics, employee pay, or the grade point average of all the students of a particular semester.

As with creation, updating can have a two fold relationship of changing the physical objects and simply updating stored objects representing the physical and abstract resources. Some examples would be the refurbishment of a room, the further education of an employee, or the modification of documentation. This in turn could demand changes to the electronic models of these objects when a room has more or less equipment available, when a teacher has new qualification, or a book receives a new assignment in the library catalog.

Deletion carries this double relationship further, but does not necessarily explicitly implicate the removal of the real resource. For example deletion could mean the simple removal of software, or the demolition of a building. However removal of objects representing resources from one system does not mean the resource itself is gone; it could simply mean the resource has become irrelevant to the organizational unit responsible for a particular information system. If a teacher who was previously active in two departments is currently only active in one, the department for which the teacher no longer holds lessons could be removed from their internal HRM system.

The concrete perspectives FM, HRM, Student Management, and Document and Library Management concern themselves more with concrete resources as previously exemplified in the above table. As QM's resources are more abstract in nature: specifically the curriculum of a course of studies and the abstract resources which come to exist in order to implement it. The concrete perspectives target the most efficient usage of the managed resources, whereas QM has as a goal the assurance of educational quality for students.

Scheduling tries to fulfill the goals of all these perspectives. It is a process of managing both concrete and abstract resources to ensure the utmost efficiency of room and teacher usage, while strictly adhering to the standards put in place by quality management, and taking special consideration of student needs. In itself it integrates the fruits of other management processes as the starting point for its own processes.

1.2. Methodology

The use of masculine pronouns when referring to people within this work has solely to do with convenience and brevity.

Acronyms are with few exceptions introduced first with their full names followed by the introduction of the acronym in (*round parenthesis*). This acronym is then used

consequently throughout the rest of the thesis, and can be referenced at any time in the nomenclature at the its end.

Independent of acronyms (*round parenthesis*) are also used to introduce terms which are synonyms for the concepts presented. This can be either taxonomical or conceptual dependent upon the context.

To aid understanding of the terms presented and their representation in the accompanying figures, where necessary the English terms are often followed by their German counterparts which appear in <<*doubled pointed braces*>>.

1.3. Limitations

This work focuses primarily on the scheduling aspect of resource management with the integration of other resource management processes as a source and a potential output goal of its processes. While it may make short references to other resource management processes, it will not go to any great lengths into the resource management processes upon which schedule management is dependent.

The version of the Untis software used within the scope of this thesis is Untis 2013. Several extra Untis modules play a significant role within this work among them the Multiuser module and the Infoschedule module. All references to Untis within this work refer to this configuration.

Untis is a very powerful scheduling program which itself is the source of several handbooks. It is impossible within the scope of this thesis to delve completely into every aspect of this program nor would it be practicable. Several features such as attendance, substitution and the WebUntis will not be discussed here.

Those aspects of the THM Organizer software which are not directly involved in the scheduling process such as curriculum and appointment management will also not be discussed.

1.4. Overview of the Thesis

The resources dealt with in school scheduling software and Untis scheduling software in particular have been handled up until this point by multiple systems; even within the same organizational structure for the same resources. Each of the systems and administrative processes for resource management have their own taxonomy and required information to model the resource both within the system and in every day speech. The handling of these resources, systems and their interaction raises important questions.

- What are the resources to be managed?

- Who manages them?
- What are the typical actions performed with them?
- What data is required by each system?
- What functionality should be offered by which system?
- How should this functionality be implemented?

Three tasks emerge for the implementation of a system based upon the resources which provide the material for chapters two, three and four. These have strong dependencies between one another and often reveal themselves to be cyclic in their development. However, in order to maintain some semblance of structure, they have been encapsulated as much as possible.

Chapter two will define and model resources. The definition will include their perceptions and current use at the THM. Before getting into the modeling key attributes of the resources will be listed which are important to the perception of the resource. These attributes in turn provide a starting point and a reference for the modeling of the resources in Untis. The modeling used will be presented along with its advantages and disadvantages with respect to the definition and attributes. The presentation of these resources provides the foundation for understanding what needs to be managed within the system to be implemented.

Chapter three will examine the system dependent theoretical and practical foundations for the integration of the systems involved. These systems Untis and THM Organizer will be examined with respect to their required system environments, security policies, concurrency features, workflows, and their place in the broader THM landscape. This then shows how the resources presented in chapter two are processed and how the systems deal with their resources both internally and in cooperation with one another.

Chapter four will discuss the practical work done in accompaniment to this thesis. Within Untis this consisted of: the taxonomical changes to the data input for three THM departments and the changes in modeling to coerce the output in Untis XML export into a usable format. Within THM Organizer this consisted of: schedule validation, schedule version control and delta calculation, schedule and resource persistence, and data retrieval and presentation. Lastly chapter four will discuss recommendations for changes to the Untis workflows and normalization database structures.

Chapter five will summarize some of the key advantages and deficits of the existing systems and workflows and the steps to harness, improve or circumvent them. It will discuss the limitations of the accompanying technical realization, and provide insight into possible directions for future development. Ultimately, it will provide a summary of the resource management systems and processes.

2. Scheduling Resources

Scheduling is in principle an association of teachers, rooms, the subjects which are to be taught, and the times at which lesson content should be taught, resulting in a set of lessons. Although this is a bit simplistic it provides us with a good starting point for the more abstract resources which result as consequences of the associations of these objects. In this chapter the administrative, basic, organizational and temporal resources, and the associations necessary for scheduling will be examined.

A detailed description will be given of each of these objects. Then extrapolating from this description we will take a look at some of its more important attributes. This will give us a somewhat more precise look at the subject of each section and its relation to other previously discussed objects within the chapter. We will then use this perspective to understand how the previously described object definitions were modeled using the resource objects available in the Untis 2013 software. Examining the functionality of the equivalence, and potential benefits and drawbacks of these design decisions.

In the first section we will take a brief look at administrative resources used to organize schedules. Then we will examine the basic resources of teachers, rooms and subjects. Organizational resources and attributes will then be explained. The concept of time in regard to lessons and lesson groups will be explored including resulting resources and the consequences of temporal modeling. Finally the lessons themselves will be introduced binding all of these elements together into a consistent whole.

2.1. Administrative Resources

When a user opens Untis 2013 using the multiuser the first things he will be confronted with are the log-in dialogue and the resources associated with it. These resources, school <<*Schule*>>, school year <<*Schuljahr*>> and version, are largely administrative resources and exist solely to restrict the amount of information available for schedule coordinators to the amount directly necessary to the scheduling process and to uniquely identify the schedules they have created.

In Figure 2.1 on the following page we see the log-in dialogue and its resource selection dialogues. These administrative resources will be discussed in the subsections on Departments, School Years and Versions. The creation dialogue is displayed in

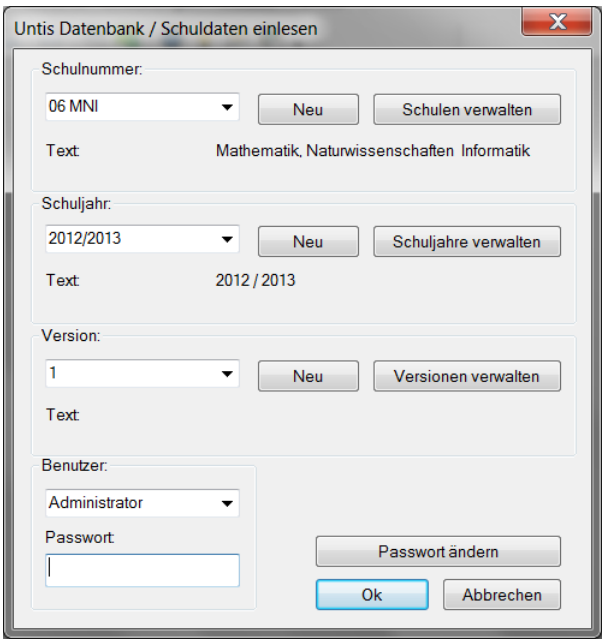


Figure 2.1.: Log-in Dialogue

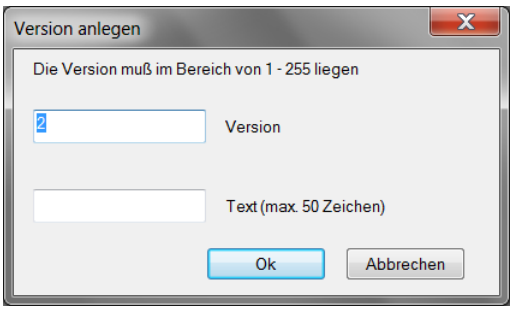


Figure 2.2.: Administrative Resource Creation Dialogue

Figure 2.2 on the current page. The dialogue for all three administrative resources is exactly the same except for the wording.

2.1.1. Departments

Departments are semi-autonomous organizational units at the THM, and as such are responsible for their own scholastic scheduling. Unlike other resources departments are a self containing entity which posses or share other resources. All resources are dependent upon them for definition and value. In the real world as well as within the Untis program, they exist as an administrative tool to ease the complexity of the scheduling process.

As a tool there is little that can be said about them, for, in effect, they allow for specific user interactions and a define roles in the workflow of the scheduling process.

As such they will receive much more attention in the next chapter Distributed Planning and System Integration on page 41 where workflows and the system security concept will be elaborated upon.

Attributes

Name	The full name of the department.
Short Name	An acronym containing the initial letters of the substantive words from the department's full name.
Number	Departments are associated with a number between one and twenty-one.

Modeling

While Untis does have a department resource that can have a partially similar workflow to that of a Untis school, the resource time restrictions and planning priorities associated with the individual departments could not be modeled using this resource within Untis. For this reason, it was the recommended that the departments of the THM use Untis schools to implement the actual departments of the THM.[1] This recommendation was used and the Untis school administrative tool is capable of modeling all the above mentioned attributes.

As stated above, the ability for departments to compete for resource availability as discussed on page 32 was the deciding factor in choosing to use Untis schools as opposed to Untis departments. Untis schools could perform this action, whereas Untis departments could not. The converse action, that of departments managing shared resources, is however supported for Untis departments and not for Untis schools. This can add administrative overhead if the schedule coordinators do not have the necessary privileges to modify each others data.

The decision to use Untis schools instead of Untis departments did however make planning conflicts more visible as resource usage became visible for the first time without schedule coordinators having to exchange information via telephone or e-mail. This under the caveat that the lesson producing the conflict had already been planned. It also allowed the department managed resources to be restricted to the amount of resources actually relevant to the planning in the respective department. If the departments had been modeled as Untis departments every room in the THM would have had to have been directly accessible for every department because they would have all been a part of the same Untis school.

[1] `Departments need to plan using differing planning priorities. Another problem is the amount of data, if all of it were modeled within one school this could result, under circumstances to performance problems.`[Gru12a]

2.1.2. School Year

The school year defines the temporal framework for scheduling. It should define the run of a school year including any subordinate planning periods.

Attributes

Name	An identifier for the school year.
Startdate	Date upon which the school year begins.
Enddate	Date upon which the school year ends.
Planning Periods	Subordinate planning periods of the school year.

Modeling

Unfortunately the Untis 'school year' is defined only by its name, and, in turn, only serves to name a group of schedules. The school year stands in no causal relation to the actual dates of the school year used within the schedule which are input using a completely different user interface. It also stands without causal relation to any subordinate planning period. This resource is completely unusable for its intended purpose is however required to create schedules with, and for this reason they cannot be ignored. However Untis does provide resources for the creation of actual school years and planning periods. These resources are discussed in Planning Periods on page 28.

2.1.3. Version

Versions are the status and data conditions of a schedule during its various stages of development.

Attributes

Department	Department responsible for the schedule.
Planning Period	Planning period in which the schedule data should take place.
Creation Date	Date and time when the version was created.
Schedule Data	Complete data and status of the schedule at the time of the versioning.

Modeling

Untis versions are completely independent working copies of a schedule that can stand without any actual relation to others of the same school and school year. This deviates sharply from that of the American Heritage dictionary which defines a version as "A particular form or variation of an earlier or original type...".[Ame12b] Although it is not guaranteed, inspected or directly supported Untis assumes that this definition is carried out through exporting, opening and saving of a previous version as a new version.

The lack of direct Untis support of versioning and revision control is one of the challenges of the integration in the practical accompaniment to this thesis. At the THM the actions of revision control implemented are functions of THM Organizer and will be discussed in Version Management and Persistence on page 82.

2.2. Basic Resources

This chapter describes the basic resources: teachers, rooms, and subjects. These items are deemed basic resources because they are unquestionable in the concept of a lesson and are described by the Untis software directly as such.

2.2.1. Teachers

Teachers are persons hired by the responsible bodies to convey the knowledge of a set fields of expertise to groups of students. Because of teacher's complexity as human beings, their description here solely encompasses that which is necessary to model them as resources for scheduling at the THM. Teachers are identified within resource management systems at the THM by means of a 'hg' number, because of it's enforced uniqueness and brevity in comparison to the teacher's name. This consists either of the letters 'hg' or four letters taken from the first and last names followed by an integer value.

Attributes

Name	Teachers are commonly identified by first and last names.
Number	A unique number assigned to the teacher.
Fields	The field or fields of expertise in which the teacher is qualified .
Departments	Departments for which the teacher holds lessons.
Availability	Lesson periods during which the teacher is available to hold lessons.

Modeling

Untis offers the resource teacher which are capable of modeling all required attributes and Teachers were relatively easy to model in Untis. Untis teachers additionally have place for attributes such as title, contact information, gender, birth date, values for cost calculation, and much more. The fields attribute can even be modeled in a special field which helps Untis optimize schedules.

The simple teacher *<<Dozent>>* dialogue is displayed in figure 2.3 on the following page. This dialogue shows how the required attributes with exception of departments have been modeled onto attributes available to Untis. While fields such as first and last name *<<Vorname/Nachname>>* are self explanatory, and the personnel number*<<Pers.Nr.>>* field can easily be explained as holding the

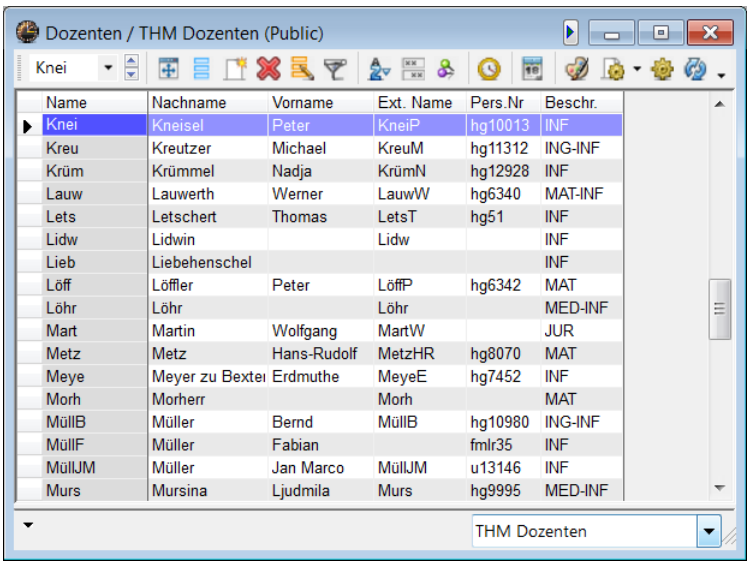


Figure 2.3.: Teachers Dialogue

teacher's 'hg' number, other fields such as name (short name), external name, and description<<*Beschr.*>> need a bit more explanation.

The name, or short name, is an Untis internal, human-readable key. This key can be used in multiple Untis dialogues for faster input of resources. In many cases the dialogue will require the use of the name whereas a select few will also allow the use of the long name, in this case last name, for data input.

The external name is a name which is used to denote resources used by multiple departments. This name is uniquely identifiable among all departments and it has been strongly recommended that the external name be the same as the short name.[2] Teachers are a problem in this area as often details such as first names and 'hg' numbers are often unknown and must be collected.The THM 'hg' number is, from the standpoint of its content, the best attribute to use for this value. However, they are very often human unreadable and when inputting data the schedule coordinators would have to cross-reference them with a list, which made this choice very unappealing. The other choice which would have provided a suitable solution would have been the use of first and last names. This solution has two larger problems. The first is that first names are often unknown to the differing schedule coordinators, much the same as the 'hg' number. The second is the amount of information the coordinator would need to input in order to get a valid result in the event of identical last names, namely at least the complete length of the last name. A further problem

[2] In Untis the names are used as IDs for the resources. If the same resource has differing IDs in separate data instances, this must be kept in mind every time data is exchanged between these instances. This is also the case for WebUntis. If the data is added using this interface, the differing names must be aggregated using aliases. However in principle Untis can resolve these differing names.[Gru12a]

with external resources is propagation, this will however be discussed in detail in Data Propogation on page 50.

The Untis description field holds the reference to the teacher's field or fields of expertise. There is a Untis field which is actually designed to hold teacher fields, however there are multiple problems with modeling the data in this manner as will later be described in the subsection Fields and Competence Centers on page 23.

Lastly as described in the section Departments on page 10, the departments had to be modeled using the Untis school resource, leaving this attribute without equivalent within the teacher objects.

2.2.2. Rooms

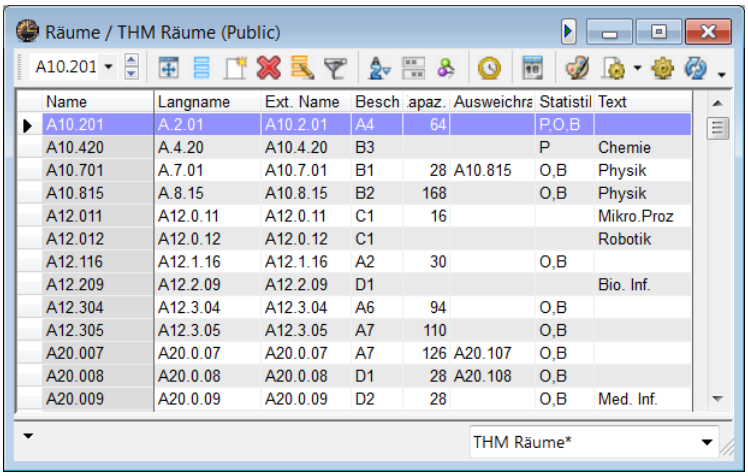
Rooms are the physical locations where lessons can take place.

Attributes

Name	An identifier for the room.
Building	Building which contains the room.
Floor	Floor on which the room is situated.
Type	The type of room.
Equipment	Special equipment which can effect what lessons can be held in it.
Capacity	The number of students it is designed to seat.

Modeling

The room *<<Raum>>* resource is offered in Untis, and most of the required attributes are existent in some form or another.



Name	Langname	Ext. Name	Besch	apaz.	Ausweichra	Statistil	Text
A10.201	A.2.01	A10.2.01	A4	64		P,O,B	
A10.420	A.4.20	A10.4.20	B3			P	Chemie
A10.701	A.7.01	A10.7.01	B1	28	A10.815	O,B	Physik
A10.815	A.8.15	A10.8.15	B2	168		O,B	Physik
A12.011	A12.0.11	A12.0.11	C1	16			Mikro.Proz
A12.012	A12.0.12	A12.0.12	C1				Robotik
A12.116	A12.1.16	A12.1.16	A2	30		O,B	
A12.209	A12.2.09	A12.2.09	D1				Bio. Inf.
A12.304	A12.3.04	A12.3.04	A6	94		O,B	
A12.305	A12.3.05	A12.3.05	A7	110		O,B	
A20.007	A20.0.07	A20.0.07	A7	126	A20.107	O,B	
A20.008	A20.0.08	A20.0.08	D1	28	A20.108	O,B	
A20.009	A20.0.09	A20.0.09	D2	28		O,B	Med. Inf.

Figure 2.4.: Rooms Dialogue

In figure 2.4 on this page we again see the Untis fields name (short name) and external name. With rooms as well as every resource other than teachers we also see the

field long name *<<Langname>>*. As with teachers the name and external names are unique, human-readable keys, with the name being for department internal use, with the external name for interdepartmental use. The long name is an additional name that had been until recently the same name used for the name field. However, at the THM, there was some discrepancy between the name used in resource management systems and that which they actually have on the physical rooms. For this reason the long names of rooms in two of the THM buildings have been used to record this discrepancy and the long name is used for external data output. This discrepancy also had the advantage that the name could also be slightly modified from the THM standard of <building>.<floor>.<room> in this case with one less period which is less typing when inputting room names and makes some of the names short enough to be completely displayed in the Units selection boxes. One of which can be seen in the upper left corner of the figure.

Both the building and the floor attribute are not available in Untis. The THM, as stated in the last previously stated, has it's own standardized coding of room names. This allows for the storage of both of these values in the name. However this has two disadvantages. Firstly some names are too long to be displayed in most Untis selection boxes without manual adjustment to their size. For example, in figure 2.4 on the previous page we see that the addition of a single period would cause the final digit of the name to no longer fit within the display of the select box. Secondly when filtering for the rooms of a particular building or floor, schedule coordinators are confronted with rudimentary regular expressions while using the Untis filter system.

The room type could be relatively comfortably input into the room's description *<<Beschr.>>* field. More on room types on page 20.

The room equipment was modeled using the Untis statistics *<<Statistik>>* field. This allowed for the comma separated input of single character abbreviations for room equipment similar to the way in which the same information was previously stored within the Excel sheets of FM. The only real disadvantage of modeling the information in this way is that statistic codes are not manageable resources in Untis. Statistic codes cannot be associated with text fields which would then be readable to tell other information systems that 'B' stands for video projector *<<Beamer>>*.

The capacity *<<Kapaz.>>* is a standard room attribute in Untis and was easily modeled.

2.2.3. Subjects & Modules

The American Heritage Dictionary defines a subject as *"A course or area of study[...]"*. [Ame12a] Sadly most other definitions are every bit as vague or self referring. In this work a subject will be defined as a thematic subset of a field of knowledge. Subjects are highly codified at the THM to meet the standards of the accrediting institutions.

Because of this they have a great many attributes. In the description only the subject number, effort, preliminary work and proof of proficiency will be discussed as their explanations are to cumbersome to fit comfortably in the attributes table.

A module is a very small group of subjects which together form a learning unit whose definition would be exactly the same as that of a subject, with the exception that it can take several subjects to take up the somewhat wider palate of themes associated with the word module. Since the typical module size at the THM is exactly one subject modules hereafter receive no mention due to their negligible importance.

The subject number is a brief interdepartmental identifier. The formatting of the subject number varies greatly between departments at the THM. MNI for example uses two letters which describe the subject field, a number indicating the niveau of complexity, and three further numbers. KMUB by contrast uses a five digit integer value.

The effort is the value in credit point of the course. At the THM one credit point corresponds roughly to 30 hours of effort on the part of the student, with 10 hours being time spent physically in the subject's lessons and 20 hours individual effort.

The preliminary work is the work which the student must complete to be able to complete the course. This is typically a set number of written homework units, but could be many things such as attendance of a certain amount of lesson units, or a team project.

The proof of proficiency is the way in which students should demonstrate their mastery of the subject's themes. While typically an examination, the proof of proficiency can be many things such as a paper, a software project, or a speech.

Attributes

Name	Full name of the subject.
Nomenclature	A short unique identifier for this subject within the subject catalog.
Responsible	The names of the teachers responsible for the development and teaching of this subject.
Goals	Themes addressed and the level of proficiency students should display in them upon completion.
Effort	Value in credit points for the course.
Lesson Types	Methods used to teach this course.
Preliminary Work	Work which must be completed during the course.
Proof of Performance	The way in which students should demonstrate subject proficiency at the end of the course.
Frequency	How often the subject is offered. (Annually/Semiannually)
Literature	Recommended and required reading material.

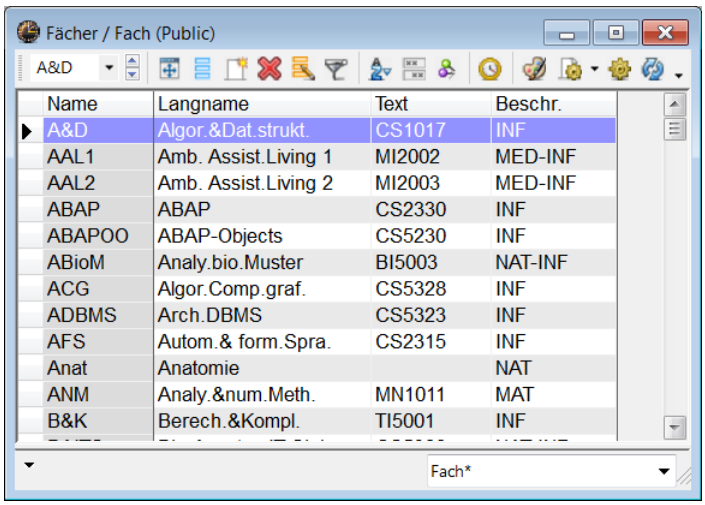


Figure 2.5.: Subjects Dialogue

Modeling

The modeling of subjects *<<Fächer>>* in Untis is somewhat rudimentary, restricted almost to the bare minimum of that which is required for the scheduling process. As seen in figure 2.5 on the current page the modeled attributes had to be restricted to four. The name and long name *<<Langname>>* have a special place within the THM as often the name as seen here is the colloquial name more often used, with the long name being the official name of the subject.

Unlike teachers and rooms the field external name does not exist for this resource. This reflects Untis design decision to implement inter-school resources as competing instead of sharing. In any account often subjects and the teachers responsible for them are shared between departments. Biology for example is taught by the same teachers for both the department MNI and KMUB. Almost every subject department Social and Cultural Sciences (SuK) teaches is shared with another department.

The subject number *<<Text>>* is a unique identifier within a department's curriculum catalog. Although differing in format in between departments it is currently used to retrieve information from the HIS-LSF information system which has been modified to accept all of the attributes listed above and many more. This number would in principle make the optimal inter-departmental identifier for this subject which, although the same themes and information are presented, may have differing names in different departments.

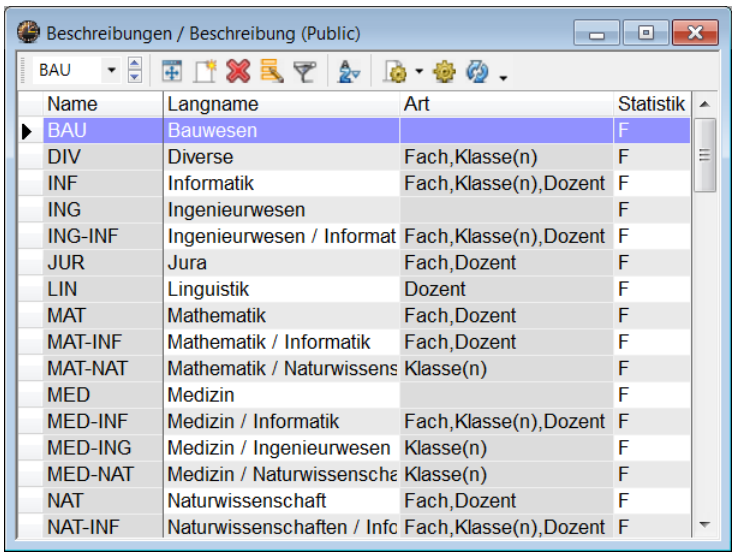
The description *<<Beschr.>>* field has been used to hold the field attribute not mentioned above. More on fields in the subsection about them on page 23.

2.3. Organizational Resources

Organizational resources help to describe and define other resources. While not directly necessary to define lessons and, by consequence schedules, they greatly enrich the basic information, make the lessons and schedules easier to understand, and can be used to organize the applicable basic resources in ways useful to business logic.

2.3.1. Descriptions

Untis does not have a description or categorization resource unique to any resource type. Instead they offer a special resource type which they call descriptions *<<Beschreibungen>>* which can be used on rooms, teachers, subjects, and subject pools.



Name	Langname	Art	Statistik
BAU	Bauwesen		F
DIV	Diverse	Fach,Klasse(n)	F
INF	Informatik	Fach,Klasse(n),Dozent	F
ING	Ingenieurwesen		F
ING-INF	Ingenieurwesen / Informat	Fach,Klasse(n),Dozent	F
JUR	Jura	Fach,Dozent	F
LIN	Linguistik	Dozent	F
MAT	Mathematik	Fach,Dozent	F
MAT-INF	Mathematik / Informatik	Fach,Dozent	F
MAT-NAT	Mathematik / Naturwissens	Klasse(n)	F
MED	Medizin		F
MED-INF	Medizin / Informatik	Fach,Klasse(n),Dozent	F
MED-ING	Medizin / Ingenieurwesen	Klasse(n)	F
MED-NAT	Medizin / Naturwissensch	Klasse(n)	F
NAT	Naturwissenschaft	Fach,Dozent	F
NAT-INF	Naturwissenschaften / Infc	Fach,Klasse(n),Dozent	F

Figure 2.6.: Descriptions Dialogue

In the current version of Untis, associating a description with a resource *<<Art>>* can only occur when while editing a resource of a particular type the description is typed in or selected. This is somewhat inefficient as it requires that either the description itself is not created from the descriptions dialogue, but dynamically from the dialogue of the resource to be described, or that the description is created in the description dialogue and then that the schedule coordinator types it in exactly as if he was dynamically creating it.

Untis descriptions have been used to model room and lesson types, and fields of expertise discussed in the next three subsections.

2.3.2. Room Types and Groups

Room types are general categorizations of rooms according to their intended purpose. The rooms of the THM are roughly grouped into eight different main categories and over 40 subcategories. Each main category is associated with a letter and each subcategory extends this with a single number. For example a lecture hall with a capacity range of 131 to 185 students has the code A8 within the FM systems. A workshop for electronics has the coding W5. For a complete list of all room type codes see Appendix A.1 on page 93.

Room groups are sets of rooms aggregated predominantly according to their room types. For example all rooms which seat over 120 students could be aggregated to a group collective.

Attributes

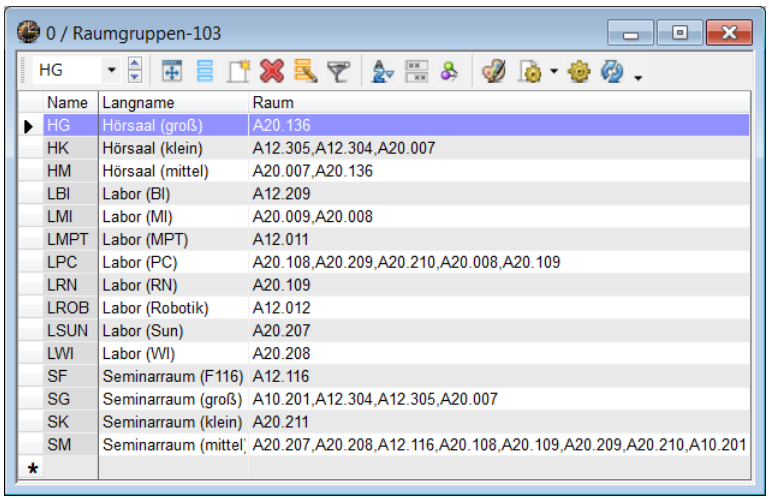
Code	A letter and a single digit which uniquely identify the room type.
Category	General categorization of the room's intended purpose.
Subcategory	A further defining feature of the room type.

Modeling

Modeling room types in Untis occurs in the two methods mentioned in the previous subsection on Descriptions. They are either created through the descriptions interface or dynamically by just typing them into the descriptions field. This has the same two disadvantages mentioned in the subsection Descriptions. These are that the data created dynamically is incomplete and that unassociated descriptions will not be recognized as describing rooms.

In order to identify room types as such both within and without Untis, the Untis Statistics *<<Statistik>>* field was used to hold the letter 'R'. Within Untis, this enables schedule coordinators to use the Untis filter tool to sort out any descriptions which do not have an 'r' in the statistics column so as to make sure the list of room descriptions is complete and correct. Outside of Untis it allows for the presorting of export data into it's correct types. More on this in the Validation subsection on page 83.

The modeling of room groups is one of the more convenient features of Untis. This allows several rooms to be aggregated according to one defining feature, or even single rooms to be named according to that defining feature. In figure 2.7 on the next page, we can see several groups defined using the rooms of the THM. The group PC Lab for example has the five general purpose computer laboratories of building A20. The group robotic laboratory and sun laboratory each only have one room, each however defined by special equipment within the room.



Name	Langname	Raum
HG	Hörsaal (groß)	A20.136
HK	Hörsaal (klein)	A12.305,A12.304,A20.007
HM	Hörsaal (mittel)	A20.007,A20.136
LBI	Labor (BI)	A12.209
LMI	Labor (MI)	A20.009,A20.008
LMPT	Labor (MPT)	A12.011
LPC	Labor (PC)	A20.108,A20.209,A20.210,A20.008,A20.109
LRN	Labor (RN)	A20.109
LROB	Labor (Robotik)	A12.012
LSUN	Labor (Sun)	A20.207
LWI	Labor (WI)	A20.208
SF	Seminarraum (F116)	A12.116
SG	Seminarraum (groß)	A10.201,A12.304,A12.305,A20.007
SK	Seminarraum (klein)	A20.211
SM	Seminarraum (mittel	A20.207,A20.208,A12.116,A20.108,A20.109,A20.209,A20.210,A10.201

Figure 2.7.: Room Groups Dialogue

2.3.3. Lesson Types

The way in which the learning goals of a particular lesson are to be reached is in no small way influenced by the typical methods of instruction. The instructors at the THM typically make use of one of five subject types: lectures, exercises, practical training, seminar, or small group work. These lesson types are by no means exclusive and are often used in tandem to provide the student with more enriching learning experience.

- *Lectures* are the most common lesson type consisting of a mostly frontal speech by the lesson teacher and most often feedback from students to themes of that speech. Lectures exist to convey knowledge which should be reflected upon by the student.

- *Exercises* serve primarily to help students to reflect upon conveyed theoretical knowledge through repetition.

- *Placement* or *practical training* serves to strengthen the students grasp of practical knowledge via hands-on learning. Typically placement has the completion of one or more projects as a goal.

- *Seminar* as held at the THM is a course in which students are given the assignment to research and strengthen their knowledge about a particular theme or interest. The students then hold presentations at regular intervals to discuss the results of their efforts.

- *Small group work* is usually similar to practical training. Lessons using this type of instruction method often attempt to teach the methodology of problem solving in a particular area while stressing social interaction and teamwork.

Attributes

Name	The name of the lesson type.
Abbreviation	A one to two letter abbreviation of the name.

Modeling

As with room types and fields this has been modeled against the Untis description resource, which can only model two similar attributes. Additionally lesson types have 'U' input in the Untis statistic <<*Statistik*>> field because of the type selection weakness previously discussed, similar to the 'F' entered within the statistics field in figure 2.6 on page 19.

A possible solution would have been to model the lesson types here defined as parts of the subject. This would have meant that for each lesson type a new subject would have to be created. This would have had the advantage of being able to say at the subject level how many hours such a pseudo-subject should take place, and theoretically making sure that the correct number of lessons take place before the lessons were put on the table each semester.

This approach however has several disadvantages.

First and foremost, the number of lessons is fit dynamically to cater to the number of students for quality assurance purposes. A lecture may be held for hundreds of students, but the corresponding exercise might only be held for 30 for quality assurance purposes. This would involve the creation of multiple exercise units, with the exact number calculated dynamically and not according to a quantity that can be laid out ahead of time attached to the subject.

Also, this would at least double if not triple the numbers of subjects. Untis was designed to handle the scheduling needs of grade schools and secondary schools. These have an extremely small set of lessons in comparison with that of schools at the university level. Whereas as school can use generic subjects such as math, English, and social studies, the subjects at University level are highly differentiated even in the same field. Creating a lesson's lecture, exercise, or other lesson type would make this type a codified part of the name. Components of Integrated Programming would then have to exist under two names Components of Integrated Programming - Lecture and Components of Integrated Programming - Exercise.

The information pertaining to every doubled and tripled subject would then also have to be copied. Not only at the time of creation, but each time changes were made to the course name or number. Since Untis provides no means of data propagation this would have to be done by hand each time, potentially leading to stale data.

Lastly, the name output of the lesson type as a part of the subject could only be adjusted by actually adjusting the name. If the lesson inclusive type should only output the acronym of the type, the full type name, or should not output the type at all cannot be set once the name has been fixed.

2.3.4. Fields and Competence Centers

Teachers by means of their education and experiences have certain particular fields of knowledge in which they excel. These fields logically mirror the fields whose themes they, the teachers, are should teach to students. These themes and their practical and logical implications form the material from which subjects are created. At the THM, it has been suggested to codify these fields to use them to build subject identifiers. As examples the abbreviation 'WIR-ING' could refer to economic engineering, 'NAT-INF' for information science having to do with natural sciences, or 'BAU' to denote a field having to do with architecture or construction science.

Often these fields also directly correlate to departmental competence centers which help to develop the curriculum for a course of studies. The teachers which work in these disciplines discuss amongst themselves contemporary themes and how best to introduce these themes into those presented within the lessons in order to assure quality and relevance.

Attributes

Name	Full name of the field or fields, optionally compounded with a hyphen.
Abbreviation	The abbreviation of the field or fields of the name.

Modeling

Modeling Fields as descriptions in Untis has the exact same weaknesses as with room and subject types. However Untis does have a unique tool to handle fields directly in the case of subjects, and indirectly in the case of teachers. The statistics field of these resources has an 'F' in it, because of the type weakness discussed previously.

Subjects can be aggregated to a set number of names. All of the subjects which pertain to a particular field can then be aggregated to a name which corresponds to their field. Not only that, they could also be aggregated to their level of complexity. The lessons, which are normally taught within the first three semesters of the six semester bachelor degree program, could be aggregated to 'Basic', or the courses of the master degree programs to 'Master'.

This way of modeling the information also enables certain Untis features. Untis can propagate the information input in the fields tab forward to the teachers calculating the number of lesson hours the teacher actually teaches while doing so. Untis is also capable of recognizing on hand of this information which fields have a shortage of personnel and while optimizing will use this information to plan the teachers with rare fields first. However there are several minor problems with this solution which make it at the present time unusable.

The first and most pertinent is the matter of visibility. The fields and levels of complexity associated with a subject can be included in the list output. This information is also output to Untis XML export which we use for the further processing

of information. Teachers, however, only display this information in the teacher abilities *<<Lehrbefähig.>>* tab of the expanded teacher information as seen in figure 2.8 on the current page. This is neither output to the list format in figure 2.3 on page 14 nor is it output to the Untis XML export. The button teacher abilities from lesson *<<Lehrbef. aus Unterricht>>* calculates the teacher's fields on-hand the lessons with which they are already associated as explained in the previous paragraph. When activated this gives the option of entering the subjects, the fields, or both in the table on the lower left side.

Another disadvantage is the lack of direct resource management. All Untis resources which have their own management propagate changes made to the resource in question within the context of its school, school year, version and period. Changing a field in a subject or a teacher does not propagate the changes made to the field name. In addition lack of management disables the Untis auto-complete list select feature, and disables comma separated entry of multiple fields. The comma separated entry of multiple fields is allowed however the resulting value is perceived as its own entity when Untis propagates the information from subjects to teachers. This may also be conjoined with the calculation of teacher hours since letting multiple fields be calculated could lead to inaccurate totals in the numbers of hours a teacher teaches.

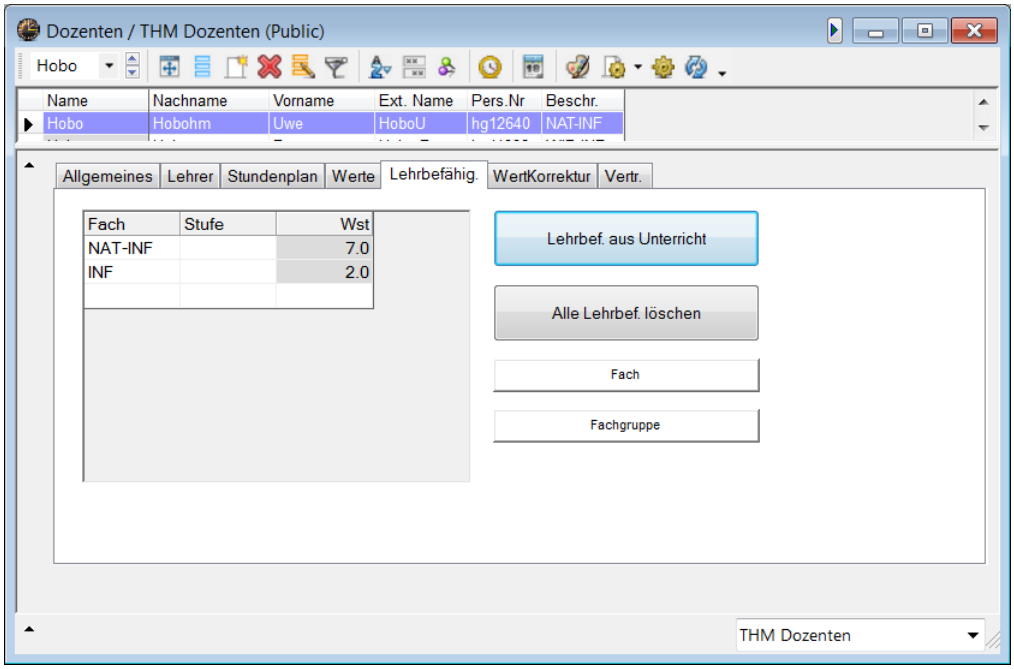


Figure 2.8.: Teacher Field Dialogue

2.3.5. Subject pools and Courses of Study

Subject pools at the THM are a group of subjects gathered together for organizational purposes based upon a many distinct criteria. In scheduling it is preferred to

plan subject pools so that as few lessons as possible take place at the same time.

The largest subject pools are a mirror of the accredited curriculum or course of studies (CoS) required for a particular degree. Which is to say every student which wishes to achieve a particular degree needs to complete a certain subset of the courses available within the CoS for his particular degree. These subject pools are seldom directly planned because of their large and unwieldy nature. When planned this subject pool will usually indicate an incomplete set where the missing subjects are either unknown or performed by an external organization. As an example of an incomplete set, Department MNI frequently has to plan other departments lessons into rooms for which it is responsible. Rather than research the information to correctly input the set, the person responsible often created placeholders representing the unknown subjects and teachers to both reserve the resources while planning and to prevent others from planning into room reservations. As an example of a largely externally conducted set, Department KMUB has a Biomechanics, Motor Activity and Movement Analysis CoS which is conducted jointly with the Justus-Liebig University in Giessen. Because of KMUB's annual planning cycle, large amounts of subjects can be seasonally omitted. The result of the externally held subjects by Justus-Liebig University and subject omission is a set of only five subjects, which can easily be planned and displayed. Within the subjects required from one degree program to another there may be significant overlapping, however all other subject pools are subsets of the courses available for one particular degree program.

The first noteworthy criteria for one such subset subject pool is the current semester of a student group. Most degree programs recommend subjects to be completed in a particular order so that knowledge is gained incrementally, with the knowledge in the latter course building on the knowledge presented in the former. This subsetting seeks to ensure a consequent and sequential acquisition of knowledge while the student progresses toward his particular degree. This criteria is the most often used for subset creation. Due to planning restrictions causing some departments to only offer subjects annually, this subsetting can be extremely important to students and their responsible department planning coordinators.

The second criteria is thematic or preferential. Students may often times choose between multiple electives or between thematic subject pools, i.e. minors. Often there are several electives from which the student can openly choose a certain number. Often courses belonging to these subsets are grouped together for a better overview. In the case of thematic subject pools students are usually required to choose a certain number of subjects to complete out of a much smaller pool to strengthen knowledge in one particular field of study.

The last criteria for subsetting is quality assurance. Often large numbers of students can be a detriment to learning via hands-on activities, or certain subjects are designed to be taught through work performed in relatively small groups. This can result in a qualitative division of students into smaller more manageable learning units. An example would be Department MNI's first semester bachelor students.

In all four bachelor programs offered by Department MNI students are required to complete the course Object Oriented Programming. Because of the hands-on nature of the material to be taught and the large amount of students over different lessons are being offered for this subject. Another example would be the load splitting performed by Department KMUB. Several courses for a particular semester are delivered to the full load of students, however other the students are broken down into several smaller groups to visit other courses where such large numbers of students would be impracticable. In effect they form a subdivision of the students for a particular semester.

A final noteworthy attribute is that of collision behavior. Within most subject pools subordinate to the CoS itself a certain degree of freedom from collision is often required or desired. For example students of the third semester of a particular course of studies should be able to visit every single one of their required subjects. These should then be planned so that none of them take place during the same lesson period.

Other subject pools such as the minors in management or marketing from the economics department should be planned so that their students should be able to visit the courses required for both their majors and minors, but at the same time, from a planning standpoint, it does not matter whether the management and the marketing students can visit the lessons of each others minors. These minors could then be planned so that they take place during the same lesson periods providing that teachers and rooms are available for parallel instruction.

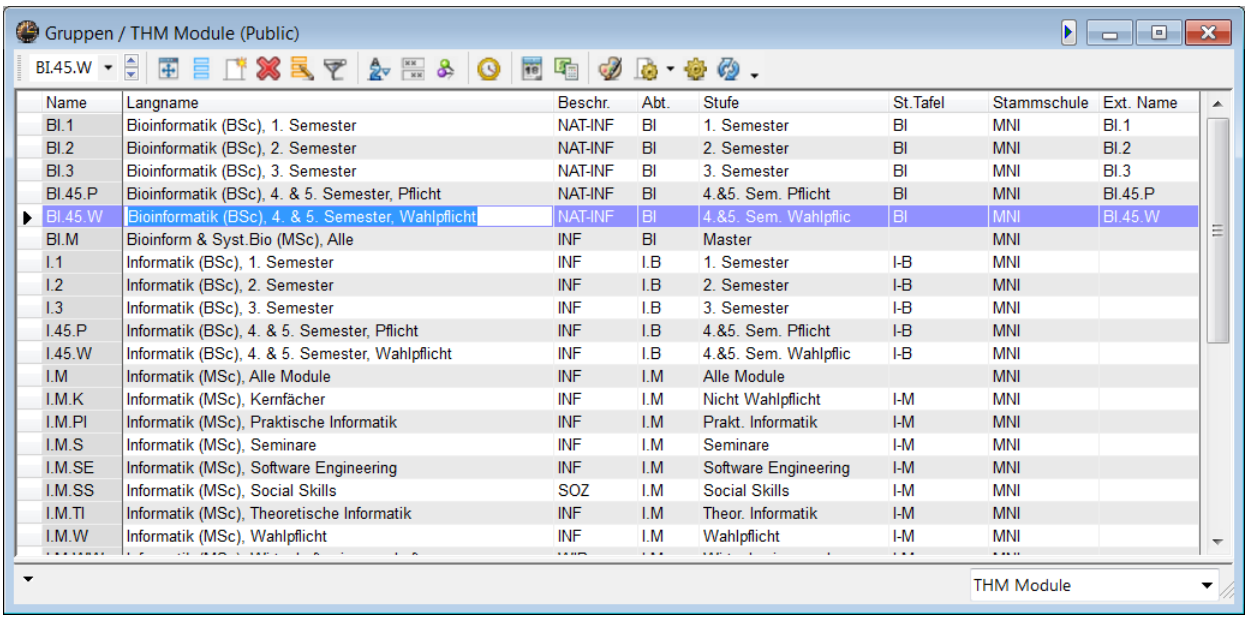
Lastly, subject pools such as the electives of MNI should be planned so that they do not collide with the required lessons, but at the same time a degree of parallel planning within its own lessons is acceptable. Students can chose the lesson which best suit their interests and their own schedules. If a conflict exists between two lessons in the same period it may be possible for the student to contact the department schedule coordinator and ask if the collision can be avoided. If no scheduling solution to the problem exists the student must chose between which subject he wishes to visit.

Attributes

Name	An identifier for the subject pool.
Parent	The name of the parent subject pool.
Subsetting Criteria	Criteria used for subsetting the parent pool.
Collision Behavior	Behavior of this subject pool with respect to lesson collision.

Modeling

Out-of-the-box Untis software refers to this resource as a class *<<Klasse>>* or subclass, in the Untis manuals, stemming from its initial development as a software designed for grade schools. While not far off the mark this name fails to convey the granularity of the second and third criteria for subsetting.



Name	Langname	Beschr.	Abt.	Stufe	St.Tafel	Stammschule	Ext. Name
BI.1	Bioinformatik (BSc), 1. Semester	NAT-INF	BI	1. Semester	BI	MNI	BI.1
BI.2	Bioinformatik (BSc), 2. Semester	NAT-INF	BI	2. Semester	BI	MNI	BI.2
BI.3	Bioinformatik (BSc), 3. Semester	NAT-INF	BI	3. Semester	BI	MNI	BI.3
BI.45.P	Bioinformatik (BSc), 4. & 5. Semester, Pflicht	NAT-INF	BI	4.&5. Sem. Pflicht	BI	MNI	BI.45.P
BI.45.W	Bioinformatik (BSc), 4. & 5. Semester, Wahlpflicht	NAT-INF	BI	4.&5. Sem. Wahlpflic	BI	MNI	BI.45.W
BI.M	Bioinform & Syst.Bio (MSc), Alle	INF	BI	Master		MNI	
I.1	Informatik (BSc), 1. Semester	INF	I.B	1. Semester	I-B	MNI	
I.2	Informatik (BSc), 2. Semester	INF	I.B	2. Semester	I-B	MNI	
I.3	Informatik (BSc), 3. Semester	INF	I.B	3. Semester	I-B	MNI	
I.45.P	Informatik (BSc), 4. & 5. Semester, Pflicht	INF	I.B	4.&5. Sem. Pflicht	I-B	MNI	
I.45.W	Informatik (BSc), 4. & 5. Semester, Wahlpflicht	INF	I.B	4.&5. Sem. Wahlpflic	I-B	MNI	
I.M	Informatik (MSc), Alle Module	INF	I.M	Alle Module		MNI	
I.M.K	Informatik (MSc), Kernfächer	INF	I.M	Nicht Wahlpflicht	I-M	MNI	
I.M.PI	Informatik (MSc), Praktische Informatik	INF	I.M	Prakt. Informatik	I-M	MNI	
I.M.S	Informatik (MSc), Seminare	INF	I.M	Seminare	I-M	MNI	
I.M.SE	Informatik (MSc), Software Engineering	INF	I.M	Software Engineering	I-M	MNI	
I.M.SS	Informatik (MSc), Social Skills	SOZ	I.M	Social Skills	I-M	MNI	
I.M.TI	Informatik (MSc), Theoretische Informatik	INF	I.M	Theor. Informatik	I-M	MNI	
I.M.W	Informatik (MSc), Wahlpflicht	INF	I.M	Wahlpflicht	I-M	MNI	

Figure 2.9.: Subject Pools View

This resource is capable of modeling solely the name of the attribute without problems. Like all Untis resources, subject pools can have both a name and a long name<<*Langname*>>. The name here is a short code indicating the CoS and its subsetting via the use of a period. In figure 2.9 on the current page, we see for example I.45.P indicating the required courses of the Information Science Bachelor degree program in the 4th and 5th semesters. The long name presents the same information in a manner which should be readable to anyone acquainted with the material being presented. The external name seen here for biological information science, a joint CoS offered by departments MNI and KMUB, reflects its use in multiple departments. However, this only allows for visibility of planned lessons on both sides of the departmental divide without actually allowing the responsible department to plan together.

The other information actually required has had to be modeled using Untis fields which are either under-implemented or unfitting. The CoS has been modeled here onto the department <<*Abt.*>>and schedule <<*St.Tafel*>>fields. The department field was used predominantly so that the subject pools could be both conveniently filtered according to the CoS and also that the CoS of a subject pool could be exported. Regarding the schedule field, it cannot be resolved to a text explaining what is meant, nor is it made a part of the exported XML. The level <<*Stufe*>> column seems like an ideal place to store the semester criteria which is for the most part incremental and therefore roughly fitting. This field is included in the XML export, however it has an enormous disadvantage in that the value has a maximal length of 20 characters. Untis does not warn the user of this behavior and allows the input of the complete value, later cutting off the excess characters.

The hierarchy reflecting courses necessary for minors can actually be modeled in two ways, the way allowed though the classes themselves is by using the Untis subclass functionality. This requires the parent courses to be marked with a value of one and the children to be marked with a value of two. This however doesn't completely address thematic subsetting because these subsets themselves may themselves have further noteworthy subsets, and the hierarchy depth is limited to one parent and multiple children exactly one tier deeper. The other way, by means of setting a subject pool collision identifier, is ironically not done by means of subject pools, rather is an attribute of lessons within the Untis program. Because of the Untis modeling method this will be discussed in Lessons on page 36.

The description *<<Beschr.>>* field contains values equivalent to those of teachers and subjects. This information is nice to have but without the granularity of actual subjects not of any value to the planning process.

The originating school *<<Stammschule>>* field seems like it would be a good method of description for the management of resources in general. From the name alone it would seem to convey a sense of responsibility for the associated resources. This is however not the case. The school in the field must be manually entered, and has no actual relation to the Untis resource school.

2.4. Time

The biggest change which the introduction of Untis 2013 brought into the system landscape at the THM was the change from a two dimensional to a three diminsional weekly plan. Two diminsional here refers to the combinatorics of days and lesson periods, while three dimensional refers to the combinatoric resulting from the use of days, lesson periods, and individual weeks. The lessons offered could, using the three dimensional plan in addition to temporal and organizational resources previously unavailable, differ from one week to the next more accurately modeling the actual run of many lessons offered. In the two dimensional plan weekly variance had been modeled through commentary to individual lessons.

2.4.1. Planning Periods

At the THM there are two planning periods per year the winter semester and summer semester. Semesters are defined not only according to the times in which standard lessons occur, but also include the range of dates until the next semester begins. This has the effect of making the summer semester somewhat longer than the winter semester. Within the semesters, cyclic lessons, block lessons and exams are planned. The length of these two planning periods can vary between departments as each can plan an individual cycle of lectures and exams.

The third planning period which deserves mention is the academic year encompassing these two semesters. As some lessons or subject pools are only offered in one semester annually, the academic year has often to be taken into consideration to confirm that the offering of subjects is in line with the guidelines created when the subject's CoS was certified. The planning period for the academic year is usually similar to the length of an actual year, however dependent upon the departmental scheduling needs there can be a shortfall between or an intersection of planning periods.

Semesters are identified by the season in which the semester ends and the year(s) in which they occur. Academic years are identified by the years in which they occur.

Attributes

Name	An identifier for the planning period.
Startdate	Date upon which the planning period begins.
Enddate	Date upon which the planning period ends.

Modeling

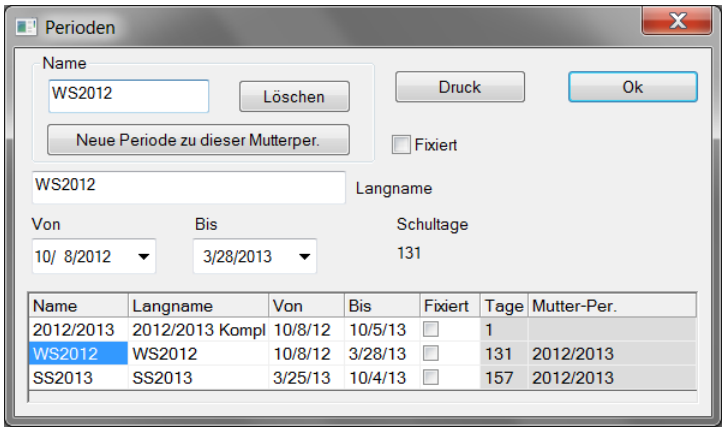


Figure 2.10.: Planning Period Dialogue

Both semesters and the academic year were implemented as the Untis resource Period, not to be confused with the, in Untis nameless, lesson periods discussed in the next subsection. Untis periods require both identity and length, are disjunct planning periods, and are hierarchically structured which makes them an almost optimal resource for modeling semesters. The only point of friction between the optimal model and the Untis model is that the Untis model limits the academic year to the length of a normal year.

All the noteworthy information can be modeled using the Untis resources, however, there are minor problems with the Untis implementation which can cause problems if not taken into consideration when performing data entry. The creation of periods demands that at least two lessons, discussed in the next section on page 36, have already been created. This requirement is counterintuitive since lessons require

a period of validity. Further, subordinate planning periods cannot consume the complete length of their parent planning period. The failure to leave at least one day belonging solely to the parent element can lead to system crashes.

Most importantly the propagation and validity of information is very strongly restricted. Resource attribute changes and lessons planned in the parent element can be propagated to the subordinate periods after a confirmation dialogue. The propagation of changes from subordinate planning periods to parent elements is not offered, nor is the propagation to chronologically sequential periods at the same or subordinate hierarchical levels. The propagation of subordinate elements into the parent element <<*Mutter-Per.*>>as far as lessons are concerned would seem like it should occur automatically since the subordinate planning periods take place within the temporal boundaries set by the parent element. Unfortunately this is not the case.

2.4.2. Lesson Periods (Blocks)

Lesson periods or blocks are units of time in which lessons can be slotted to occur. At the THM periods have three typical forms dependent upon the context of the week in which they take place.

Standard lesson periods take place during most of the semester. These are 90 minutes in length with 15 to 20 minute breaks between them and an hour planned for lunch. Depending upon the department and CoS these range from 30 to 33 in number. Six periods can take place Monday through Friday with three possibly taking place on Saturday.

Exams take place three weeks per semester. Examination periods are two hours in length and have no scheduled breaks. However finals are no longer than 90 minutes in length allowing students and teachers 30 minutes free time in between exams. Such periods are 25 in number with five taking place per day Monday through Friday.

Finally block lessons take place in breaks in standard lesson scheduling and take place for certain range of days from morning into the afternoon. These times are however set by the instructor of the block lesson and cannot yet be defined in the exact manner of the other two daily periods.

Attributes

Identity	Only the standard lessons have named identity. This is derived from the namce week day and the sequential number of the period upon that day.
Starttime	Time at which the lesson period begins.
Endtime	Time at which the lesson period ends.
Date	Date upon which the lesson period occurs.

Modeling

Untis has a resource designed specifically for this purpose. However the lesson periods are not handled individually, instead being created en masse as part of a time grid which is valid for the school.

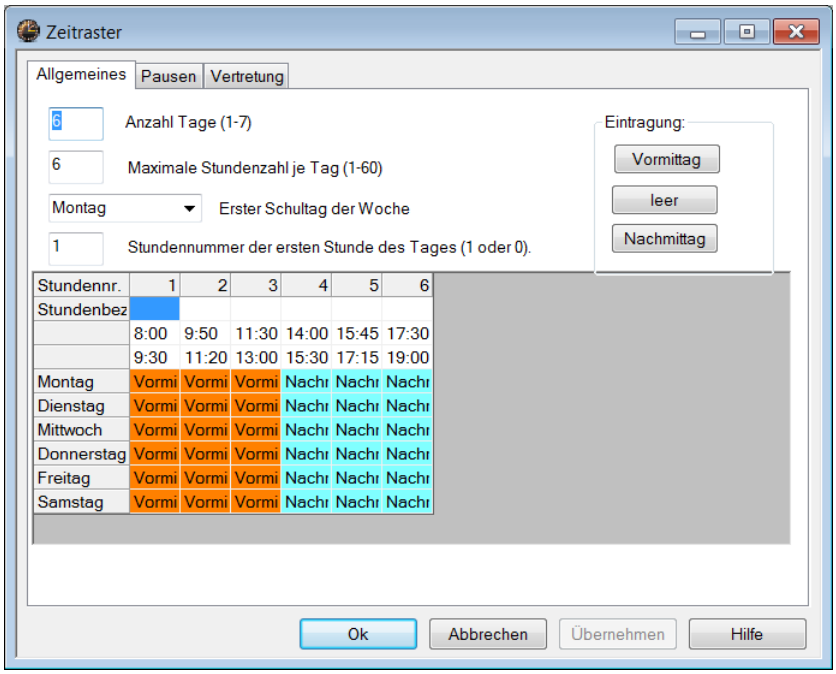


Figure 2.11.: Periods View

The schedule coordinator inputs the number of days *<<Anzahl Tage>>* upon which lessons occur and the number of lessons per day*<<Maximale Stundenzahl je Tag>>*. This creates a grid consisting of a number of blocks corresponding to the days multiplied by the number of blocks. This rule-of-thumb handling at the THM leads to three fictitious lesson periods occurring Saturday afternoon. The schedule coordinator further sets upon which day the school week begins *<<Erster Schultag der Woche>>* and with what integer value the numbering of lesson periods should begin *<<Stundennummer der ersten Stunde des Tages>>*. Lesson period start and end times must then be individually entered for each block. Lesson periods can also be marked as occurring during the morning or afternoon. The entry of morning or afternoon can effect the planning priorities dialogue for teachers and subject pools.

With the 'Multiple Time Grid' Untis extension module, multiple grids of lesson periods can be created. This multi-grid would have been able to directly address the lesson period needs for exams and block lessons, however there are three differences between the Untis implementation and the expectations of the scheduling coordinators which make the use of the multi-grid problematic. First the use of differing lesson period plans at the THM occurs only along the lines of predetermined calendar weeks, in Untis once multiple grids have been created they automatically run

parallel to one another. Secondly the choice of which lessons can be assigned using which grid occurs by assigning grids to subject pools. At the THM the same subject pools use both lesson period plans dependent upon the type of lessons being offered in the given week. Finally when the use of multi-grid is enabled the Untis lesson output is no longer displayed along the crisp lines of defined lesson periods, instead Untis schedule output graphic approximations of the lesson period run against a range of hours. While not a showstopper the non-explicit association of lessons to times is comparatively vague and could lead to confusion among students.

2.4.3. Time Restrictions and Planning Priorities

Time restrictions and planning priorities are not a resource per se, they are more of a department specific perspective of the availability of teachers and rooms. Although in principle restrictions and preferences are neither the same in meaning or importance, they can have a similar effect on the development of a working schedule and are implemented in Untis according to the same program tool.

Time Restrictions

Time restrictions have the most impact upon scheduling. Time restrictions are the consequence of resource reservations. These restrictions can effect teachers and rooms.

At the THM departments are allotted a certain set of rooms. However certain room types, most specifically large lecture halls, are few enough in number that their availability for each individual department must be discussed and planned in advance.

This discussion occurs between the department scheduling coordinators and FM. Schedule coordinators calculate on hand the expected number of students and the courses held in front of large numbers of students then number of lesson periods they need in rooms. They then discuss the assignment of rooms for specific lesson periods. As a result, each department leaves the discussion with a set number of lesson periods available only to them. These resource-lesson period assignments then theoretically cover the planning needs of the respective department.

Once the assignment has been made schedule coordinators go about the actual business of scheduling. This is an iterative process where, in order to attempt optimal room usage, schedule coordinators try to fit a lesson in the smallest room available to them which will fit the expected number of students. Should none of the rooms in the first selection be available, they try to schedule the lessons in the next larger room type. In the end should the coordinator be unable to plan the lesson he then requests unused assigned slots from other departments. Once an agreement has been made the schedule coordinator plans the lesson and informs FM about the change to the room availability assignment.

The availability of teachers is a less complex process. Teachers know for the most part when they have obligations during the work week and inform the schedule coordinators. The schedule coordinator then attempts to schedule lessons held by that teacher so as not to impinge upon those obligations. Should this not be possible, the coordinator discusses the situation with the teacher. If the schedule coordinator and teacher are unable to find a solution, the teacher with the scheduling conflict may then be exchanged with one available for the time in which the lesson must occur.

Planning Priorities

Planning priorities like time restrictions can affect teachers, rooms and subject pools. Teacher planning priorities are optional notations as to when the teacher prefers to hold his lessons, or would rather not hold them. Planning priorities as applied to rooms or subject pools allow the schedule coordinator to guide high and low volume scheduling times.

Attributes

Department	Department from whose perspective the availability or planning priority exists.
Resource	Resource whose availability or planning priority is to be recorded.
Lesson Period	Lesson period whose availability or planning priority is to be recorded.
Value	Availability or priority for the resource during the lesson block from the perspective of a department.

Modeling

Untis allows planning priorities to be associated with rooms, teachers, subjects and subject periods. However, instead of being related to the departments resource available within the Untis software, they are related to the school administrative resource. This has had dramatic repercussions for the further scheduling process as previously stated in the section on Departments on page 10.

The lesson periods for resources can be given an signed integer value ranging from -3 to 3. These values can be interpreted from the implementation as 'absolutely do not plan here' for the -3 value up until 'plan here if at all possible' for the 3 value. While the assignment of values is handled the same for all values, the -3 value has a blocking value when using any planning dialogue, fulfilling the need for the THM time restrictions.

Figure 2.12 on the next page displays the planning priorities *<<Zeitwünsche>>* of one of the teachers at the THM. As we can see in the illustration there are multiple ways to assign planning priorities by lesson period, weekday, mornings, or evenings. Entering a value in a larger block of time overwrites the subordinate units of time.

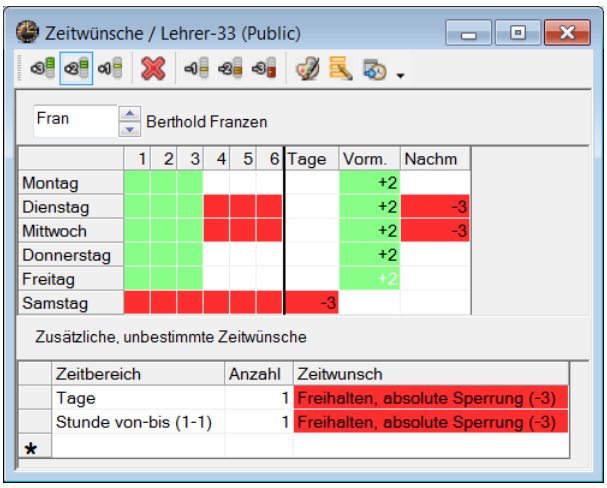


Figure 2.12.: Teacher & Subject Pool Planning Priorities

In this example the teacher prefers to hold his lessons mornings and has for these lesson periods the value two. The teacher also has obligations which prevent him from teaching Tuesday and Wednesday afternoons for which the value -3 has been recorded. Additionally the teacher's department in the example does not schedule lessons on Saturdays. Consequently, the teacher also has a restriction recorded for that day. The Untis developers suggest that whenever possible to use an aggregate such as day, morning, or afternoon to assign planning priorities to lesson units to make the assignment of the same value to multiple lesson periods more comfortable.[3]

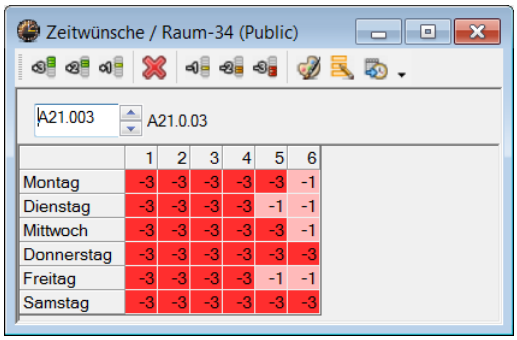


Figure 2.13.: Room & Subject Planning Priorities

As seen in figure 2.13 on this page, rooms and subjects have a more rudimentary mask for the recording of planning priority values than that for teachers and subject pools. As regards rooms, this entry mask fits very well with the workflow of the planners at the THM. Rooms which are not shared in between departments are unrestricted for the responsible department. Rooms which are shared are assigned for specific lesson periods and typically not by any lesson period aggregate.

[3] It's about comfort.[Gru12a]

2.4.4. Temporal Behavior

Often times lessons will occur with a certain amount of weekly periodicity, that is to say a certain lesson may occur in intervals with a number of weeks as multipliers. For example, if a lesson could only be held every second week, it would be said to have a weekly periodicity of two. It is also conceivable that a certain few lessons may occur over several weeks starting or ending abruptly within the otherwise ordinary run of a semester. In Untis these behaviors are called 'lesson groups'. Untis allows this sort of temporal lesson behavior to be modeled for a single lesson or group of lessons and form a separate type of lesson categorization.

Attributes

Name	The unique names of the lesson group.
Startdate	Date upon which the lessons assigned to this temporal behavior should commence.
Enddate	Date upon which the lessons assigned to this temporal behavior should cease.
Periodicity	Weekly behavior of this temporal behavior grouping.
Calendar	A more specific description of temporal behavior using a set of dates.

Modeling

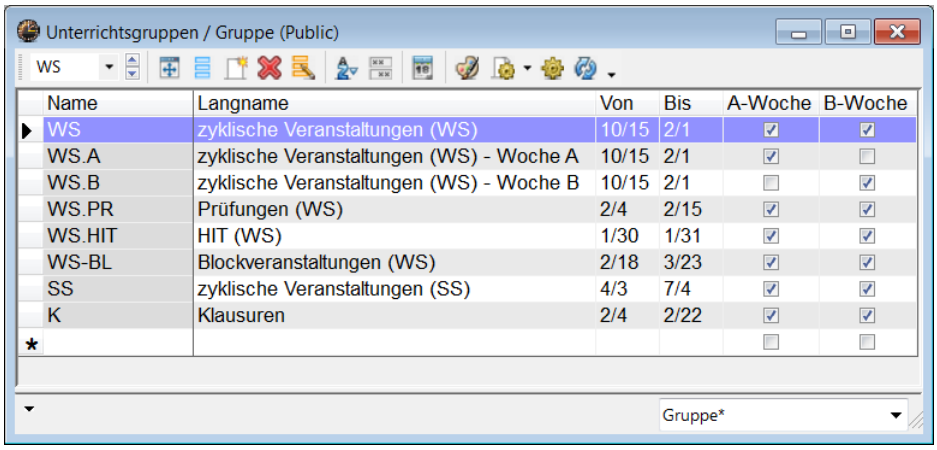


Figure 2.14.: Temporal Behavior Dialogue

Untis is capable of modeling all of the the mentioned attributes for temporal behavior in its lesson groups. Like most other Untis resources these have a name which is a short unique identifier and a self explaining long name.

Each lesson group can use individual methods of defining temporal behavior separately or in tandem. The basic restriction of a begin and end date can be defined directly using the appropriate dates.

Should the lessons of the planning period require periodicity, i.e. if they repeat themselves in a weekly rhythm of a few weeks, the periodicity must be calculated.

The lowest common denominator of the number of weeks in the cycle of repetition is calculated to determine the overall periodicity and the lesson groups are marked accordingly. For example should the period have lessons which occur every two weeks and lessons which occur every three weeks, the periodicity for the planning period would then be six. The lessons which occur every two weeks would place a check mark in the boxes for weeks a, c and e, or b, d and f. The lessons which occur every three weeks would have to be marked as occurring as either on weeks a and d, b and e, or c and f.

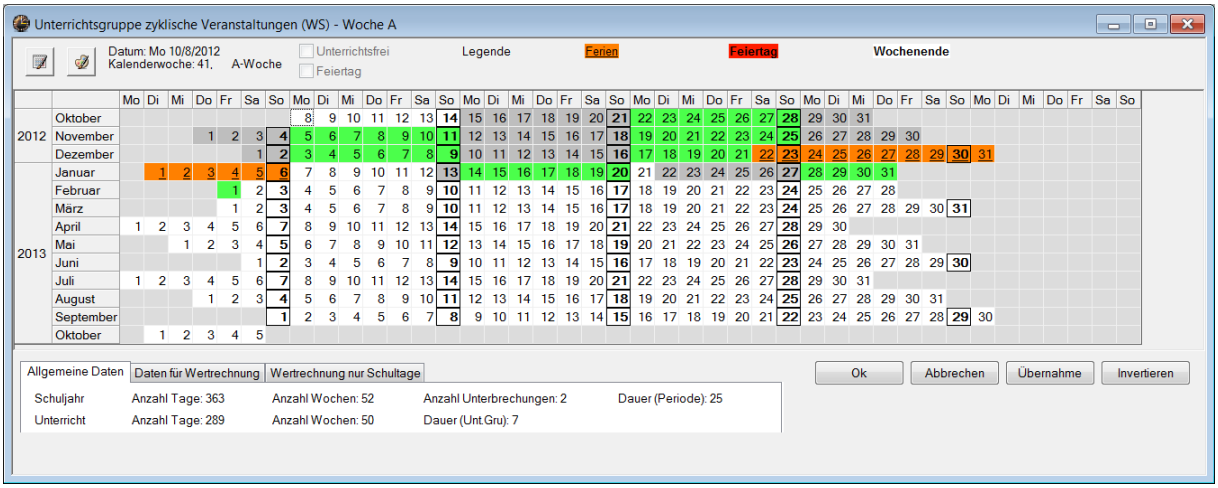


Figure 2.15.: Temporal Behavior Dialogue Calendar

Further the use of the calendar can also restrict the run of a course to certain days, weeks, months, or any combination of them. In figure 2.15 on the current page, we see the calendar where one can set the days directly in which no lesson is to occur (white). The weeks expressing the periodicity of two are shown here in alternating green and gray. Orange marks the winter holiday season. As seen in the figure, Untis automatically sets the first week after the holiday to that week which would have occurred first in its absence. The 21st of January in the figure is also white, displaying that no lessons are to be planned for that day.

2.5. Lessons

The end product of scholastic scheduling are lessons. As a resource they exist as the association of all the previously mentioned resources, attributes, and associations. Lessons are planned by a department for a particular school year or semester. They are instances of a subject or subjects taught by one or more teachers and take place in one or more rooms. These lessons are by extension of their subjects and teachers themes in at least one field of knowledge. The rooms the lessons take place in are often restricted by the lesson type or method of instruction to a suitable room type.

Lessons should be visited by students trying to fulfill the requirements of one or more CoSs or subject pools. They take place in a planning period on particular lesson periods, according to a specific temporal behavior and the restrictions and planning priorities of the resources involved.

2.5.1. Attributes

Subjects	Subjects which will be taught within the lesson.
Teachers	Teachers which will convey the subject content during the lesson.
Rooms	Rooms in which the lesson will take place.

Form	The way in which subject content will be taught.
Quantity	Number of lesson instances which must take place to convey the subject(s) contents.
Departments	Departments responsible for offering this lesson to students.
Subject Groups	The subject groups for which the lesson will be taught.
Lesson Group	Lesson group to which this lesson belongs defining the lesson's temporal behavior.
Commentary	Extra information relevant to the lesson which does not fit into one of the above rubrics.

2.5.2. Modeling

Optimally the planning/modeling of lessons in Untis is a two phase process. These phases are the entry of all relevant data and the temporal planning or the abstract and the concrete planning.

Abstract Planning

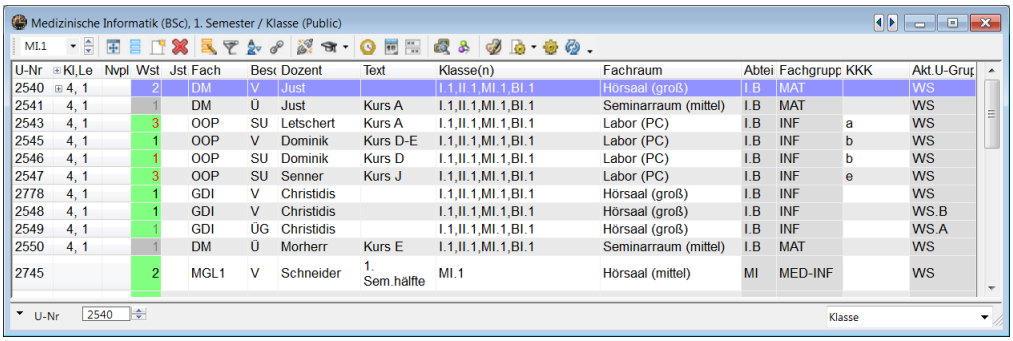


Figure 2.16.: Lessons Dialogue

All of the discussed resources and associations could be modeled using the Untis lesson resource. Unlike most other Untis resources lessons themselves have neither

a name or a short name, instead they have a lesson number provided by Untis at the time of creation.

Teachers can be selected or typed into the appropriate column. The addition of further teachers occurs by means of a so-called coupling row. When a row is highlighted, as lesson 2540 in figure 2.16 on the preceding page, a tiny plus symbol appears on the left. This creates an additional row underneath the active one. Here as before the teacher is then selected or typed in the appropriate column.

The name, if there were to be one, would be inherited from the subject or subjects which the lesson instantiates and the lesson type. A single subject can be selected from a list or typed in the appropriate column. Further subjects can be added behind the first subject as long as they are separated by a comma. This guarantees standardized output within and without Untis. Rooms, or room groups, *<<Fachraum>>* and subject groups *<<Klasse(n)>>* are both handled the same as subjects. Should a coupling row be added for a teacher, the scheduling coordinator should copy the values of the initial row for subjects, rooms and subject groups, then paste them into the appropriate columns of the other rows of the lesson.

The department *<<Abteilung>>* or CoS value is inherited from the first subject group in the list of subject groups. Unfortunately, this makes this value worthless as it then disregards all others and is not editable.

The field *<<Fachgruppe>>* column is inherited from the subject, and while it is consequently accurate it is also superfluous information because it is an inherited attribute.

The subject group collision *<<Klassen Kollisionskennzeichen (KKK)>>* value represents the allowed or desired collision behavior of a particular lesson in respect to the other lessons of its subject group. The empty value indicates that a collision should not be allowed. Alphabetic values indicate that lessons with the same value should be planned collision free, differing values indicate that a collision has no effect on the planning process. While not seen here this field can also carry numeric values. Numerical values operate on the opposite principle, namely that lessons with the same numerical value can be planned with collisions whereas lessons with differing numerical values should not be planned to collide.

The lesson group *<<Akt.U-Gruppe>>* indicates which lesson group defines the lesson's temporal behavior. In figure 2.16 on the previous page, we see lessons 2548 and 2549 which take place on alternating weeks indicated by the values WS.B and WS.A respectively.

Lastly, and most important, the quantity of blocks per week *<<Wst>>* or year *<<Jst>>*are entered in the appropriate columns in the list. Before the actual planning is completed, a number representing the number of unplanned lesson blocks will also be displayed *<<Nvpl>>*.

Concrete Planning

The concrete planning in Untis is the core of the program and has multiple powerful functions which implement it. These are extremely complex and cannot be given the attention they deserve within the scope of this thesis. In very vague terms there are two general methods: optimized planning and manual planning.

Optimized planning is the real strength of the program which uses an algorithm developed by Gruber & Petters to assign abstract lessons to concrete lesson periods, rooms, and teachers.

Manual planning can occur using multiple dialogues. The most powerful of these is the "planning dialogue" which provides an overview of resource usage in parallel with the resource currently selected to be used in planning. Other dialogues provide an overview of certain more manageable planning periods such as a generalized week or a specific calendar week among others. Here the schedules are presented using the weekly lesson period grid. To the left of this grid unplanned lesson hours are displayed as rectangles which can be dragged and dropped into specific lesson periods within the grid.

3. Distributed Planning and System Integration

Scheduling at the THM is a process which involves a multiple systems which are responsible for the electronic records of the resources managed, and several scheduling and facility management coordinators. Defining the system landscape was and is an iterative process based upon reflection and changes to the system environment. These changes to the environment can be in terms of new resource types, personnel, systems, or software.

The driving questions in this process are:

- Who is involved in the process?
- What actions can those involved perform?
- Which actions do they need to be able to perform to efficiently function?
- How can the various workflows be made more efficient?
- How can data best be aggregated?
- Have all necessary and desirable attributes and properties of the resources been recorded?
- How is data is the data to be processed?

The answers to these questions vary greatly according to the systems examined. Within the scope of this thesis were Untis and the THM Organizer component for the Joomla! CMS were to be integrated. While there are other systems in use in resource management within the THM system landscape such as LSF and Excel, their use is predominantly tangential to that of scheduling which receives the brunt of the content of this thesis.

For this reason, the two systems directly responsible for the scheduling will be examined independently according to the same criteria in this section. Each system will be briefly introduced. Then its required system environment, security policy and workflows will be discussed. Its view of concurrency, and the system's place in the broader system landscape including existing and potential interfaces to other systems will be explained.

A security policy as defined here is the definitions of users and roles, the actions these roles can perform and the resources upon which these actions may be performed. Security policies are enforced by access control mechanisms.

3.1. Untis

Over the summer of 2012 the Untis 2013 scheduling software was added to the THM system landscape. It's purpose is to provide clarity in interdepartmental resource management as regards scheduling and resource usage. It fulfills the THM's needs to model the required resources within schedules, but it's true strength is its powerful optimization algorithm.

3.1.1. Environment

Untis, although a complete program in itself, requires in its multiuser mode a subordinate landscape of systems. This subordinate landscape of systems is both the result of the requirements of Untis 2013 with its Multiuser module and the security requirements of the THM.

The first system requirement of Untis is a Windows operating system. In other operating systems this can be simulated with a virtual machine. The decision to solely offer Untis for Windows environment probably had to do with the market share of Windows on home and office computers.

Although not listed directly in any Untis documentation this Windows environment must have the .Net Framework in a version 3.5 or higher installed upon it. For the last several years, Windows has included this framework with its releases. Versions at or before Windows Vista or Windows Server 2008 will not meet this requirement and the .Net framework 3.5+ must be manually installed. The failure to have a .Net framework of this version installed will cause an Untis error message requesting the framework without specifying the version required.

The next requirement is a data source. The source can be either MS Access, MS SQL, or MySQL. Untis documentation makes no recommendations as to the choice of a data source. However there are several criteria useful in the decision.[Com08, W3C12, Wik12] Some of these criteria made MS Access completely infeasible for use at the THM. It is difficult to prioritize them as they are all equally decisive factors for not using Access. The THM will have large quantities of data to store which must be remotely accessed. Access is neither remotely accessible nor can it efficiently handle large quantities of data. Even Untis support of Access usage has side effects for system integrity and performance. Access supports only rudimentary data types.[W3C12] This results in talking down to the more advanced MS SQL and MySQL, only making use of their most basic and rudimentary features. Often going as far as to simplify the data itself such as storing dates as eight digit integers instead of using the SQL date type. The MS SQL and MySQL systems are roughly equal in size and scope with a slight edge in favor of MS SQL. The decision to use MySQL was one of personal comfort for the Untis administrator and server administrators at the THM.

The Multiuser module requires an ODBC connection to this source. The configuration of this source is well documented in the appropriate handbook.[PED12a] In order to avoid data inconsistency or loss Untis insists that both the ODBC connection and the source database be of the same base (32 or 64 bit). Although the configuration is well documented in Untis, the source providers often have vague identification of which base connection they are using. In the event of a base mismatch Untis outputs an equally vague warning that the system architecture is not supported.

The Untis conceptual landscape would depict this data source laying somewhere accessible in the internet with the ODBC connection and the Untis program installed and configured on every computer that was to access this data source. There are three problems with this concept. The first problem is that people that have little to no IT experience would be required to configure the ODBC connection on their personal computer, or that the system administrator would have to do this for each individual having to work with the system. The next problem is that each person involved would be responsible for updating his personal installation of Untis. Finally, the source server would be at the mercy of ODBC connections from an unknown number of personal computers whose system security cannot be guaranteed.

Since the THM had an in-house project dealing with the virtual systems, it was decided upon that all three of these problems would be dealt with simultaneously. A machine image was created with the ODBC connection preconfigured and Untis preinstalled. This image was then used as the template for the creation for a pool of virtual machines. These machines could then be remotely accessed from the personal computers of those involved in the scheduling and resource management processes.

3.1.2. Security Policy

Untis is a proprietary system which has a user definable security policy.

Users and Roles

Untis allows the creation and configuration of both users and user roles. Upon installation Untis automatically installs the preset user and user role administrator. All other roles and users must be created by the administrator or a user given the rights to do so by the administrator.

The workflow regarding the creation of users demands first the creation of user roles via the user rights dialogue *<<Benutzerrechte>>* as seen in figure 3.1 on the following page. A new role is created with the temporary name new *<<neu>>* which has no rights *<<keine Rechte>>* to perform any action on all objects.

Once a role has been created users can be created to join the group of users assigned to that role. This is done by means of the new user *<<Neuer Benutzer>>* dialogue

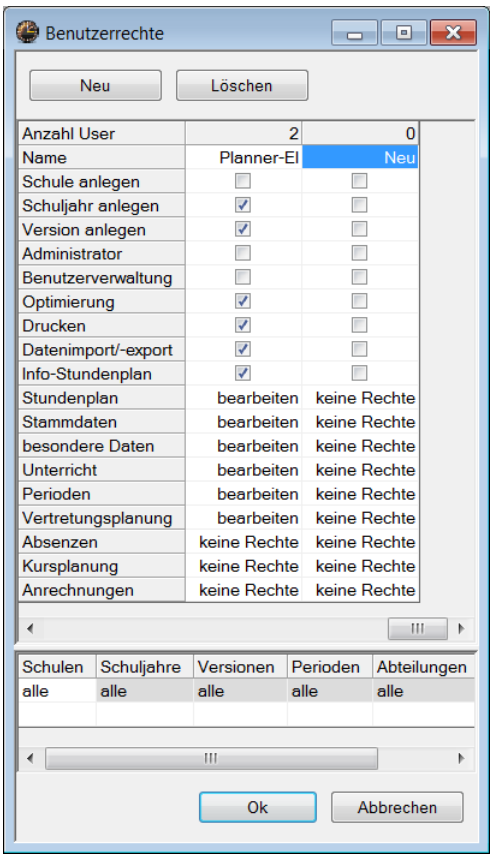


Figure 3.1.: User Rights Dialogue

in the right part of figure 3.2 on the next page. Here the user can be given a name and must be assigned to a group of users with a defined role *<<Benutzergruppe>>*.

In the new user dialogue the user creating the new user can also assign a password to the new user. This dialogue is the only place where the password can be assigned by a user other than the one being created. Should the created user later forget the password, his entry in the user dialogue must be deleted, and a new user must be created, because the password cannot be changed by any other user at any point in time. This has the consequence that all personalized data created by the user is also deleted and must be created anew.

Managed Resources

In the bottom part of figure 3.1 on this page we can see the five resources for which Untis explicitly can manage access to: school *<<Schulen>>*, school year *<<Schuljahre>>*, version, planning period, and department *<<Abteilungen>>*. As explained in chapter 2 the school, school year and version define the working schedules, with the planning period defining a swath of dates within the schedule. The Untis departments were used in this implementation to associate subject pools

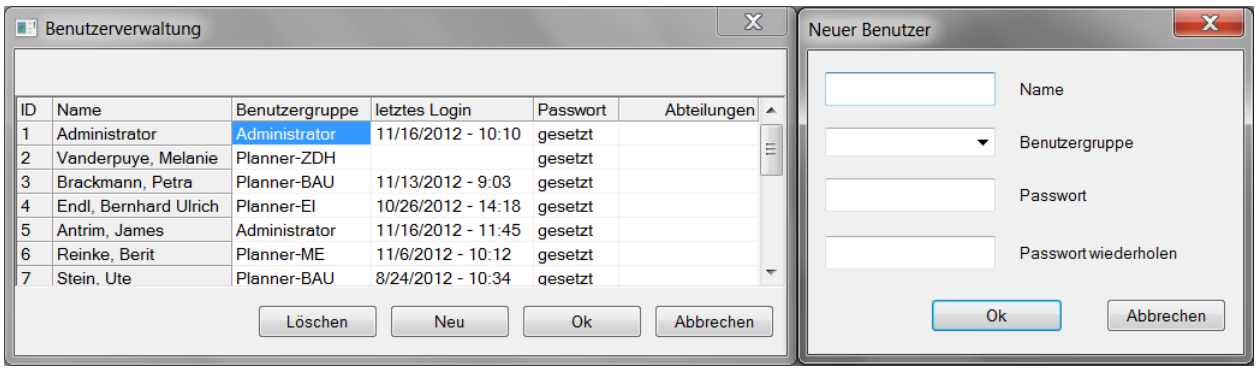


Figure 3.2.: User Management Dialogues

to CoSs, but in all other resources where they could have been used were left blank.

All user created roles can also have multiple lines of restrictions and multiple restrictions in each line as seen in figure 3.3 on the current page. The selection of managed resources is hierarchically structured so that without first choosing a school one cannot chose a school year, without choosing a school and school year one cannot chose a version. Once a version has been selected both planning periods and departments may be selected so that within the actual schedule the managed resources within a particular schedule may be selected as if from a matrix.

Schulen	Schuljahre	Versionen	Perioden	Abteilungen	nur lesen
01 BAU,03 ME	20122013	1	WS2012,SS2013	ME.BA,ME.DIP,ES.DIP,ME.MA	☑
06 MNI	20122013	1	2012/2013	I.B	☐
					☐

Figure 3.3.: Managed Resource Restrictions

According to this implementation of managed resources we see a continual degree of restriction from all data for a school down to the perspective of a single CoS in a single planning period in a particular version of a school year for one school. This is a very logical and consequent system for selecting a single resource or set of resources. However, this logic is slightly flawed, because it sets the end equivalent to the means.

Fore example, the administrator could wish to restrict a user's access to a certain perspective even for something as simple as reading. The administrator could want to restrict access along the lines of the hierarchy provided, it is however more likely that the administrator would like to restrict access to a resource within the plan itself irregardless of its place in the hierarchy. If for example, a certain teacher should be able to see or modify the resources of a particular department he should be able to assign that right directly, irregardless of the other managed resources which stand higher in the artificial hierarchy. This would allow that teacher to perform his duties for that department in every schedule and planning period without having these explicitly listed. Moreover, it would allow directed access to the resources in question. While from an implementation standpoint such a hierarchy may be

comfortable, it creates an enormous workload for the users managing user access, and prohibits that resources be managed across hierarchical boundaries.

What is sorely missed in the implemented granularity is the ability to restrict access to the level of individual resources such as teacher or subject pool across the boundaries of the currently implemented managed resources. The ability to define access as pertaining to these items would be a great help to the management of interdepartmental resources.

Actions

Untis has a plethora of predefined actions these are administered in the user rights dialogue as seen on page 44. These actions make up what can be explicitly done and many implicitly involve resources and other actions.

While some actions are either allowed or not allowed, many of the functions with lesser consequences can have four values: no rights, read only, edit views, and edit. No rights disallows the user to access that particular function. Read only allows in principle that the data for the respective resource can be seen. Edit views allows the user to edit which resource information he gets to see and its format. Edit allows users to edit the information recorded in the resource entries as well as edit which information is displayed.

Administrator The action administrator implicitly carries with it most other actions, these can however be explicitly removed by taking the appropriate check mark from the box. The most important implicit action for which administrator is required is the creation and management of external resources.

Create School/School Year/Version Actions such as create school/school year/version *<<Schule, Schuljahr, Version anlegen>>* are rather self explanatory. They also allow the user role access to the management dialogue for these resources and to edit their existing names. Atypical for Untis the delete button is not greyed out in the management dialogue although deletion is not allowed by these actions.

User Management User Management *<<Benutzerverwaltung>>* is one of the most important actions which can be assigned to the user. This action gives the user the ability to directly influence the security policy of the Untis installation by allowing him to perform almost all the functions in this subsection.

Periods/Basic Resources/Special Resources/Lessons/Schedule/Optimization The activation of periods *<<Perioden>>* allows users with this role to create planning periods within a schedule. Basic resources *<<Stammdaten>>* grants access to users for the management of teachers, rooms, subjects, and subject pools. Special resources *<<besondere Daten>>* allows the management of room groups, descriptions, and departments. Lessons *<<Unterricht>>* lets the association of these resources to create and manage lessons, while schedule

<<Stundenplan>> lets users to assign lessons to lesson periods via one of the many different Untis planning dialogues. Optimization *<<Optimierung>>* allows users to use the powerful Untis schedule optimization algorithms to create and optimize schedules. The granularity of these six actions allows the users with user management rights to customize workflows using each of these individual actions.

However, all of these options are to a greater or lesser degree required for the creation of schedules. It could be wagered to say that similar to the administrator action there could be a scheduler action which would in effect activate all six of these options.

Data Import and Export/Print/Infoschedule These three options allow differing data import and export options for data, print and a user interactive web directory.

The Untis MultiUser Handbook has further actions and more specific information on the actions available in Untis.[PED12b]

Control Mechanisms

Implementing the security policy of Untis administrators and users assigned to roles with user management privileges are Untis various control mechanisms.

Login Dialogue The first with which the user is comes into contact with is the login dialogue. This dialogue has multiple functions bound to it the most important of which with relation to the security policy is user authentication. This is the association of the user attempting to access system function with a user created by means of user management through use of the user's password. This dialogue is also the entry point for the creation of new schools, school years and versions, the saving of schedules in the database, and allows users to change their passwords.

Unfortunately once logged in all user access rights are relegated to the current school, school year, and version. Should a user often have to work in differing schedules this will require renewed logging in as opposed to being able to select from a list of resources to which one has access from already having logged in.

Preemptive Restriction With the exception of the login dialogue, Untis restricts functional access by building the user interface around the access rights of the user. In most cases this is accomplished by building inactive or 'grayed-out' menu entries.

Reactive Restriction In addition to the login dialogue and preemptive restriction, Untis performs access checks at the time of user interaction. Should the user not be allowed to perform the requested action, he is alerted via a pop up notification.

3.1.3. Workflows

Workflows within Untis focus upon the modeling of resources and the planning of schedules based upon these resources.

Data Aggregation

Data aggregation is the collection of the information necessary for the further scheduling processes. Initially this also includes all the data discussed in chapter 2. This is culled from a variety of sources FM and HRM files, web directories, discussions with teachers and schedule coordinators among many others.

However in a productive environment, much of this initial effort dissipates leaving only a few data aggregation tasks which must be performed before every planning period. The names and numbers of lessons needs to be calculated based upon the semester's curriculum and the number of students projected to visit the subjects of the curriculum. Based upon this information, room usage must be reserved through discussion among schedule coordinators and FM representatives for shared rooms. After this discussion the appropriated department specific room availability must be recorded into the planning priorities dialogue for each shared room. Then discussion with teachers must take place to determine their availability and preferences, which also must be recorded by the appropriate teacher resource.

This process is carried out by the Untis administrator, the schedule coordinators and the employees of FM responsible for room usage. It requires the actions periods, basic data, special data, and lessons, where each individual is responsible for data maintenance within the confines of their area of responsibility. It tangentially effects all manged resources but theoretically requires the restriction of managed resources for the schedule coordinators to their particular school managed resource.

Optimization and Planning

The optimization functionality can be an enormous relief to the workload of planners. Instead of having to plan each lesson by hand, the optimization functionality plans lessons automatically according to the number of unplanned lesson units and the teachers fields of expertise recording during data aggregation.

The reduction in the number of visible resources via departmental modeling as schools is also a great boon to the planning coordinators and to the Untis optimization algorithm in that they are not confronted with hundreds of rooms, teachers, subjects, and subject pools which are outside the responsibility of the respective department.

The Untis optimization algorithm recognizes rooms which disproportionally many subjects require. It recognizes teachers with rare qualifications. It recognizes the

allowed collision behavior of the lessons and the number of weekly and yearly hours which these lessons must be held.

Based upon this information the scheduling algorithm creates very good, but not necessarily perfect schedules. These must then be supervised by the schedule coordinator to assure that the produced schedule corresponds to the desired behavior.

This department based optimization and planning should be undertaken by schedule coordinators and requires the actions optimize and schedule. This also directly affects every managed resource, although the restriction to a particular school is fully sufficient to manage this workflow.

Interdepartmental Planning

However, at the THM multiple resources are planned through interdepartmental cooperation and can only be optimized to the extent of room and teacher reservations by means of the time restrictions and planning priorities dialogue discussed on page 32.

These restrictions and planning priorities can however only logically model needs met in advance at while room negotiations occurred. They cannot anticipate the iterative nature of interdepartmental planning and the chain reaction which can be caused by plan changes within one particular department. This is not a problem so long as the chain of reaction to scheduling changes can be dealt with within the assigned resources and their availability.

This brings us to real weakness in the current modeling and workflow: the visibility and availability in Untis of resources used by multiple departments. Aside from the issue of the freshness and visibility of resource usage across departments addressed in concurrency on page 52, shared resources such as the biological information science bachelor program are only partially exported according to which department planned which lesson block or part. This necessitated the workaround of local resource creation and planning to mimic the shared resources involved within each lesson planned in the other department.

Interdepartmental planning is undertaken by the schedule coordinators and FM representatives. It requires the action schedule and in a worst case scenario the action optimize. Schedule coordinators must coordinate iterative schedule changes around existing resource reservations and usage. When their own resource allotment no longer has the capacity for them to handle the lessons which must take place they must negotiate with other schedule coordinators to gain the allotment required. This alteration must then be reordered in the planning priority data in the resource in both schools and in the case of rooms reported for tracking by FM. Once the reassignment has occurred the schedule coordinators can associate the new resources or plan the lessons in the new lesson period availability of the contended resource. In a worst case scenario schedule optimization can be used should the chain of reaction

cause far reaching changes. Here, as in optimized planning, all managed resources are affected. Although necessary to create a workflow between departments and other institutions, the restriction to a school in this case is both an advantage for the optimization and planning and a disadvantage in terms of visibility and usage.

Data Propagation

Data Propagation within Untis is sorely underimplemented. Untis has external resources which solely exists to equate resources across the divisions of the managed resources mentioned in the user rights dialogue. These, theoretically, would be the ideal method of data propagation

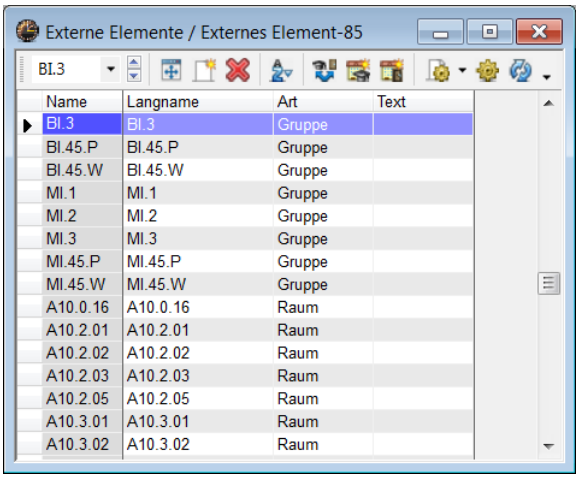


Figure 3.4.: External Resources Dialogue

In figure 3.4 on the current page, we see the internal resources dialogue. The individual entries of the dialogue are not fitted to the resources types modeled. Instead they record the name, long name, the resource type and a text. As pertains to external teacher resources, the long name within this dialogue then corresponds to the surname field of the teacher dialogue.

The propagation of these resources to their corresponding parts to the individual data stock of each school can occur by selecting the resources desired and clicking the button “transfer to basic data”. The direct propagation via this functionality sets as a prerequisite that the name of the resource in the external resources dialogue and the name of the field in the schedule specific dialogue correspond exactly. The direct transfer of external resource data to a dialogue void of these elements is not supported.

From this very rudimentary implementation we see that these resources are not conceived of as a means of data propagation but rather a means of enabling visibility between Untis school resources. This leaves the direct propagation of resource details up to the manual entry by several Untis users, and the consistency of the data

entered across managed resources becomes the job of the Untis administrator. The effort required for this work is compounded by the fact that resource management is dependent upon the managed resources school, school year and version.

Data propagation in itself is the responsibility of the Untis administrator, the schedule coordinators, and the FM employees responsible for room usage. As far as concerns the direct entry of resource data the workflow is exactly the same as that of data aggregation. As far as the maintenance of external resources this can only be carried out by the Untis administrator as the action administrator implicitly enables the management of external resources. External resources exist independent of the managed resources which are a part of the user rights dialogue.

Versioning

Untis offers the resource version as a means of defining a schedule. The involved workflow is relatively simple. In order to create a new version a user must be assigned to a role which has the action create version assigned to it. Either the user with such privileges then creates a completely new schedule in that version or saves an existing schedule as a version.

The documentation in the Untis MultiUser manual states that up to 255 versions can be created per school and school year. However, the documentation makes no statement as to the effects of versioning on resources managed from the perspective of multiple school resources.[PED12c] Because Untis was directly used productively, it was decided to avoid using this functionality until a test system could be put in place to test the effects of Untis versioning on external resources and interversional and interdepartmental collisions at a later date.

Exporting

Because of the fragile structure of the Untis database, the THM was advised to use the XML export as the interface to Untis external systems. This made the export of these files a vital workflow within the Untis system.

THM Organizer requires the export of the exporting organization and the planning period. Optimally these two attributes would be set automatically according to the Untis resource school and the active planning period. Whereas the active planning period can be exported by setting a check mark next to period, as seen in figure 3.5 on the following page, the school name exported is “Untis 2013”. To be able to associate the exported schedule with a school, the first header *<<Überschrift>>* field has been used to export the department name. Should THM in the future use the Untis version resource this value too could be exported by setting a check mark next to print schedule version, however, instead the automatic selection of the active version, or a select box with a list of available versions, Untis uses a text field here which the schedule coordinator must then correctly enter upon export.

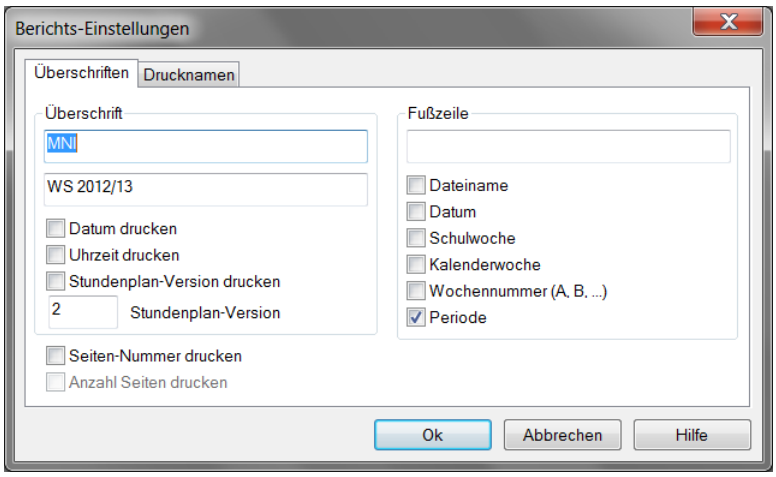


Figure 3.5.: Report Settings Dialogue

3.1.4. Concurrency

Within the Untis environment concurrency plays a rather central role as multiple users can perform actions virtually simultaneously. Untis deals with two different concurrency issues in two different ways.

The first concurrency issue is that of user interaction with stale data. User action is made locally persistent as soon as the user leaves the field being worked in. However database persistence is processed in a fixed 10 second tact.[PED12d] Every ten seconds, all local changes to data are checked against the corresponding entries in the database. Should no change have been made to the data in the database and that of the previous version of the local copy, the data is saved to the database. Conversely, instead of allowing the user to perform a dirty write, writing over data which has changed since the last update, Untis disallows the action and displays the message displayed in figure 3.6 on the current page.

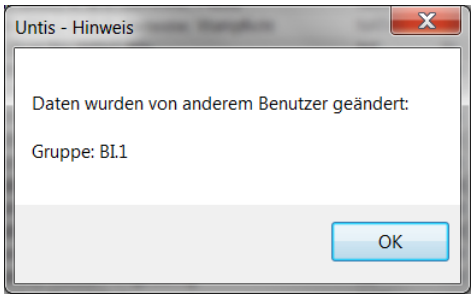


Figure 3.6.: Dirty Write Error

Unfortunately, this evinces that although the user was not in the situation when the data was retrieved, the user is now in a state similar to a dirty read, because the data he is using is no longer valid. Even after stale data has been noticed by

the tacted persistence mechanism, Untis does not force an actualization of the data displayed to the user. In order for the user to have fresh data displayed he must either refresh the window manually by pressing F5 or by closing and reopening the resource dialogue.

A similar, but further reaching, concurrency issue is that of external resource reservation. The booking of external resources blocks this resource across all schools which use this resource, and shows the blocking agent among other things in the appropriate planning dialogues of other schools using this resource. However, there have been instances where planned reservations in one school were not shown to other schools in a timely manner, even as far as a matter of days. As this the implementation of external resource persistence is not directly documented, it is hard to construct test cases because one does not know what should be expected.

Another concurrency issue is the restriction of key resources. Key resources are those which have consequences for the structure of the Untis resource school such as the management of school data and planning periods. Should a person with the appropriate access currently be working with one of these protected resources, login attempts by other users to this school will be denied. Should other users already be logged on while attempting to access these areas, the user attempting access will either be denied access to the dialogue in the case of planning periods as displayed in figure 3.7 on this page, or be reduced to read privileges in the dialogue in the case of school data.

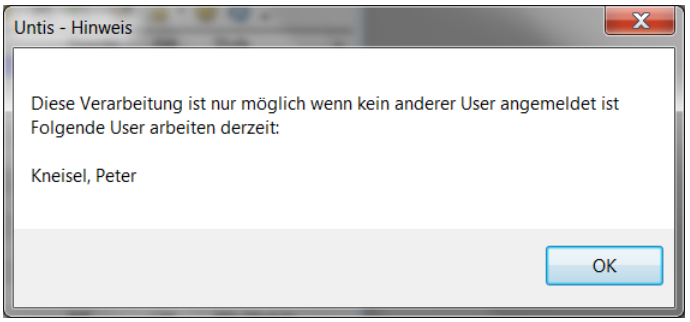


Figure 3.7.: Protected Access Denied Notice

3.1.5. Untis in the THM System Landscape

When Untis was originally being purchased, the driving force behind the purchase was its central database. This was to be a part of a broader concept for real time applications. These application would then use the data managed by Untis for the display of schedule information and any other data which Untis could provide. However during the Untis training, it was strongly recommended that no applications be directly based upon the Untis database. Its structure and consistency could and would not be guaranteed. Instead, it was recommended that Untis XML export

functionality be used to export the required data for scheduling alone, as its format would be guaranteed. This relegated Untis from the strong central position initially envisioned to the position of a data source solely for the THM Organizer component.

3.2. THM Organizer Component

THM Organizer is a two and a half year old component which is responsible for the presentation of schedules, the presentation and management of appointment and events, and since recently for the presentation and management of curricula. The schedules and the reservable resources of appointments and events have as their source the imported XML exports of Untis schedules.

3.2.1. Environment

THM Organizer is a component developed by students and employees of the THM for the Joomla! CMS in the version 2.5.x. As such it requires an installed and active Joomla! 2.5.x CMS and the server environment required by it. THM Organizer requires Untis XML exports for its stock of data, and the THM Organizer content plugin to ensure data consistency of persistent events. Beyond this, there are several other small Joomla! extensions which are dependent upon THM Organizer and supply additional views and functionality.

3.2.2. Security Policy

While THM Organizer's security policy is directly dependent upon the security policy as granted by the Joomla! CMS, there is enough room to allow for a somewhat individual customization of the security policy within the component.

Users and Roles

THM Organizer builds upon the Joomla! platform and functions within a Joomla! website. Correspondent to this, Organizer relies upon the Joomla! user management system. The creation and management of users is administered within Joomla!.

Managed Resources

Within the scope of this thesis the only explicitly managed resources which will be discussed are the schedules imported from Untis. The modeled resources contained within them are implicitly managed at the time of import as will be discussed in the following section pertaining to workflows.

Actions

Joomla! implements an access control list (ACL) security policy which lets administrators dynamically assign which users have which rights upon which resources. Resources in Joomla are structured hierarchially so that every persistent item has a parent. This chain follows from component specific items to component specific categories to the component itself to the root instance equivalent to the Joomla! CMS itself. Each item in this chain may be assigned several different actions, these actions in turn may be assigned to user groups.

THM Organizer's security policy, as regards scheduling, is in comparison extremely simple. It checks, using Joomla! platform functions, whether or not the user can perform the Joomla! administration action on the root resource. Should this be the case the user may then perform any action available in THM Organizer. This simplicity stems from Organizer's development within MNI before Untis' deployment as the central scheduling system. During this time MNI was the only department interested in displaying schedules based upon the XML exports from Untis, and consequently only needed one administrative user, the department schedule coordinator.

Control Mechanisms

The control mechanisms implementing the Joomla!'s and THM Organizer's security policies, while somewhat deviant from those of Untis, fall into the same general pattern.

Login Dialogue The most important control mechanism within Joomla! is the login dialogue, as seen in figure 3.8 on the next page. The user enters his name and his assigned password. This data is then authenticated by the Joomla! framework, and the user is thereafter associated with one or more Joomla! groups. Joomla! users are only required to sign in once to perform any action for which they have access, as the Joomla! framework makes extensive use of persistent session management.

Preemptive Restriction Although Joomla! through its ACL does offer preemptive restriction, THM Organizer does not yet make use of this feature. In Joomla! this removes menu entries for which the user does not have the appropriate rights. However, because Joomla! is a web based content management system this would only remove the display and use of URLs directly from such menu items, it would not be able to block their entry of these URLs into a web browser.

Reactive Restriction As a logical consequence of the limits of preemptive restriction, all Joomla! software must make use of reactive restriction. Within components, this restriction must be implemented at the two levels which can be directly accessed via URLs. The first level is within the component's views,

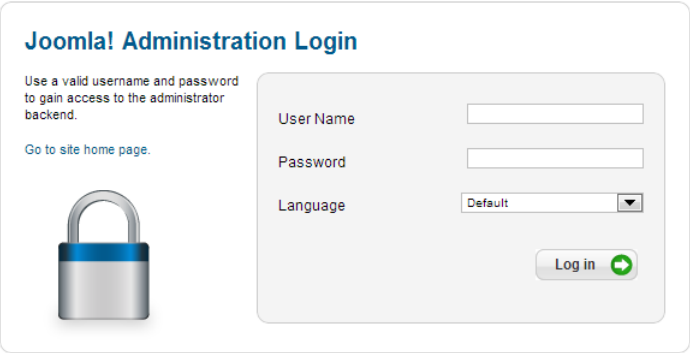


Figure 3.8.: Joomla! Login Dialogue

these display information and can offer actions to the user. The second level is the components controllers, these receive form data from user interaction. These are the more dangerous of the two as these most often call functions of the models which could then modify or destroy data.

A more detailed explanation of THM Organizer's reactive restriction follows in the next chapter discussing the technical implementation.

3.2.3. Workflows

All workflows directly having to deal with scholastic scheduling administered from the THM Organizer by any user which can perform administrative actions upon the Joomla! root resource. As pertains to this thesis the only noteworthy managed resource is the schedule imported from Untis.

Add a New Schedule

The schedule management process begins with the import of a new schedule in THM Organizer. This is performed over the new schedule dialogue as seen in figure 3.9 on the facing page. There the administrator chooses the new plan for import by means of a file selection dialogue, and can add a small commentary which can serve to add remarks to a schedule as to anything noteworthy about the particular schedule.

The upload of a new schedule is the starting point for the following implicit workflows: validation, version management, and delta calculation.

Validation Validation is performed upon upload of a new schedule. It checks the uploaded file for consistency and completeness of data. While Untis does have its own schedule diagnostics tool, this tool cannot account for the structures required within its own data for correct display and usage in THM Organizer.

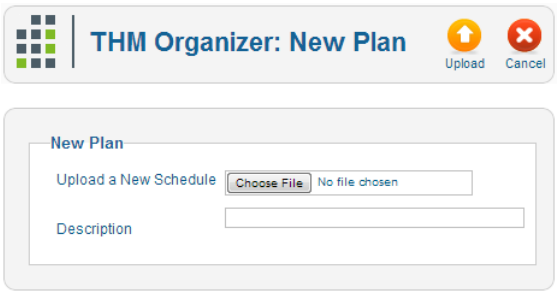


Figure 3.9.: New Schedule Dialogue

Before an uploaded file is saved within THM Organizer it must be checked for references that would create inconsistencies. These come in two varieties: critical errors such as a missing resource long name or missing lesson type, and warnings such as a teacher who has incomplete first name or personnel number data.

Because there can be no guarantee that the export from Untis was first examined by the Untis diagnostic tool, and because the structures must be examined for inconsistencies within Organizer, Organizer's validation also performs many of the same validation checks performed within the Untis diagnostic tool. This checks, among other things, for the presence of items referenced in associations, whether the number of lesson instances matches the number required, and whether all lesson instances have been assigned rooms.

Version Management Version management is performed implicitly with the upload of the exported schedule into THM Organizer. This saves the exported schedules making the various stages of schedule development persistent.

Delta Calculation Upon upload, Organizer automatically sets the uploaded schedule as active within the component for the given organization and planning period. These two attributes are set within the Untis report settings dialogue as discussed in Exporting on page 51. Organizer checks whether another schedule for the uploaded schedule's organization and planning period is already active. Should this be the case, Organizer compares these two schedules and calculates a delta which is saved as a part of the uploaded schedule.

Display The display of schedules in Joomla! is it's highest priority. Students use the displayed schedules for the construction of their personalized schedules. These are created by means of a view implemented as a rich internet application, allowing them to drag and drop lessons from various tabs based upon the schedule from the perspective of a particular resource. While this resource perspective is most often that of a subject group, it can also be according to a field of knowledge, a teacher, room, or subject. This view is displayed in figure 3.10 on the following page. Teachers use this view to check for their schedule for the current semester and to suggest changes which would better fit

both the schedules of their students and their private schedules. Additionally many rooms at the THM are equipped with monitors mounted in front of their doors. These monitors display the lessons and appointments scheduled within the room for the current day, and highlight any lesson or appointment currently being held.

Figure 3.10.: Schedule Rich Internet Application

Set an explicit Reference

In order to highlight changes, which may have occurred in an earlier version, the schedule coordinator may wish to take an earlier version of the schedule as its reference for the delta calculation. This compares the schedules in much the same way as the delta calculation, however lets the schedule coordinator explicitly set which version he wishes to take as a reference. The distinct advantage of such a system is that it allows the schedule coordinator to display a greater number of schedule changes, but at the same time erasing any changes which had been reversed over the course of versions which had come about during the planning of any versions between the reference schedule and the active schedule.

This occurs via the Organizer schedule manager view as displayed in figure 3.11 on the next page. The schedule coordinator chooses a schedule from among the others

saved and sets a check mark in the box to the left. He then clicks on the reference button in the upper tool bar.

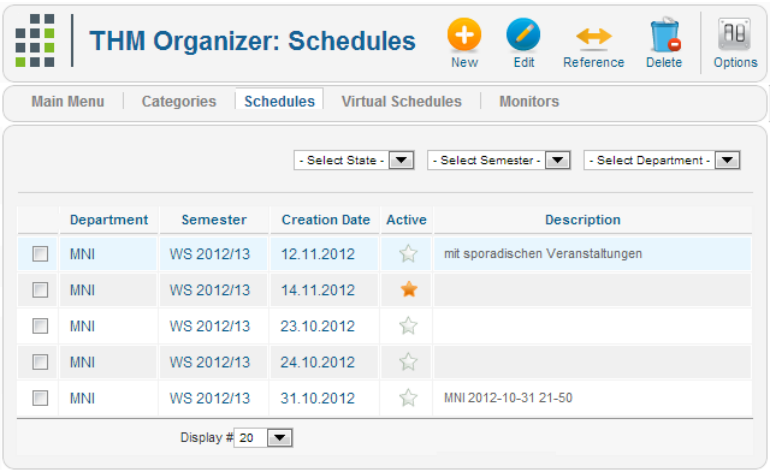


Figure 3.11.: Schedule Manager View

3.2.4. Concurrency

THM Organizer was designed to be used by a single schedule coordinator within a single Joomla! CMS instance. It does however anticipate and avoid concurrency. In contrast to other Joomla! resources which lock resources upon access, Organizer uses transaction based queries when performing data modification, locking access only for the length of an active function call. This also adds the security of commit and rollback actions should any error occur over the length of the transaction.

3.2.5. THM Organizer in the THM Landscape

THM Organizer is at the moment solely used by department MNI, but is under consideration for use throughout the THM. Its greatest strength and most often used functionality is the processing and display of schedules created and exported from Untis. The resources contained within the schedules also serve to provide resources for appointment and event management. This enables in contrast to proprietary private scheduling systems like eGroupware, iCalendar, or Outlook the inclusion and reservation of rooms within the THM and teachers as resources.

This enables the ramification of such appointments and events as regards the schedule to be shown as a part of the schedule. Beyond that, it creates Joomla! content. Here students, teachers and employees can read detailed information about events. Joomla! content can also be subscribed to using RSS feeds which not only keep subscribers informed, but can also be a tremendous help should, for example, a teacher be taken ill. Students which may have other wise had lengthy travel times to come

to the THM for lessons which no longer occur on that day, can then be informed in advance of such cancellations.

4. Technical Realization

4.1. Untis XML Output

As a proprietary system, there was little direct technical implementation within the Untis system. However during the course of the work, contact was closely maintained with Mr. Gruber the managing director of Gruber and Petters. Whenever problems occurred within Untis or additional attributes needed to be added to the Untis XML export, Mr. Gruber had them implemented almost immediately so that indirectly the work at the THM did have an impact upon Untis.

The bulk of the technical implementation in Untis had to do with the modeling of real world structures with the resources available in Untis. For this purpose the structures previously used by MNI in the GP-Untis 2008 version were adapted to those available within the Untis 2013 software. The data available from KMUB and BAU was then fitted to these structures.

Multiuser Untis 2013 opened up many new possibilities and many consequences as a result of, among other things, the modeling the departments as schools and the modeling of temporal behavior. The taxonomical and structural changes are described in this subsection.

As this chapter relates the more technical aspects of the work accompanying the thesis the structure and content of this section is, for the most part based upon the XML export which represents the data contained within Untis schedules to other systems, specifically to THM Organizer.

4.1.1. Taxonomical and Structural Modeling

The terms and structures discussed here are representative of those of GP-Untis 2008 and those of Untis 2013 from the daily version of November 12, 2012. Both exports have, by and large, the same main nodes general, time periods, descriptions, departments, rooms, subjects, teachers, classes (subject groups), students and lessons. Untis 2013 also has the node reductions which, like students in both versions, remains unused. The nodes reductions and students will not be discussed within this section due to their lack of use. In the following, the remaining nodes will be addressed with a short description of their purpose, taxonomical and structural use and any changes that may have been made to them or their subordinate nodes and leaves.

4.1.1.1. General

The general node in the XML structure models data specific to the schedule document itself. In both versions the school name is not editable and gives out the name of the creating program instead of the creating school resource available in Untis 2013.

The header information modeled in the 2008 version reflected the department information as discussed later in this section. This has been changed to model the school name information not given out by the appropriate element.

Both versions output the dates of the school year as input in the school data dialogue. In the 2008 version this information was used for the definition of the valid planning period. In the 2013 version planning periods within an actual school year can be specifically addressed making this information somewhat superfluous.

The output of a more specific planning period in the 2013 version allows the output of the name of the current planning period directly as entered as well as its start and end dates. The input of the planning period name is however not parced within Untis, solely displayed. This makes for a name output with a rather large amount of white space, which then needed to be trimmed on external parce.

4.1.1.2. Time Periods

Time periods correspond to the lesson periods discussed in chapter two. They both remain tragically underused in both the 2008 and 2013 versions. They output the numerical weekday, period, a start and end time.

Within themselves these structures

Although stamped with a reference id, they are nowhere referenced within the rest of the document. Lessons which logically have to be assigned to lesson periods have the exact same information in each child node of their times node. This creates a huge amount of redundancy and bloat within the export.

4.1.1.3. Descriptions

Descriptions are currently used to model fields for teachers, subjects and subject pools, and types for rooms and lessons. Although Untis detects program internal references from resources of these types in the descriptions dialogue, it does not mark the descriptions as being referred to by resources of these types upon export. To avoid having to reprocess this resource while processing the resources which reference it the use of the statistics field, which corresponds to the flags element within the export, has been used to code the types of resources which reference these. This mitigates the lack of explicit association within the resource itself.

The previous structure was solely used to model room types this allowed one time parcing without use of the statistics field. Lesson types were coded within the lessons themselves, which prohibited their direct management. Teacher subject and subject pool fields had not been modeled in the 2008 version.

4.1.1.4. Departments

Currently departments are being used to model the differing CoSs offered by each department. In the previous version they had been used to model organizational hierarchies of resources.

```
<longname>FH, Giessen, MNI, Bioinform.</longname>
```

As we seen in the above example this had a four tier hierarchy of institution, campus, department, and subdepartment. This style of resource modeling had several reasons behind it.

Departments, like descriptions, can be referenced by multiple resource types. In regard to rooms this hierarchy made a organization of responsibility for the room evident. In data built from this information it allowed for the restriction to a particular responsible organization. In regards to teachers and subject pools this allowed the restriction to group affiliation or responsibility for individual CoSs.

Conversely lessons held by external organizations could be modeled in such a way that external teachers and subjects could be marked as such rather specifically. Before modeling departments as schools and the introduction of external resources, this allowed both a visible reservation of shared resources and vague output of department external lessons within rooms which are otherwise the responsibility of that department.

4.1.1.5. Rooms

There have been several modeling changes to rooms, all of which exporting more room information than in the previous version. The structural changes to room modeling now account for more data required by facility management such as the number of seats and some of the room's equipment. These are now exported in the leaves capacity and flags respectively.

The biggest change however is the export of the resource's external name. This newly exported information is both a change in the modeling of the room itself and to the structure of the XML export. This information opens up many new possibilities in THM Organizer. Unfortunately this was first exported with the Untis daily version released November 12, 2012, it precluded any meaningful changes within the scope of this thesis.

4.1.1.6. Subjects

Subjects have received no structural changes within the XML document, but the modeling has changed the way attribute information is exported. The subject's unique identifier, previously exported modeled in the subject field and exported in the corresponding subject group element, has been modeled in the text field and exported in the corresponding node. The subject's field is currently modeled and exported redundantly as a reference in the description element and the more accurate subject group element.

4.1.1.7. Teachers

Teachers have undergone fairly extensive changes. In the newer version the first name can be exported in the forename node. This allowed the last name to be set to its actual value instead of a mix of first and last names as had previously been the case in the event of multiple teachers with the same last name.

Teachers no longer reference departments as previously done to associate them with fields and competence centers. This is now accomplished by use of a description reference, as before there is unfortunately no more accurate node such as the subject group of subjects to export this information. The use of descriptions however provides a manageable external reference for subjects, teachers and subject groups.

Additionally, as with rooms, the external name is now exported. This however has the same caveat that due to its rather late release this information was not yet able to be used in the implementation.

4.1.1.8. Classes (Subject Groups)

The classes node is used in both versions to model subject groups. In both versions the long name is an extended human readable name with the name of the degree program and any criteria which further subset the CoS.

The attempt had been made to model the subsetting criteria as a part of the grade attribute and consequently the class level element in the export. However ass commented upon earlier in chapter 2 this field has a fixed number of characters which made it unusable in this regard. While this information has not been removed and is still exported, the only reliable subsetting information is structured within the long name itself as seen below.

```
Bioinformatik (BSc), 4.  &amp;amp; 5.  Semester, Wahlpflicht
```

In this example we see the structure for the CoS biological information science with for the degree Bachelor of Science for electives in the 4th and 5th semesters. The criteria for subsetting is listed in the order of its hierarchy separated by a comma.

The competence center which designs the CoS and the general field of knowledge taught in this subject group is exported as a reference in the description. While the overall CoS is exported as a department reference.

The teacher was previously modeled within the 2008 version. This information was included in the understanding that this teacher was responsible for the development and planning of the COS. The teacher is in Untis cannot be associated with the any Untis user which makes it unusable in the security policy. This information was also not used for any functionality within THM Organizer, consequently it is no longer modeled.

4.1.1.9. Lessons

Lessons have undergone quite extensive changes both structurally and taxonomically.

IDs and Coupling Rows

A lesson is first defined by the redundant resource identifier "LS_", the actual lesson number and two final digits. These two digits serve to represent the content of the coupling rows within the lesson dialogue. These coupling rows are exported with the same prefix and lesson number as the initial row but increment the last two digits. These rows serve to add additional room, teacher, and subject associations to the first row entered should the lesson be held in multiple rooms, by multiple teachers, or convey multiple lessons.

In the older version the exception to this handling was the resource subject group. In both versions this resource was added in the same row by adding additional subject group names separated from the previous by a comma. It's export in the XML was just as consequent, although a different separator is used. In the export this resource is output as the id attribute of the element lesson classes as shown below. While this is the abuse of what logically should be a single id, it does provide the logical product of the entered information.

```
<lesson_classes id="CL_BI.45.P CL_I.45.W CL_II.45.W
                    CL_MI.45.W"/>
```

In Untis 2013 this functionality is also applied to subjects and rooms, that is to say these may both be entered in the lessons dialogue in the same manner as classes. The teacher resource unfortunately has not yet received this treatment as there were some value calculations dependent upon teachers being listed independently.[1] However the exported schedule treats the new form of information entry the same as a coupling

[1] Every teacher requires his own row. This is important for the clear definition of which teacher is responsible for which subject groups.[Gru12b]

row during the output. This creates, in the best case scenario, a redundancy of almost 100% per coupling row, where all of the lesson's data is presented twice with only the value of one particular element attribute changed.

Temporal Behavior

Temporal behavior is based upon a number of differing attributes within each lesson. The first two leaves which give temporal information are the effective start and end dates. These dates give the boundaries for the lesson dependent upon dates inherited, in order of least to most precedence, from the school year, planning period, lesson group or explicitly set dates within the lesson dialogue.

The next element giving temporal information is the occurrences. This is a sequence of character values which are equal in number to the length in days of the document school year. Each character value has one of three values: 0 for does not occur, 1 for can occur, and F for holiday.

These three elements set up the mold into which the individual lesson instances can be formed.

Individual lesson instances provide additional data for association with individual lesson periods either according to a patterned match or with the explicit output of a specific date. This dichotomy stems from the weekly versus yearly hours provided in the lessons dialogue.

Whether or not a lesson exists with weekly or yearly hours it has the elements assigned day, assigned period, assigned start time, and assigned end time. These four items are in effect redundant information because these four elements must correspond to the four child elements of a specific time period element. However the corresponding time period is not referenced which makes this redundancy necessary in the context of its absence. Lessons which are associated with yearly lesson hours have the additional element assigned date which says explicitly upon what date this lesson instance take place.

In the earlier version, yearly lesson hours as well as intermittent planning were not yet supported. As such, this earlier version had solely the four shared temporal behavior elements within the lesson instance elements.

Descriptions

Lastly, since the beginning of October 2012, Untis exports the lesson description reference. This previously had been modeled against the text field as a simple character value.

4.1.2. Recommended XML Structure

The current XML export structure has grown historically as we have seen from the comparison of the 2008 and 2013 exports. However there is much work that could still be done to eliminate a great deal of redundancy and bloat, make structures more easily read and parced, and conform to XML design consensus. While the recommended structures are explained below a direct comparison can be seen in Appendix B.1 on page 97. There, an excerpt of the actual schedule starting from two lessons and including all resource dependencies extending from them is presented, followed by the same information presented in the recommended structure.

Internal References

Internal references are for the most part made not only specific within the context of their usage, but within the context of the document itself. Document context as used here, means that the references used receive a prefix solely used within the export to identify its resource type. For example, although a teacher may be directly addressed as such in the context of its use, it is also artificially extended with the resource prefix "TR_". In MNI's schedule export these characters account for 8KB of unnecessary file size.

```
<lesson_subject id="SU_OOP"/>
```

The above example shows a typical reference in this case for the subject of a lesson, in this case, object-oriented programming. These references are based upon the resource's name value as discussed in chapter 2. These are stored within Untis at face value, only during the export they are affixed with this artificial resource prefix.

The name values they mask are often the colloquial names used to describe resources in everyday speech. This makes them desirable for display output which must be abbreviated. This necessitates the parcing of the reference values in order to retrieve this value from doubly specific reference values. While not particularly problematic this does make the processing of the hundreds of resources and references a rather inefficient process, and creates bloat in code handling both for the parcing and for the referencing as seen below.

Here the parcing of the subject as a child element of the subjects node:

```
$subjectID = str_replace('SU_', '', $gpuntisID);
```

Here the parcing of the subject reference within the lessons node:

```
$subjectID = str_replace('SU_', '', trim((string)
    $lessonnode->lesson_subject[0]['id']));
```

During the latest changes to the XML structure itself, i.e. from the versions after September 2012, the Untis developers seem to have noticed this logical inconsistency and have started producing output without the double specificity. In the example below the lesson type is addressed directly without the prefix “DS_”.

```
<lesson_description>Ü</lesson_description>
```

Dates and Times

Dates are modeled in Untis using an eight digit unsigned integer value. This occurs corresponds to the ISO 8601 notation YYMMDD, which is a four digit year followed by the numerical month and day with leading zeros.[Dat12] Unfortunately in order to ensure database compatibility, allow comparisons between and perform date algorithms with these values they must first be converted into the standard 'Y-m-d' format. This necessitates a similar parcing of the values as with internal references in order to be able to productively use these values.

Times are modeled using a four digit integer value. This could easily be fixed to allow for the colon, bringing it inline with every conventional time output in the world. If implemented in the Untis export structure without the following changes to the lesson instances this could account for a slight increase in the size of the export. The standardization of data would however more than make up for this small inconvenience.

General

The school name should reflect the school choice in the corresponding Untis dialogue with the school year or planning period name reflecting the active planning period at the time of export. This would reduce the dates output in the general section solely to those necessary within the schedule while reducing user interaction to achieve a more targeted and expected goal.

This would not only ensure the correspondence of the exported data to the entries within the appropriate dialogues, but shrink overhead for the general element by three lines.

Lesson Periods

The current structure creates an enormous redundancy and bloat within the periods. Because Untis periods are created using a grid functionality, they display not only the times entered for each of the individual periods, these are then multiplied by the number of days. The reference id of the period and the value of the period's period element could and should functionally have the same content and be referenced directly by the lesson instances.

What is missing are multiple grids. The assumption is that Untis would simply pack the lesson periods from multiple grids after the lesson periods from the previous.

This would then compound the redundancy without bringing any extra value to the document, alone from the simple fact that the lesson periods are not referenced.

What would help bring clarity to the periods would be first the introduction of the grid resource. The parent element for all time values would then be grids followed by the element grid. This grid would receive a numerical id and have as child elements individual lesson period elements with an id unique within its grid a start time and end time.

This would add two lines of overhead for each grid but reduce overhead to from six lines of code to one for a single period and by a multiplicative factor corresponding to the number of days and periods in the current grid structure.$s = (p * d * 6) - p - 2$ Where s is the savings in lines of code, p the number of lesson periods per day, d the number of planned days per week, and 2 the overhead for a single grid element. The overall benefit for multiple grids would be additive with $s = g^1 + g^2 + ...$ where s is the cumulative savings, $g^1$the savings from grid 1, $g^2$the savings from grid 2, and so on.

Descriptions, Fields, Room Types and Lesson Types

The descriptions field could be replaced with resource specific managed attributes. Each of these would have an id unique to its type and a name. This name need not be the value of a subordinated name element as is currently the case. This can be the value of the field itself. The current statistic field and the corresponding flags element would no longer need to be used and could be removed from the export and from the corresponding Untis dialogues.

This would add two lines of overhead for each of the new resources, less the two that are already existent from descriptions. It would additionally reduce overhead per description from four lines to one. $s = (d * 3) - 4$ Where s is the savings in lines of code, d the current number of descriptions, three the number of lines of code saved per current description and 4 the number of additional lines from the overhead of the three new resources minus the two existing lines from descriptions.

Room Equipment

Currently implemented as the statistics code of the rooms element, room equipment could become it's own element. This would require the implementation of equipment as a resource, but the association should be easily managed as it would be exactly on par with the functionality of the above mentioned resources.

This would create pure overhead in the sense that this resource is not yet managed. However it would allow the management in Untis of a valuable resource type, and in effect completely remove the need for Untis external modeling of rooms by FM. $o = e + 2$ where o is the amount of overhead generated, e the amount of equipment entries, and 2 the overhead for the room equipment element.

Rooms, Subjects and Teachers

Rooms, Subjects and Teachers could have not only their id as an attribute, but also their internal references and fixed attributes. The capacity for room resources could for example be implemented in this way. Their elements with multiple values such as room equipment and with special characters such as names and external names would could continue to be subordinate elements. Room's flags element could be replaced by an appropriate room equipment element referencing any noteworthy equipment to the room. This could have multiple empty space separated values similar to the current implementation of lesson subject groups.

This would result in a minimal savings in overhead for all resources of one to three lines, multiplicative for the number of entries of the respective resource type. It would however also standardize output by placing fixed model attributes without special characters appropriately as attributes rather than as elements, and allow attributes with a series of values or special characters to continue to be appropriately exported as elements.

Subject Groups

Subject groups could have their ids and internal references modeled as attributes with their full names as their element values.

This would reduce overhead per subject group from six lines to one per subject group. $s = g * 5$ where s is the savings in lines of code, g the number of subject groups, and 5 the amount of code saved per entry.

Lessons

Lessons could have the fixed attributes id, period, and grid, as well as the internal reference for the lesson types implemented as attributes. With the associations to teachers, subjects and classes continuing their existence as elements. However additional teachers, subjects, and rooms could be added in the same line separated by an empty space as the current implementation does with subject groups. This would implement the semantic of multiple teachers teaching the same subject or subjects in one or more rooms for students fulfilling the requirements of the same subject groups in tandem and could be implemented in the lessons dialogue by the lack of a coupling row. The inclusion of a coupling row in the lesson dialogue as is yet the only method of input could then make clear the semantic difference intended if desired.

The effective dates and occurrences elements would also be removed, as explained in the next section.

This change would in the worst case scenario of a single teacher or subject have minimal advantages in reduced overhead similar to teacher, subject and room elements. However in the best case scenario of multiple teachers or subjects this would save overhead multiplicatively since it would remove the undesired output of single

lessons as multiple lessons because of forced coupling rows. $s = (t * j * r * l * 10) - 5$ Where s is the savings in lines of code, t the number of teachers, j the number of subjects,r the number of rooms, l the number of lessons, 10 the number of lines of original code, and 5 the number of lines of code after the recommendation.

Lesson Instances

In the current implementation the export leaves the calculation of when lessons actually take place in the hands of the importing system. The implementing system must parce the occurrences element according to both the school year data and the effective begin date and effective end date elements, because the occurrences element has bears no association with the other two elements.

In the case of lessons planned using yearly hours, it also had to add the assigned date element to the lesson instances because it was otherwise impossible to calculate the occurrences of lessons under this implementation.

In order to ensure uniform instance handling and that all data necessary for the calculation of the lesson instance's occurrences is actually within the lesson itself, lesson instances could use an implementation similar to that of the lessons planned using yearly hours. Lesson instances would need to know during which method, yearly or weekly, they were created with, the period in which they take place, the date upon which they take place, and the rooms in which they take place. The first three values could be implemented as lesson instance attributes, with the room references being implemented as a series of references separated by an empty space, as is currently the case with subject groups.

This implementation would reduce the individual instances to one line of code from five. However this reduction in an individual instance code is balanced by the increased number of instances necessary to correctly model instances created using the weekly method. For example, if a lesson using only one teacher, room, and subject takes place twice a week for the entire length of a 16 week planning period this would result in 32 lines of code, one for each occurrence. The corresponding code in the current system would be 17 lines of code with 7 for each instance and three for the occurrences, effective start and end dates within the lesson element itself. In this example that would then be a 15 line increase in overhead. This is however somewhat unrealistic at the THM because over the course of a typical planning period many instances would not occur because of holidays and other organizational reasons.

However, in the case of lessons intended semantically to use more than one teacher, subject, or room using the weekly hours method, this system reduces overhead significantly. In the case of multiple teachers, subjects. or rooms this would amount to a savings of 7 lines per period plus 3 lines for the occurrences and effective dates elements per additional resource used.

In the case of lessons planned using the yearly method this would result directly in a reduction of overhead by reducing the lines of code from eight to one per instance

and removing the three lines of code from the lesson element for the occurrences and effective dates which were superfluous using this method anyways. This is multiplied as with the weekly method in the event of additional teacher, subject or room resources which would have resulted in direct redundancy using the current method.

The most significant effect of this aspect of the recommendation is not the simplification, improved readability, or potential reduction in the lines of code, it is the ability to standardize how this information can be processed.

4.2. A Untis Entity Relationship Proposal

It is difficult to describe the Untis database table structure in a professional manner. Untis had up until the introduction of the MultiUser module been designed to be used as a single user program on a local system. Data was persisted as a local file. Untis Multiuser uses the database for permanent persistence but operates locally with a file generated from this data. This makes Untis backwards compatible to a large extent and allows for newer program versions to add new functionality while using the same system interfaces they have developed over the years. Unfortunately in order to support this local file based program design, the database tables and values continue to represent lines of this locally processed file.

Most of the problems mentioned in dealing with Untis stem from its continued use of these local file based data model. In order to help Untis modify their system architecture with respect to the resources and processed involved an entity-relationship diagram (ERD) has been developed to account for resources and attributes directly involved in scheduling, but excluding to a greater extent those attributes used to generate the visual output, or calculate values.

Accompanying the ERD will be commentary as to which problems the design behind the diagram intends to solve and addresses possible resulting workflow simplifications. The ERDs presented depict resources with a simple box. Typically in ERDs ovals are used to model entity attributes, here these will be listed directly underneath the entity in order to conserve space. Instead of a diamond shape representing relations these are represented using boxes with a double border, this allows them to be modeled in the same way as entities with their principle attributes underneath them. Half circles on lines connecting entities indicate generalization and specialization with the open end towards the more general resource and the curved end toward the more specialized. Where necessary dashed lines have been used to mark associations which had to be diagrammed as crossing the lines of other associations.

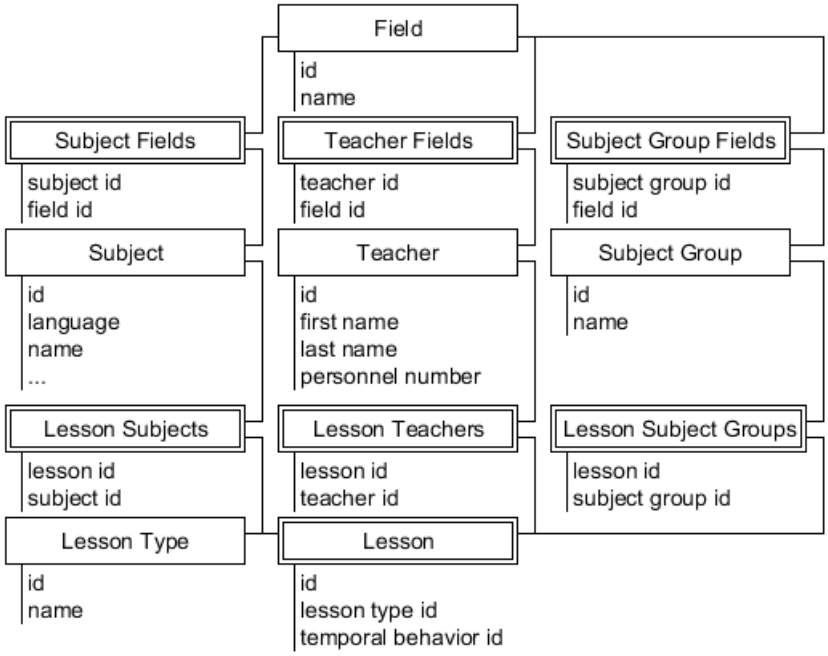


Figure 4.1.: Abstract Lesson

4.2.1. The Abstract Lesson

Lessons exist both are paradox in that they exist as a resource that is the association of multiple other resources. Although lessons realize subjects, the fact that they can under circumstances realize more than one subject necessitates a table depicting the relation between itself and the subjects which it realizes. This precluded the correct symbolism of a semicircle over the line that would theoretically connect the lesson to the subject. Similarly a lesson can be taught by more than one teacher at a time, and may be used for more than one subject group. It could also be said that a lesson implements a particular lesson type. This necessitates no relational table because a given lesson can only implement one lesson type. The name for the lesson as a resource is typically derived from the subject name and the lesson type.

It may seem that the association of subjects and subject groups has been overlooked. While there is a logic to the direct association of the two resources, the association by means of lessons allows for slight variation in the content. This variation, while it doesn't change the subject content enough to warrant a new subject, might be geared more toward students that required to take courses in one subject group as opposed to another. As an example, we will say there is a math lesson. While students of biology and architecture may theoretically have to learn the same basic techniques the examples and calculations involved could be borrowed more from one discipline than the other.

Further fields of knowledge may have multiple subjects and subject groups which deal, for the most part, with integral partial disciplines. Teachers may be versed and certified in multiple fields. Thus necessitating the associative tables with each of these elements.

Figure 4.1 on the previous page depicts a possible model for an abstract lesson, i.e. a lesson which is not concretely associated with a time and place of occurrence.

4.2.2. Temporal Behavior

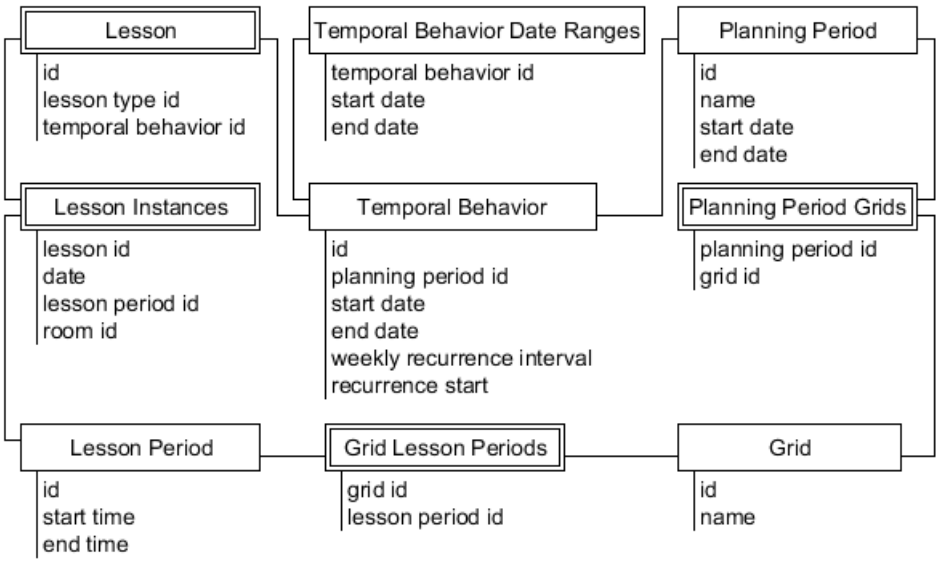


Figure 4.2.: Temporal Behavior

The temporal behavior of lessons was slightly more difficult to model. It involves the aspects of cause and effect. As discussed in chapter 2 a lesson's temporal behavior describes when the lesson is to take place, where the actual assignment of lesson instances to particular lesson periods over the course of a planning period is the result of this behavior.

For this reason the diagram representing the entities involved in this process form a circle. The lesson itself is assigned a temporal behavior, this temporal behavior has the information necessary to describe how the lesson is to be planned over the course of a given planning period. This temporal behavior is the link between a lesson and the planning period for in which it takes place. If additional restrictions are necessary to the shape of the temporal behavior such as direct restrictions to specific ranges of days this information can be modeled using an entity designed for this purpose. This is easily combined with the current method of assignment

according to the weekday only this behavior implied and implemented through the association with actual lesson periods.

The planning period itself references grids through means of an associative table. This allows assignment of multiple grids to a single planning period, and allows these grids to be reused by multiple planning periods.

Each grid is composed of multiple periods. These are referenced through means of the associative table grid lesson periods. The periods here contain minimal redundant data in that one set of periods is used for every scheduled day of the grid's planning periods. This eliminates the redundant data used to store identical start and end times, days and period numbers for every individual day of the week.

Lastly, based upon the temporal behavior, lesson instances are assigned to specific lesson periods. This association is the end product of planning and while, as a result, the assignment is very specific. However is specificity is not sufficient to deduce the attributes of the planning period, temporal behavior, or grid used to plan it.

The proposed entity relationship is depicted in the diagram in figure 4.2 on the facing page.

4.2.3. Rooms and Room Assignment

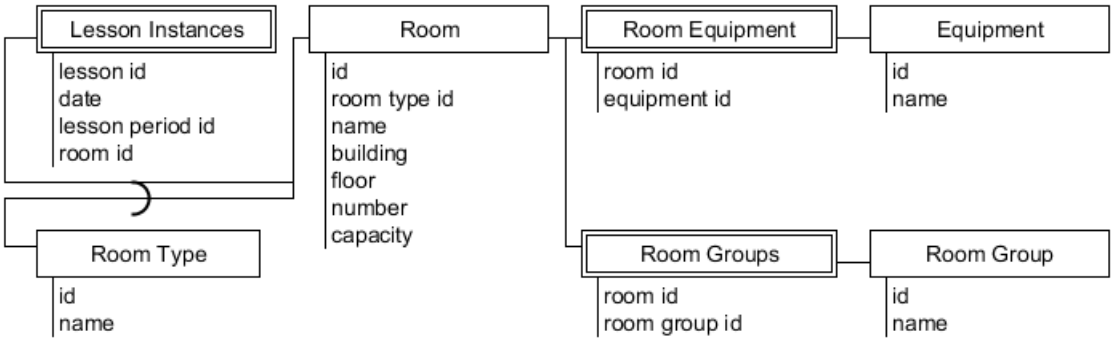


Figure 4.3.: Rooms and Room Assignment

Rooms and their attributes were previously discussed in chapter 2. They require the certain additional attributes currently not directly available in Untis such as the building and floor in which they are situated, or are only rudimentary implemented such as room equipment which is currently modeled by means of the statistics entry.

These rooms are realization of certain archetypes such as an auditorium, workshop, or office. This can be modeled by their direct association with a specific room type. It was worth considering if the room type did not necessarily implicitly include specific equipment. This is by all means certainly the case as it would be hard to

imagine an auditorium without seats or some sort of podium. However room types only leave general notions for more specific equipment such as overheads or video projectors.

It could also be that certain equipment which theoretically could be standard in a specific room type exist only in limited stock within the THM. Should for example a workshop for precision electronics may require a certain welding instrument for the subject to be taught. Should this equipment not be available within all rooms of this type it becomes necessary to take note of this specific equipment in the planning dialogue. Should this be necessary under the current system the schedule coordinator must know in advance which subjects require which special equipment and which rooms have this equipment. He must then create a room group corresponding to this equipment and assign the lesson to be planned within this room group.

The management of room equipment as resource within Untis and its persistence within the database would go a long ways to simplifying the planning process in regards to this criteria. This could then be implemented in the database by a table of equipment and a table associating the equipment to the specific rooms.

Rooms can also exist as an element of multiple room group sets. These sets are defined by the user to group rooms according to their usefulness for certain lesson types and numbers of students. This could also be implemented as a steady resource with its own table and a table associating the room groups with the constituent member rooms.

Lastly rooms are assigned to specific lesson instances based upon their availability, the lesson's temporal behavior, and the required type and equipment. The concrete room or rooms assigned to these lesson instances could have been modeled in one of two different ways.

First the lesson instance could be modeled consisting of a lesson id, lesson period id and date. This could either be a compound id of these three items or a separate id for each row of these items. This would require a second table associating these temporal instances with actual rooms.

The second way and the one depicted in the recommendation is that these three items and the room id be entries in the same table. While, in the case of lessons which take place in more than one room, this would generate some redundancy due to the nature of the association the redundancy would not have been completely removed with the addition of a further associative table which would have had to be indexed according to the lesson id, lesson period id, and date anyways.

The recommended entity relation for this resource model is depicted in figure 4.3 on the previous page.

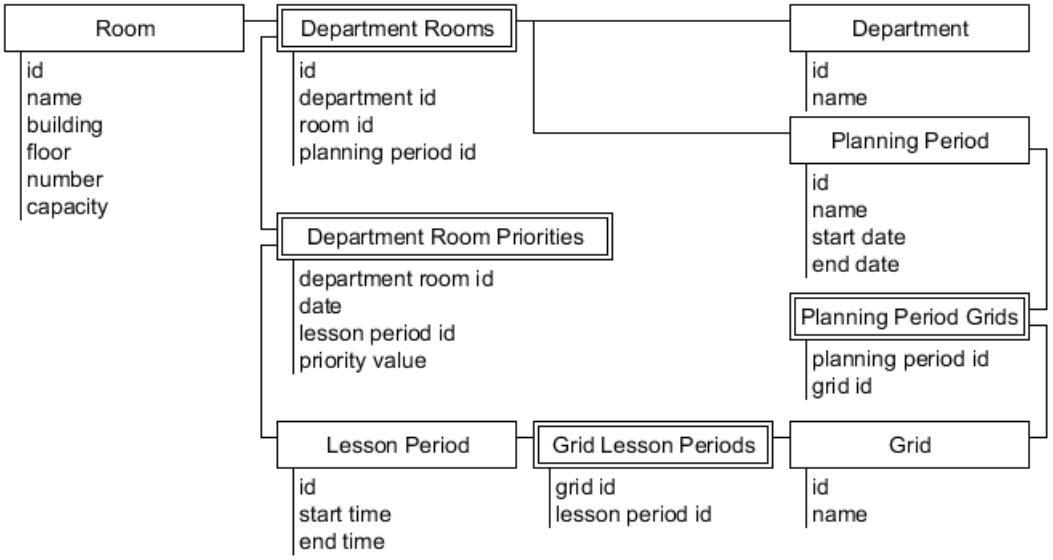


Figure 4.4.: Room Department Association and Planning Priorities

4.2.4. Departments and Planning Priorities

Departments are through their extension in the person of their planning coordinators and directly as a resource an extremely important part of the planning process. They give definition to what is planned by whom and what resources are available in specific planning processes.

In order to accomplish this departments must be able to be associated with the resources. Because resource association can vary between planning periods, the validity of the association is enforced by bringing the planning period into the association.

Because departments rely on individual lesson period planning priorities for resources the association of resource department and planning period is given its own id. This id can then be used to associate this association with a specific lesson period on multiple dates and a value relating the specific planning priority for the period. Similar to the temporal behavior this is set according to a default weekly pattern, however the association with specific dates allows targeted restriction of resources and allows this targeted restriction to be planned around using yearly hours.

For example should a teacher be typically available every Thursday, but knows in advance that he has other obligations on a specific Thursday at some point within the planning period, this need not be modeled by creating a temporal behavior specific to this one lesson. This could be modeled by a restriction for that particular day, allowing the Untis planning algorithm to assign yearly hours instead of weekly

for the absence.

The association of these lesson periods to planning periods over grids is exactly the same as previously mentioned in the section discussing the modeling of temporal data.

An ERD relating the specifics of this model for the room resource can be seen in figure 4.4 on the preceding page.

4.2.5. ERD Summary

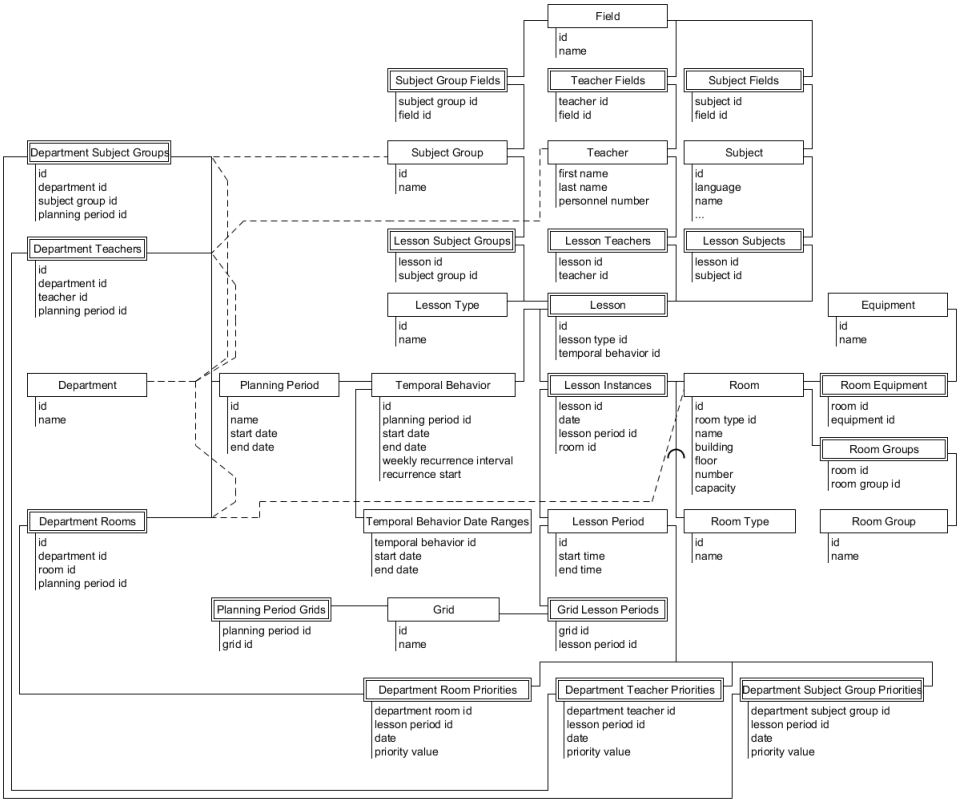


Figure 4.5.: Complete Untis ERD

The proposed restructuring of the database would not only bring about the many advantages of the new implementation methods discussed, it would also make the management of the database itself much easier for the Untis developers. MS SQL and MySQL (using the InnoDB engine) support foreign keys and cascading of data. They allow transaction based queries to enable consistency checks and table restoration.

This would remove much of the program code responsible for data aggregation and propagation as this would then be automatically carried out by the algorithms of the database management system itself. The deletion of a planning period for example

could cascade far enough to delete all of the associations with grids, remove subordinate temporal behavior, departmental allotments, and departmental planning priorities.

It would ensure that the data provided locally reflects the actual data stored within the database in concurrency situations.

Figure 4.5 on the preceding page displays the recommendation as described in the previous sections in its completeness.

4.3. THM Organizer

The existing workflow of the THM Organizer Component in regards to schedules started with the export of a schedule from the Untis system. The initial hope with the acquisition of Untis 2013 for the THM was that direct management of existing resources would be made possible. This would have eliminated data inconsistency not only between Untis and external systems, but also enabled external data propagation and management within the Untis system. As a practical example it would have enabled teacher first names and personnel numbers to be directly propagated through the Untis database tables so that all scheduling coordinators were on the same stand.

Sadly the tables used by the Untis system are still in a state of transition and the Untis representatives strongly recommended that this approach be avoided, not only for data creation and manipulation, but also for read access. Instead recommending that we rely on the data exported through the XML export.[2]Although less than optimal and a great disappointment after the initial high hopes for clean data, it was conceded that this would continue to be the start point for the schedule workflows within the THM Organizer component for the foreseeable future.

The redesign and refactoring of THM Organizer to use the new XML export structures and data formed the greater part of the accompanying technical implementation.

[2] Our database structure is not documented. During the migration of Untis software to newer versions, the structure of the database may change at any time. Interfaces, which access the database directly, may as a consequence no longer function. This cannot occur using the XML structure. We implement changes to the XML structure as additions to the existing structure, meaning all the expected elements will always be there although new elements may appear over time. This ensures a compatibility to both older and newer interfaces.[Gru12b]

4.3.1. Reactive Control Mechanisms

As discussed in section about reactive restriction within Joomla! on page 55, THM Organizer relies heavily upon reactive restriction to enforce its security policy.

Within views reactive restriction can take place in two possible ways. User access can be checked upon view access. This is optimal to protect the display of sensitive information, and eliminate extensive business logic before it is even performed. Should the user not be allowed to perform a certain action or actions the code being processed can simply be stopped.

User access can also be restricted in views while constructing the toolbar. This method allows fine tuned user access methods to the resource involved, but also allows the display of information itself as a potential access method, similar to the Joomla! view access levels construct. This would allow for example a user who is allowed to create but not edit items access to the new button, but stop the performance of the code which would have rendered the edit button.

Of these two THM Organizer uses the first method because of the complex nature of the information to be displayed, and the current lack of a refined security policy defining the access rights of multiple users.

Within controllers reactive restriction is checked upon function access, checking the access in much the same way as the toolbar building in the component views. This prevents unauthorized user interaction by means of URL manipulation or the developer tools available for most common browsers.

4.3.2. XML Modeling with the SimpleXML Library

The import of schedules into THM Organizer occurs using the XML export prepared by Untis. XML is not natively readable within PHP. Multiple libraries exist for this purpose the one chosen for use within the scope of this thesis was the SimpleXML library. First the XML document must be converted into a SimpleXMLElement using the code below.

```
$xmlSchedule =
simplexml_load_file($_FILES['file']['tmp_name']);
```

Once created the objects can be addressed in several different ways corresponding to PHP standard methods for other objects. As the method above delivers a SimpleXMLElement object. This corresponds to a tag in the XML syntax, in the case of the example above the document tag.

Many tags will have important attributes written within the tag itself. The access to attributes requires both the element being processed and the attribute names to be handled in a similar manner to array indexes in PHP. The element itself must be

addressed with the zero index, with the appropriate attribute name as the second index as seen below.

```
$creationDate = trim((string) $xmlSchedule[0]['date']);
```

The information must first be converted to one of the PHP value types in order to be saved as a variable. Here the trim() function is used to remove any additional white space stored within the value.

Nodes themselves can be accessed in a similar manner to object properties in standard PHP by means of the object operator ->.

```
$departmentname = trim((string)
$xmlSchedule->general->header1);
```

In this example above the general tag, a tag subordinate to the document tag is first accessed. This would deliver a further SimpleXMLElement. Going further the `header1` tag subordinate to the general tag is accessed. Since this element has no children the object operator makes reference to the value of the element and not the element itself. Here the name of the department which produced the schedule is retrieved. As in the previous example this is first converted to a string and stripped of white space.

The last major function necessary for this implementation is the iterator. Several times an element will have multiple subordinate elements with the same tag name, such as teacher, room, or time. In order to process such subordinate elements the iterative function of the SimpleXML library must be used. This works much the same as the standard PHP iterator.

```
foreach ($xmlSchedule->subjects->children() as
                $subjectnode)
```

This example displays the manner in which the individual subjects are addressed for processing. First the subordinate subjects element is accessed. All the subordinate elements of the subjects element are individual subjects. As no differentiation can be made amongst the children while all have the same name they must be iterated over for further processing.

These methods were then the building blocks for the creation of structures necessary building blocks for the building of structures which made the rest of the workflows possible.

4.3.3. Version Management and Persistence

In previous versions of THM Organizer the contents of the XML files were made persistent as individual table entries. The schedule data and resources were stored in their own respective tables building an entity relation model similar but much simpler than the one recommended in section 4.2.

The first idea for the persistence of schedule data before the recommendation to rely solely on the XML export was that there be almost no persistent data. Data should be retrieved dynamically from the central database. As this was recommended against the idea was to use the data contained within the XML to fill the then current entity relationship model. This too was ruled out because of the implementation of the three dimensional schedule model and the implementation of yearly hours, both of which were impossible under the previous model.

This left two options the restructuring of the current THM Organizer model to fit the new schedule information, or the creation of a completely new structure. In the end the second option was chosen. This allowed for the restructuring of the entire component making it much lighter. The idea was that only the schedules needed to be directly managed and that these would be made persistent in the form of strings holding JSON data.

This implementation allows for a completely dynamic structuring of the resources imported from Untis. The structure is, however, loosely based upon the structure of the imported XML. This structure could then be saved to the database in place of the previous persistence of the actual XML documents. From the previous tables then only the schedules, teachers, teacher fields, rooms and room types needed to be made available in the database. Schedules for the presentation of the results of scholastic scheduling, and the others so that they could be used within the appointment and event management functionality of the component.

The uploaded schedules then form defacto a very rudimentary form of version management. The organization, the planning period, and the date of export allowing insight into the development of the schedule over time. The dynamic structure of the JSON strings allows the changes between various local copies of the schedule to be calculated and stored as a part of the schedule itself.

The persistence of the schedules as JSON string also had other advantages. Their content can be directly decoded into PHP, the language in which Joomla! code is written. This allows their direct translation into the structures required for further use within the system.

Also the JSON format was native to the scheduling output view implemented as a rich internet application within Organizer. This program is largely JavaScript based and allowed the structures created to be put directly into use. This allowed an ease of transition from the old system to the new system within a matter of days.

4.3.4. Validation and Delta Calculation

The implementation for the validation of the imported XML and the calculation of the delta between the schedule being imported and the schedule currently marked as active for the given semester rely on the same structures. The two processes feeding seamlessly into one another.

Using the SimpleXML tools discussed in section 4.3.3, THM Organizer tries to create PHP objects to model the data retrieved from the XML structure. It follows the structure of the XML document for the most part, first creating the data needed to be referenced in later sections and then later checking the references to ensure that the data referenced exists.

The biggest difference between the structure of the XML document and that of the PHP structure modeling it is the calendar property. As previously mentioned in multiple places in the thesis, lesson occurrences are not directly written into the XML document in its current format. Instead these occurrences must be calculated by iterating over the course of the effective data elements expressed as a substring of the massive occurrences element. This process is extremely cumbersome and entirely unnecessary. To avoid having to perform this calculation every time the actual date of a lesson is asked this information is calculated once at the time of validation.

Each resource is checked to ensure that the information has been entered according to the guidelines expressed in chapter 2. At each stage of the process a decision has to be made if an inconsistency is a cause for the validation to be terminated, or this is information which the user uploading the schedule should later bring in line with the internal standard.

This firstly ensures that all the time information is syntactically and semantically valid according to the information entered. Secondly it creates a structure with extremely low redundancy which is easily read. Each date of the term is created as an index in a calendar element, with the periods occurring on that day acting as indexes for that particular date. Further the lessons scheduled to occur within each period are applied as indexes with the scheduled rooms further used as indexes within the lesson. The use of indexes instead of values allowed for a later ease of marking during the development of the delta.

The calculation of the delta compares partial structures in one plan and another. This occurs in several places at different levels in the structural hierarchy.

First the lessons are compared directly according to the keys assigned to them by Untis. Should a lesson key exist in one plan or the other the lesson is marked as new or added to the schedule and marked as removed according to the perspective of which schedule is to later be marked active.

Then the abstract lesson's resources are examined subjects, teachers, and subject groups. Should these deviate from one schedule to another the resource itself is marked as new or removed and the lesson itself is marked as changed.

Lastly the concrete lesson's occurrences and room assignments are checked. This occurs within the previously discussed calendar element of the created structure. Every date and period are checked for lesson assignment. Should a lesson occur in one schedule and not the other this occurrence is marked as new or added to the schedule and marked as removed. Lastly the lesson occurrence's rooms are inspected should the assigned rooms vary the variance for each room is marked as new or removed, and the lesson occurrence is marked as changed.

For a look at the complete structure of the THM Organizer schedules see appendix B.2 on page 106.

5. Conclusion

5.1. Lessons Learned

Throughout the modeling within Untis and the refactoring of THM Organizer to accommodate the new structures several lessons have been learned regarding system design, user interaction, and system interfacing.

5.1.1. Untis

The service offered by the Gruber & Petters software house has been top notch. They interest themselves for bugs and feature requests, often bringing smaller requests into the next daily build. Mr. Gruber himself has always handled the e-mails with efficiency, even routing them directly to the Untis developers when he personally is unavailable.

Untis support of the data from older versions of the software is key in making the system user friendly, to the extent that it allows users of previous Untis systems to import instead of manually adding all the data needed then needed to model the schedules. Even now this data can be used to directly create two dimensional planning within Untis as the use of hierarchical planning periods, weekly periodicity, and temporal behavior is not forced. Each of these things can be used as the user requires them.

The import of such data is as stated above indispensable, however, that the program still relies internally on antiquated data structures hinders more than helps the continued development in the direction required by true multiuser functionality. The current implementation relies upon the distributed management of resource data and distributed planning. The distributed management of resource data has inconsistency among entered data between Untis schools and stale data read actions where multiple users work within the same school as a consequence.

Untis' greatest strength is it's powerful planning algorithm. It not only produces almost useable plans when correctly configured, it also recognizes intrinsically rareness of teacher fields and rooms and by extension room types. This recognition and the ability to model the resources of the various departments would and could allow the extension of Untis functionality to cover the needs of FM and HRM as concerns resource modeling, eventually allowing Untis itself to grow into the campus management system it has the ability to become.

This development has the requirements of clean data management, and the strengthening of the administrative aspect to this software developed to be used completely decentralized. This would include centralized management of resource data and attributes. The decentralized departments would have to act as data consumers picking which available resources should be used within their view of resources, so that the consistency of data could be centrally assured. This could even be functionally supported in perpetuity by giving the users a dialogue to import existing resource settings from a previous planning period.

The design principles behind Untis often reflect a top-down or schedule-centered perspective. This enables users to figuratively jump into planning by entering resource information directly from the resource's use in associations. While this may have a less steep learning curve and initially offer much in the way of user comfort, the sometimes lacking support of bottom-up planning can prove slightly frustrating to advanced users. Bottom-up planning enables resource administrators to operate from a resource or data-centered perspective first entering all the required data then working down the chain of dependency to create functionally rich, meaningful, and detailed schedules.

There need not however be a dichotomy between these two perspectives. Instead Untis developers could open a new dialogue related to the detailed resource view when resources are created dynamically. This would keep the fast top-down perspective centered user interaction in the foreground, but make efforts to ensure that the required data is properly entered instead of just creating a resource based upon a name-key.

5.1.2. THM Organizer

THM Organizer still has far to go. Its development has only taken place over the course of the last three years and has evolved from a front-end module for the upload of XML documents to a very powerful tool for the handling of scheduling, events, resources, and curricula. It has much work left to be done to standardize its place in coordination with the Untis program and any future campus management system. Should Untis not evolve to accept more data and have a cleaner database oriented handling of more assets THM Organizer may end up developing even more in this direction.

While its distant future may, for now, remain uncertain its immediate future as discussed in section 5.3 is bright and highly functional.

5.2. Development Limitations

During the conception of this thesis it was originally intended that Untis' database would serve as a stable source of data. This would have allowed the implemen-

tation of a web services platform for the propagation of information both within and without Untis. This direct propagation would have been a source of accurate real time data usable in multiple systems, principally THM Organizer, but there were also plans in the works for access for campus navigation and other consumer applications.

Mr. Gruber the managing director of Gruber & Petters recommended that this approach not be implemented as Gruber & Petters cannot guarantee the information structure and quality of the Untis database at this time. This had the consequence of a much more limited interaction between Untis and other systems within the THM landscape.

The time necessary for acclimatization to and modeling in Untis and the iterative development of Untis structures in response to our criteria had at certain points applied the brakes to development in that a minimal wait time was at times necessary for the necessary structure elements to be exported. This led to a bit of the accordion effect as the development of structures within THM Organizer are completely dependent on those offered by Untis.

For this reason and the general time limitation imposed upon the scope of a master thesis, there is much that remains to be developed within THM Organizer.

5.3. Directions for Future Development

The limitations to the development in THM Organizer were until now only restricted by the amount of time available for investment into the system. However this restriction soon no longer stands in the way and there is much to be done, such as:

- resolving resources across schedules imported for many different departments
- merge of department schedules for the creation of aggregate schedules both between departments and for the entire school
- resource management protocols when uploading a schedule so as to allow the selection of a specific resource actualization
- delta calculation between aggregate plans
- data synchronization methods through targeted exporting
- possible imports of generated files modeling existing resource information from differing sources
- functions designed to support the calculations and queries requested by FM
- interfaces to and import of data from other university resource management systems, most notably the HIS family

5.4. Summary

In the overview in chapter one several questions have been raised these have been explicitly and implicitly answered over the course of the thesis. In this final section these questions will be posed again this time with brief answers drawing on the conclusions drawn in the previous chapters.

- What are the resources to be managed?

The resources managed within the scope of this thesis are schedules and the resources modeled within them. Schedules themselves are managed in tandem by departmental schedule coordinators and representatives of Facility Management. These schedules are valid for the run of a particular planning period.

Principle among them is the lesson itself an association of several other resources. Lessons are the realization of subjects. They are taught by teachers to fulfill the requirements of CoSs. Subjects and CoSs are special perspectives of a particular field of knowledge for which the teacher or teachers are trained experts. Lessons occur during specific lesson periods over the course of a planning period according to their associated temporal behavior.

- Who manages them?

The resources are managed by the schedule coordinators of the respective departments and members of facility management. These people correspond to users associated with roles and groups. These in turn are assigned actions which can be performed upon the managed resources.

- What are the typical actions performed with them?

The actions are system dependent and aggregate to form workflows. Untis is principally responsible for data aggregation, the planning of schedules, and interdepartmental visibility of resource usage. It has many predefined actions to achieve these ends such as administration, export, and data management.THM Organizer is responsible for the validation, versioning, persistence, and web based display of schedule information.

- What data is required by each system?

Untis requires comprehensive data over all modeled resources to create a complete and informative schedule. Output and planning even as far as generic placeholders can be scheduled with far less information.

THM Organizer requires the schedule information included within the export from Untis inclusive of all the resources listed above.

- What functionality should be offered by which system?

The palate of functionality offered by Untis can be increased and decreased by the purchase or omission of extension modules. The required functionality is that which is necessary to fulfill the actions described above. Key among the necessary modules

is the MultiUser module which allows for asynchronous remote data aggregation and planning by the various departments.

Organizer requires the upload and persistence of the schedules exported by Untis. A key tool for transparency is the versioning process which hold the various standpoints of the schedules imported into Organizer and can analyze and compare them to calculate and display any changes made.

- How should this functionality be implemented?

Untis functionality, as a proprietary system, cannot be directly influenced except through the way in which data is modeled in the various resources. However, the Gruber and Petters software company is very appreciative of suggestions and is exerts a constant effort to ensure quality within their product.

THM Organizer's functionality is implemented using a variety of tools offered by various systems. Chief among those are the PHP core libraries, the Joomla! platform, MySQL, the SimpleXML library, and the ext-js JavaScript framework.

Thank you for reading.

Acknowledgments

Many thanks to those that have supported me on my way these last few years: my girlfriend and my daughter for their love, Professor Kneisel for his mentorship and brainstorming, my colleagues at webMedia for their camaraderie and laughter, and Mr. Christ, Mr. Pfaff and Mr. Foerster for their IT support.

A. Appendix

A.1. Room Types

The complete list of all codified room types in use at the THM.

A1	Hörsaal / Seminarraum, 1 - 20 Plätze
A2	Hörsaal / Seminarraum, 21 - 33 Plätze
A3	Hörsaal, 34 - 47 Plätze
A4	Hörsaal, 48 - 69 Plätze
A5	Hörsaal, 70 - 84 Plätze
A6	Hörsaal, 85 - 105 Plätze
A7	Hörsaal, 106-130 Plätze
A8	Hörsaal, 131-185 Plätze
A9	Hörsaal (gr), 186-270 Plätze
AZ	Hörsaal (gr), 271-400 Plätze
B1	Fachsaal, Physik-Hörsaal (kl)
B2	Fachsaal, Physik-Hörsaal (gr)
B3	Fachsaal, Chemie
C1	Fachsaal, Praktikumsraum mit. spez. Einbauten
C2	Fachsaal, Praktikumsraum keine spez. Einbauten
C3	Fachsaal, Forschungslabor
C4	Fachsaal, Mess-Stand
C5	Fachsaal, Nebenlabor
C6	Fachsaal, Vorbereitungsraum
C9	Fachsaal, Betriebsraum
D1	Rechnerraum, allgemein
D2	Rechnerraum, fachspez. Raum
D3	Rechnerraum, fachspez. (Forschung)

D4	Rechnerraum, Peripherie- / Geräteraum
D5	Rechnerraum, Serverraum Raum
D7	Rechnerraum, Schulung
F1	Projektraum, allgemein Raum
F2	Projektraum, Gruppenarbeitsraum
F3	Projektraum, Übungen
F5	Projektraum, Studiebüro Raum
F6	Projektraum, Sozialraum Raum
H1	Sonstige Räume, Archiv
H2	Sonstige Räume, Seminar-Nebenraum
H3	Sonstige Räume, sonstiges Lager
H4	Sonstige Räume, Labor Lager
H5	Sonstige Räume, Abstellraum
I1	Büro, Büro
I2	Büro, Labor-Ingenieur
I3	Büro, Werkstatt
I4	Büro, Besprechungen Raum
I5	Büro, Ergänzungsraum
I6	Büro, Videokonferenzen
W1	Werkstatt, Feinmechanik
W2	Werkstatt, Metall
W5	Werkstatt, Elektronik
W7	Werkstatt, Nebenraum
W9	Werkstatt, Lager

A.2. Fields

BAU	Bauwesen
INF	Informatik
ING	Ingenieurwesen
ING-INF	Ingenieurwesen / Informatik
JUR	Jura

LIN	Linguistik
MAT	Mathematik
MAT-INF	Mathematik / Informatik
MAT-NAT	Mathematik / Naturwissenschaften
MED	Medizin
MED-INF	Medizin / Informatik
MED-ING	Medizin / Ingenieurwesen
MED-NAT	Medizin / Naturwissenschaft
NAT	Naturwissenschaft
NAT-INF	Naturwissenschaften / Informatik
NAT-ING	Naturwissenschaften / Ingenieurwesen
SOZ	Sozialwissenschaften
WIR	Wirtschaft
WIR-INF	Wirtschaft / Informatik
WIR-ING	Wirtschaft / Ingenieurwesen
WIR-JUR	Wirtschaft / Jura
WIR-MAT	Wirtschaft / Mathematik

B. Source Code

B.1. Comparison of XML Structures

Listing B.1: The actual structure of the XML files.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<document version="1.4" date="20121119" time="122354">
  <general>
    <schoolname>Untis 2013</schoolname>
    <schoolyearbegindate>20121008</schoolyearbegindate>
    <schoolyearenddate>20131005</schoolyearenddate>
    <header1>MNI</header1>
    <header2>WS 2012/13</header2>
    <footer>WS2012   </footer>
    <termbegindate>20121008</termbegindate>
    <termenddate>20130324</termenddate>
  </general>
  <timeperiods>
    <timeperiod id="TP_13">
      <day>3</day>
      <period>1</period>
      <starttime>0800</starttime>
      <endtime>0930</endtime>
    </timeperiod>
    <timeperiod id="TP_15">
      <day>3</day>
      <period>3</period>
      <starttime>1130</starttime>
      <endtime>1300</endtime>
    </timeperiod>
    <timeperiod id="TP_16">
      <day>3</day>
      <period>4</period>
      <starttime>1400</starttime>
      <endtime>1530</endtime>
    </timeperiod>
    <timeperiod id="TP_25">
      <day>5</day>
      <period>1</period>
      <starttime>0800</starttime>
      <endtime>0930</endtime>
    </timeperiod>
    <timeperiod id="TP_26">
```

```
39        <day>5</day>
40        <period>2</period>
41        <starttime>0950</starttime>
42        <endtime>1120</endtime>
43      </timeperiod>
44      <timeperiod id="TP_29">
45        <day>5</day>
46        <period>5</period>
47        <starttime>1545</starttime>
48        <endtime>1715</endtime>
49      </timeperiod>
50      <timeperiod id="TP_30">
51        <day>5</day>
52        <period>6</period>
53        <starttime>1730</starttime>
54        <endtime>1900</endtime>
55      </timeperiod>
56      <timeperiod id="TP_32">
57        <day>6</day>
58        <period>2</period>
59        <starttime>0950</starttime>
60        <endtime>1120</endtime>
61      </timeperiod>
62      <timeperiod id="TP_33">
63        <day>6</day>
64        <period>3</period>
65        <starttime>1130</starttime>
66        <endtime>1300</endtime>
67      </timeperiod>
68      <timeperiod id="TP_34">
69        <day>6</day>
70        <period>4</period>
71        <starttime>1400</starttime>
72        <endtime>1530</endtime>
73      </timeperiod>
74      <timeperiod id="TP_35">
75        <day>6</day>
76        <period>5</period>
77        <starttime>1545</starttime>
78        <endtime>1715</endtime>
79      </timeperiod>
80    </timeperiods>
81    <descriptions>
82      <description id="DS_NAT">
83        <longname>Naturwissenschaft</longname>
84        <flags>F</flags>
85      </description>
86      <description id="DS_NAT-INF">
87        <longname>Naturwissenschaften / Informatik</longname>
88        <flags>F</flags>
89      </description>
90      <description id="DS_SOZ">
```

```
91        <longname>Sozialwissenschaften</longname>
92        <flags>F</flags>
93      </description>
94      <description id="DS_C1">
95        <longname>Fachsaal, Praktikumsraum mit. spez. Einbauten</
              longname>
96        <flags>R</flags>
97      </description>
98      <description id="DS_D1">
99        <longname>Rechnerraum</longname>
100       <flags>R</flags>
101     </description>
102     <description id="DS_SU">
103       <longname>Seminar. Unterricht</longname>
104       <flags>U</flags>
105     </description>
106     <description id="DS_P">
107       <longname>Praktikum</longname>
108       <flags>U</flags>
109     </description>
110   </descriptions>
111   <departments>
112     <department id="DP_BI">
113       <longname>Bioinformatik (BSc)</longname>
114     </department>
115     <department id="DP_TRMD">
116       <longname>Technische Redaktion &amp;amp; Multimedia Dokumentation<
              /longname>
117     </department>
118   </departments>
119   <rooms>
120     <room id="RM_A20.210">
121       <longname>A20.2.10</longname>
122       <capacity>28</capacity>
123       <flags>OB</flags>
124       <room_description id="DS_D1"/>
125       <external_name>A20.2.10</external_name>
126     </room>
127     <room id="RM_C10.630">
128       <longname>C.6.30</longname>
129       <capacity>24</capacity>
130       <text>Chemie</text>
131       <room_description id="DS_C1"/>
132       <external_name>C10.6.30</external_name>
133     </room>
134   </rooms>
135   <subjects>
136     <subject id="SU_Chem2">
137       <longname>Chemie 2</longname>
138       <text>MN101</text>
139       <subjectgroup>NAT</subjectgroup>
140       <subject_description id="DS_NAT"/>
```

```
141     </subject>
142     <subject id="SU_TRMD">
143       <longname>TRMD</longname>
144       <subjectgroup>SOZ</subjectgroup>
145       <subject_description id="DS_SOZ"/>
146     </subject>
147   </subjects>
148   <teachers>
149     <teacher id="TR_Feld">
150       <forename>Doris</forename>
151       <surname>Feldrappe</surname>
152       <payrollnumber>hg9290</payrollnumber>
153       <teacher_description id="DS_NAT"/>
154       <external_name>FeldD</external_name>
155     </teacher>
156     <teacher id="TR_Tyne">
157       <forename>Tjani</forename>
158       <surname>Tyne</surname>
159     </teacher>
160   </teachers>
161   <classes>
162     <class id="CL_BI.2">
163       <longname>Bioinformatik (BSc), 2. Semester</longname>
164       <classlevel>2. Semester</classlevel>
165       <class_description id="DS_NAT-INF"/>
166       <class_department id="DP_BI"/>
167     </class>
168     <class id="CL_TRMD">
169       <longname>Techn. Red. &amp;amp; Multimed. Doku.</longname>
170       <classlevel>TRMD</classlevel>
171       <class_description id="DS_SOZ"/>
172       <class_department id="DP_TRMD"/>
173     </class>
174   </classes>
175   <students/>
176   <lessons>
177     <lesson id="LS_252200">
178       <periods>1</periods>
179       <lesson_subject id="SU_Chem2"/>
180       <lesson_teacher id="TR_Feld"/>
181       <lesson_classes id="CL_BI.2"/>
182       <effectivebegindate>20121015</effectivebegindate>
183       <effectiveenddate>20130201</effectiveenddate>
184       <occurence>000000
             F111111F111111F111111F111111F111111F111111F111111F111111F111111F11111FFF
             </occurence>
185       <lesson_description>P</lesson_description>
186       <times>
187         <time>
188           <assigned_day>3</assigned_day>
189           <assigned_period>4</assigned_period>
190           <assigned_starttime>1400</assigned_starttime>
```

```
191           <assigned_endtime>1530</assigned_endtime>
192           <assigned_room id="RM_C10.630"/>
193         </time>
194       </times>
195     </lesson>
196     <lesson id="LS_285500">
197       <periods>14</periods>
198       <lesson_subject id="SU_TRMD"/>
199       <lesson_teacher id="TR_Tyne"/>
200       <lesson_classes id="CL_TRMD"/>
201       <effectivebegindate>20121008</effectivebegindate>
202       <effectiveenddate>20131005</effectiveenddate>
203       <occurence>111111
             F111111F111111F111111F111111F111111F111111F111111F111111F111111F11111FFFFFF
             </occurence>
204       <lesson_description>SU</lesson_description>
205       <times>
206         <time>
207           <assigned_day>3</assigned_day>
208           <assigned_period>1</assigned_period>
209           <assigned_starttime>0800</assigned_starttime>
210           <assigned_endtime>0930</assigned_endtime>
211           <assigned_date>20121107</assigned_date>
212           <assigned_room id="RM_A20.210"/>
213         </time>
214         <time>
215           <assigned_day>3</assigned_day>
216           <assigned_period>3</assigned_period>
217           <assigned_starttime>1130</assigned_starttime>
218           <assigned_endtime>1300</assigned_endtime>
219           <assigned_date>20121107</assigned_date>
220           <assigned_room id="RM_A20.210"/>
221         </time>
222         <time>
223           <assigned_day>3</assigned_day>
224           <assigned_period>4</assigned_period>
225           <assigned_starttime>1400</assigned_starttime>
226           <assigned_endtime>1530</assigned_endtime>
227           <assigned_date>20121107</assigned_date>
228           <assigned_room id="RM_A20.210"/>
229         </time>
230         <time>
231           <assigned_day>5</assigned_day>
232           <assigned_period>1</assigned_period>
233           <assigned_starttime>0800</assigned_starttime>
234           <assigned_endtime>0930</assigned_endtime>
235           <assigned_date>20121109</assigned_date>
236           <assigned_room id="RM_A20.210"/>
237         </time>
238         <time>
239           <assigned_day>5</assigned_day>
240           <assigned_period>2</assigned_period>
```

```
241           <assigned_starttime>0950</assigned_starttime>
242           <assigned_endtime>1120</assigned_endtime>
243           <assigned_date>20121109</assigned_date>
244           <assigned_room id="RM_A20.210"/>
245         </time>
246         <time>
247           <assigned_day>5</assigned_day>
248           <assigned_period>5</assigned_period>
249           <assigned_starttime>1545</assigned_starttime>
250           <assigned_endtime>1715</assigned_endtime>
251           <assigned_date>20121109</assigned_date>
252           <assigned_room id="RM_A20.210"/>
253         </time>
254         <time>
255           <assigned_day>5</assigned_day>
256           <assigned_period>6</assigned_period>
257           <assigned_starttime>1730</assigned_starttime>
258           <assigned_endtime>1900</assigned_endtime>
259           <assigned_date>20121109</assigned_date>
260           <assigned_room id="RM_A20.210"/>
261         </time>
262         <time>
263           <assigned_day>6</assigned_day>
264           <assigned_period>2</assigned_period>
265           <assigned_starttime>0950</assigned_starttime>
266           <assigned_endtime>1120</assigned_endtime>
267           <assigned_date>20121110</assigned_date>
268           <assigned_room id="RM_A20.210"/>
269         </time>
270         <time>
271           <assigned_day>6</assigned_day>
272           <assigned_period>3</assigned_period>
273           <assigned_starttime>1130</assigned_starttime>
274           <assigned_endtime>1300</assigned_endtime>
275           <assigned_date>20121110</assigned_date>
276           <assigned_room id="RM_A20.210"/>
277         </time>
278         <time>
279           <assigned_day>6</assigned_day>
280           <assigned_period>4</assigned_period>
281           <assigned_starttime>1400</assigned_starttime>
282           <assigned_endtime>1530</assigned_endtime>
283           <assigned_date>20121110</assigned_date>
284           <assigned_room id="RM_A20.210"/>
285         </time>
286         <time>
287           <assigned_day>6</assigned_day>
288           <assigned_period>5</assigned_period>
289           <assigned_starttime>1545</assigned_starttime>
290           <assigned_endtime>1715</assigned_endtime>
291           <assigned_date>20121110</assigned_date>
292           <assigned_room id="RM_A20.210"/>
```

```
293         </time>
294         <time>
295           <assigned_day>3</assigned_day>
296           <assigned_period>1</assigned_period>
297           <assigned_starttime>0800</assigned_starttime>
298           <assigned_endtime>0930</assigned_endtime>
299           <assigned_date>20121031</assigned_date>
300           <assigned_room id="RM_A20.210"/>
301         </time>
302         <time>
303           <assigned_day>3</assigned_day>
304           <assigned_period>3</assigned_period>
305           <assigned_starttime>1130</assigned_starttime>
306           <assigned_endtime>1300</assigned_endtime>
307           <assigned_date>20121031</assigned_date>
308           <assigned_room id="RM_A20.210"/>
309         </time>
310         <time>
311           <assigned_day>3</assigned_day>
312           <assigned_period>4</assigned_period>
313           <assigned_starttime>1400</assigned_starttime>
314           <assigned_endtime>1530</assigned_endtime>
315           <assigned_date>20121031</assigned_date>
316           <assigned_room id="RM_A20.210"/>
317         </time>
318       </times>
319     </lesson>
320   </lessons>
321   <reductions/>
322 </document>
```

Listing B.2: The recommended structure of the XML files.

```
1  <?xml version="1.0" encoding="ISO-8859-1"?>
2  <document version="1.4" date="20121119" time="122354">
3    <general>
4      <schoolname>MNI</schoolname>
5      <planning_period>WS2012</planning_period>
6      <begindate>20121008</begindate>
7      <enddate>20130324</enddate>
8    </general>
9    <grids>
10       <grid id="1">
11         <period id="1" starttime="08:00" endtime="09:30"/>
12         <period id="2" starttime="09:50" endtime="11:20"/>
13         <period id="3" starttime="11:30" endtime="13:00"/>
14         <period id="4" starttime="14:00" endtime="15:30"/>
15         <period id="5" starttime="15:45" endtime="17:15"/>
16         <period id="6" starttime="17:30" endtime="19:00"/>
17       </grid>
18   </grids>
19   <fields>
```

```
20      <field id="NAT">Naturwissenschaft</field>
21      <field id="NAT-INF">Naturwissenschaften / Informatik</field>
22      <field id="SOZ">Sozialwissenschaften</field>
23    </fields>
24    <room_types>
25      <room_type id="C1">Fachsaal, Praktikumsraum mit. spez. Einbauten
           </room_type>
26      <room_type id="D1">Rechnerraum</room_type>
27    </room_types>
28    <lesson_types>
29      <lesson_type id="SU">Seminar. Unterricht</lesson_type>
30      <lesson_type id="P">Praktikum</lesson_type>
31    </lesson_types>
32    <room_equipment>
33        <equipment id="B">Beamer</equipment>
34        <equipment id="O">Overhead</equipment>
35    </room_equipment>
36    <rooms>
37      <room id="A20.210" type_id="D1" capacity="28">
38        <name>A20.2.10</name>
39        <room_equipment>O B</room_equipment>
40        <external_name>A20.2.10</external_name>
41      </room>
42      <room id="C10.630" type_id="C1" capacity="24">
43        <name>C.6.30</name>
44        <text>Chemie</text>
45        <external_name>C10.6.30</external_name>
46      </room>
47    </rooms>
48    <subjects>
49      <subject id="Chem2" field_id="NAT" subject_no="MN101">Chemie 2</
           subject>
50      <subject id="TRMD" field_id="SOZ">TRMD</subject>
51    </subjects>
52    <teachers>
53      <teacher id="Feld" personnel_no="hg9290" field_id="NAT">
54        <forename>Doris</forename>
55        <surname>Feldrappe</surname>
56        <external_name>FeldD</external_name>
57      </teacher>
58      <teacher id="Tyne">
59        <forename>Tjani</forename>
60        <surname>Tyne</surname>
61      </teacher>
62    </teachers>
63    <module_pools>
64      <module_pool id="BI" field_id="NAT-INF">Bioinformatik (BSc)</
           module_pool>
65      <module_pool id="BI.2" parentid="BI" field_id="NAT-INF">2.
           Semester</module_pool>
66      <module_pool id="TRMD" field_id="SOZ">Techn. Red. &amp;amp; Multimed
           . Doku.</module_pool>
```

```
67  </module_pools>
68  <lessons>
69    <lesson id="2522" type_id="P" periods="12" grid="1">
70      <lesson_subjects>Chem2</lesson_subjects>
71      <lesson_teacher>Feld</lesson_teacher>
72      <lesson_module_pools>BI.2</lesson_module_pools>
73      <instances>
74        <instance method="w" period="4" date="2012-10-17">C10.630</
              instance>
75        <instance method="w" period="4" date="2012-10-24">C10.630</
              instance>
76        <instance method="w" period="4" date="2012-10-31">C10.630</
              instance>
77        <instance method="w" period="4" date="2012-11-07">C10.630</
              instance>
78        <instance method="w" period="4" date="2012-11-14">C10.630</
              instance>
79        <instance method="w" period="4" date="2012-11-21">C10.630</
              instance>
80        <instance method="w" period="4" date="2012-11-28">C10.630</
              instance>
81        <instance method="w" period="4" date="2012-12-05">C10.630</
              instance>
82        <instance method="w" period="4" date="2012-12-12">C10.630</
              instance>
83        <instance method="w" period="4" date="2012-12-19">C10.630</
              instance>
84        <instance method="w" period="4" date="2013-01-16">C10.630</
              instance>
85        <instance method="w" period="4" date="2013-01-23">C10.630</
              instance>
86      </instances>
87    </lesson>
88    <lesson id="2855" type_id="SU" periods="14" grid="1">
89      <lesson_subjects>TRMD</lesson_subjects>
90      <lesson_teachers>Tyne</lesson_teachers>
91      <lesson_module_pools>TRMD</lesson_module_pools>
92      <instances>
93        <instance method="y" period="1" date="2012-10-31">A20.210</
              instance>
94        <instance method="y" period="3" date="2012-10-31">A20.210</
              instance>
95        <instance method="y" period="4" date="2012-10-31">A20.210</
              instance>
96        <instance method="y" period="1" date="2012-11-07">A20.210</
              instance>
97        <instance method="y" period="3" date="2012-11-07">A20.210</
              instance>
98        <instance method="y" period="4" date="2012-11-07">A20.210</
              instance>
99        <instance method="y" period="1" date="2012-11-09">A20.210</
              instance>
```

```
100         <instance method="y" period="2" date="2012-11-09">A20.210</
                instance>
101         <instance method="y" period="5" date="2012-11-09">A20.210</
                instance>
102         <instance method="y" period="6" date="2012-11-09">A20.210</
                instance>
103         <instance method="y" period="2" date="2012-11-10">A20.210</
                instance>
104         <instance method="y" period="3" date="2012-11-10">A20.210</
                instance>
105         <instance method="y" period="4" date="2012-11-10">A20.210</
                instance>
106         <instance method="y" period="5" date="2012-11-10">A20.210</
                instance>
107       </instances>
108     </lesson>
109   </lessons>
110 </document>
```

B.2. THM Organizer Schedule Structure

The following is the structure of the THM Organizer schedules. These have been removed of all data to allow a better examination of the underlying structure. The removed data has been replaced by place holders which give hints as to the data's general type or the resource id it represents. Where repetitious elements, which did not help to further understand the structure, have been removed ",..." has been added. Optional structural information, specifically information pertaining to delta calculation, has been marked in *[square braces]*.

Listing B.3: The implemented schedule structure in THM Organizer.

```
1  {
2      "creationdate":"<DATE>",
3      "creationtime":"<TIME>",
4      "departmentname":"<TEXT>",
5      "semestername":"<TEXT>",
6      "startdate":"<DATE>",
7      "enddate":"<DATE>",
8      "periods":{
9          "<#>":{
10             "gpuntisID":"TP_<#>",
11             "day":<#>,
12             "period":<#>,
13             "starttime":"<TIME>",
14             "endtime":"<TIME>"
15         },...
16     },
17     "fields":{
18         "<field id>":{
19             "gpuntisID":"DS_<field id>",
```

```
            "name":"<TEXT>"
        },...
    },
    "roomtypes":{
        "<room type id>":{
            "gpuntisID":"DS_<room type id>",
            "name":"<TEXT>"
        },...
    },
    "lessontypes":{
        "<lesson type id>":{
            "gpuntisID":"DS_<lesson type id>",
            "name":"<TEXT>"
        },...
    },
    "degrees":{
        "<degree id>":{
            "gpuntisID":"DP_<degree id>",
            "name":"<TEXT>"
        },...
    },
    "rooms":{
        "<room id>":{
            "gpuntisID":"RM_<room id>",
            "name":"<room id>",
            "longname":"<TEXT>",
            "capacity":"<#>",
            "description":"<room type id>",
            "typeID":"<#>"
        },...
    },
    "subjects":{
        "<subject id>":{
            "gpuntisID":"SU_<subject id>",
            "name":"<subject id>",
            "longname":"<TEXT>",
            "subjectNo":"<TEXT>",
            "description":"<field id>"
        },...
    },
    "teachers":{
        "<teacher id>":{
            "gpuntisID":"TR_<teacher id>",
            "surname":"<TEXT>",
            "firstname":"<TEXT>",
            "username":"<TEXT>",
            "description":"<field id>",
            "fieldID":"<#>"
        },...
    },
    "modules":{
        "BI.1":{
```

```
72              "gpuntisID":"CL_<subject group id>",
73              "name":"<subject group id>",
74              "longname":"<TEXT>",
75              "restriction":"<TEXT>",
76              "degree":"<degree id>",
77              "description":"<field id>"
78          },...
79      },
80      "calendar":{
81          "sylength":<#>,
82          "termlength":<#>,
83          "<DATE>":{
84              "<#>":{
85                  "<lesson id>":{
86                      "<room id":"<delta>",...[,]
87                      ["delta":"<delta>"]
88                  },...
89              },...
90          },...
91      },
92      "lessons":{
93          "<lesson id>":{
94              "gpuntisID":"LS_<#>",
95              "subjects":{
96                  "<subject id>":"<delta>",...
97              },
98              "description":"<description id>",
99              "name":"<TEXT>",
100             "teachers":{
101                 "<teacher id>":"<delta>",...
102             },
103             "modules":{
104                 "<subject group id>":"<delta>",...
105             },
106             "comment":"<TEXT>"[,]
107             ["delta":"<delta>"]
108         },...
109     }
110 }
```

C. E-Mail Sources

C.1. Peter Kneisel

Unsere Hochschule hat sich in den letzten 10 Jahren wesentlich weiterentwickelt:

- Die Studierendenzahl ist von von unter 10000 auf über 14000 Studierende gestiegen.
- Neue Standorte in ganz Hessen wurden erschlossen – mit Wetzlar ein Standort mit über 1000 Studierenden.
- Neben der Lehre ist als zweites und drittes Standbein die Forschung und die Weiterbildung hinzugekommen
- Alle Studiengänge haben auf Bachelor/Master umgestellt. • Mit über 50 Studiengängen hat sich die Anzahl der Studiengänge fast verdoppelt.
- Mit Studium+ ist ein eigener berufsbegleitendes Lehrkonzept mit mittlerweile über 1000 Studierenden entstanden.

Tatsächlich ist in den letzten zehn Jahren keine Stein auf dem anderen geblieben. Allein – unsere Systemlandschaft ist nicht mitgewachsen:

- Die Studierendenverwaltung wird im HIS-SOS-System umgesetzt.
- Die Notenverwaltung findet in den meisten Fachbereichen (außer MNI) in HIS-POS statt.
- Die Verwaltung von Modulen findet entweder gar nicht oder unvollständig in HIS-LSF statt
- Die ECollaborations-Funktionalität wird in eStudy und – jetzt – in Moodle abgebildet
- Die Stundenplanung erfolgt in den unterschiedlichsten Systemen, eines davon ist Untis.
- Die Intranet-Funktionalität ist auf die vielfältigen Websites der Fachbereiche und der eGroupware der Verwaltung verteilt
- Die Internet-Funktionalität ist mit unterschiedlichen Techniken auf die dezentralen und zentralen Organisationen verteilt

- daneben gibt es noch weitere dezentrale Systeme für die Bibliothek, Raumverwaltung, Dokumentenverwaltung, ...

HIS, als ein großes, zentrales System ist veraltetet und soll 2014 abgelöst werden. Ein zentrales Campus-Mangement-System ist als Ersatz geplant, wird aber die vielfältigen Anforderungen aller benötigten Funktionen nicht vollständig erfüllen können. Wir benötigen also eine System-
landschaft, in der möglichst viele Funktionalitäten sinnvoll zusammengefasst werden und dort, wo das nicht möglich ist, sauber Schnittstellen entwickelt werden.

Diese Herausforderung sind wir in den vergangenen Jahren mit dem iCampus-Projekt angegangen und so hat sich zwischenzeitlich aus den vielen unterschiedlichen Systemen, die sich, oft mit jeweils denselben, Funtkionalitäten parallel entwickelt haben, ein zwar hetereogenes aber doch (weitgehend) einheitliche System-
landschaft entwickelt:

- Die Studierenden- und Notenverwaltung wird nach wie vor in den HIS-Systemen vorgenommen
- HIS-LSF entwickelt sich zur zentrale Datenhaltung für Modul-
beschreibungen und verfügt über Schnittstelen zu deren Dar-
stellung
- Moodle hat sich als eCollaborations-Plattform durchgesetzt.
- UniUntis entwickelt sich zur zentrale Datenhaltung für die Stundenplanung an der THM (ohne Studium+) und verfügt über Schnittstellen zu deren Darstellung
- Joomla (2.5) entwickelt sich zur alleinigen Plattform für Internet und - zunehmend - auch Intranetfunktionalität und wird um Schnittstellen zu allen wesentlichen Systemen erweitert

Eine wesentliche Funktionalität, die in keinem bislang untersuchten Campus-Management-System (z.B. "Daten-Lotse") mit hinreichender Funktionalität vorhanden ist, ist die Stundenplanung. Hier hat sich als Untis als, bislang dezentral eingesetztes Werkzeug über viele Jahre bewährt. Untis verwaltet die stundenplanspezifischen Objekte (Curricula, Module, Fächer, Dozenten, Räume, Semester-
Gruppen) und hat dabei natürlich auch Schnittstellen zu anderen Systemen, die teilweise dieselben ressourcen betrachten. Auch die für die Stundenplanung typischen Prozesse haben inahltliche und zeitliche Verknüpfungen mit anderen hoschulspezifischen Prozessen.

C.2. Peter Gruber

- Sie haben empfohlen, die Untis Schulen als unsere Fachbereiche einzusetzen. Dies hat sich durchaus bewährt wegen den schul- und periodenbezogenen Zeitwünschen und Zuständigkeiten. Wie sind Sie darauf gekommen?

"Zum einen war es die Erfahrung (Sie sind nicht die erste FH, die auf Untis umsteigt) - auch bei anderen FHs haben die einzelnen Fakultäten die Bedingung, unterschiedliche Zeitwünsche einzugeben. Zum anderen war es die Datenmenge. Wenn Sie alles in einer Schule hätten, hätten wir u.U. Performance-Probleme."

- Sie haben empfohlen, dass wir die gleichen Kurznamen verwenden wie externe Namen. Ist das nur wegen der Eindeutigkeit und automatisierten Pflege, oder gibt es einen technischen Hintergrund?

"In Untis sind die Kurznamen die IDs der Elemente - wenn gleiche Elemente in unterschiedlichen Datenbeständen unterschiedliche IDs haben, muss man bei jedem Datenaustausch darauf Rücksicht nehmen. Das gilt übrigens auch für WebUntis, wenn die Daten dort eingespielt werden, ist es notwendig, die unterschiedlichen Kurznamen per Alias-Namen zusammen zu fassen. Grundsätzlich können wir mit unterschiedlichen Kurznamen aber umgehen."

- Bei den Zeitwünschen haben Sie empfohlen, dass wir die gröberen Zeitperioden wie Vormittags, Nachmittags, oder den Tag selbst mit Werten versehen anstatt die einzelne Stunden mit Werten zu versehen. Außer der Bequemlichkeit von mehrfachem Markieren gibt es vielleicht eine technisches Vorteil dieser Vorgehensweise?

"Der Grund liegt in der Bequemlichkeit."

- Ich erinnere mich vage dass gesagt wurde das Dozenten, bezüglich Kopplungszeilen, nicht die gleiche Handhabung wie Fächer, Klassen und Räume haben weil da irgendwelche Werteberechnungen durchgeführt werden so dass die Dozenten ihre eigene Zeile benötigen. Können Sie dies bestätigen oder erinnere ich mich falsch?

"Korrekt, jeder Dozent benötigt seine eigene Zeile. Das ist wichtig, weil damit klar definiert ist, welcher Dozent für welche Gruppen zuständig ist. Semestergruppen und Räume können jeweils mit Komma getrennt eingegeben werden."

- Sie haben empfohlen dass wir nicht direkt auf die Datenbank zugreifen sondern uns auf die XML-Export Datei verlassen sollen. Können Sie bitte dazu ein paar Sätze schreiben?

"Die Datenbank wird von uns nicht dokumentiert, bei einer neuen Version kann sich daher die Struktur der Datenbank jederzeit

ändern. Schnittstellen, die direkt auf die Datenbank zugreifen, funktionieren daher bei einem Releasewechseln u.U. nicht mehr. Das kann bei der XML Schnittstelle nicht passieren, Änderungen passieren dort nur in der Form, dass Felder dazukommen (was wiederum "alte" Schnittstellen nicht stört). Eine Kompatibilität zu neuen Versionen ist daher von Vornhereingewährleistet.”

Bibliography

[Ame12a] American Heritage Dictionary. Definition of subject, November 2012. http://www.ahdictionary.com/word/search.html?q=subject.

[Ame12b] American Heritage Dictionary. Definition of version, November 2012. http://www.ahdictionary.com/word/search.html?q=version.

[Com08] Choosing between MS-Access, MSSQL or MySQL databases !, March 2008. http://forums.eukhost.com/f31/choosing-between-ms-access-mssql-mysql-databases-4530.

[Dat12] Date formats, November 2012. http://www.php.net/manual/en/datetime.formats.date.php.

[fSA12] Ministry for Science and Art. Staatsministerin Kühne-Hörmann gibt Startschuss für Kooperation bei Ausbildung und Forschung in Medizininformatik und Medizintechnik, August 2012. http://www.hessen.de/irj/hessen_Internet?rid=HStK_15/hessen_Internet/sub/4f2/4f2704ef-96d2-5931-79cd-aa2b417c0cf4,,,11111111-2222-3333-4444-100000005004%26_ic_uCon_zentral=4f2704ef-96d2-5931-79cd-aa2b417c0cf4.htm.

[Gru12a] Christian Gruber. E-Mail from 20 November, 2012. Appendix B.2, November 2012.

[Gru12b] Christian Gruber. E-Mail from 23 November, 2012. Appendix B.2, November 2012.

[Hel07] Martin Heller. REST and CRUD: the Impedance Mismatch, January 2007. http://www.infoworld.com/d/developer-world/rest-and-crud-impedance-mismatch-927.

[Kne12] Peter Kneisel. E-Mail from 13 November, 2012. Appendix B.1, November 2012.

[PED12a] PEDAV das Softwarehaus für Schulen GmbH & Co. KG. *Das Handbuch zu Untis Multiuser*. PEDAV das Softwarehaus für Schulen GmbH & Co. KG, Mai 2012.

[PED12b] PEDAV das Softwarehaus für Schulen GmbH & Co. KG. *Das Handbuch zu Untis Multiuser*. PEDAV das Softwarehaus für Schulen GmbH & Co. KG, Mai 2012.

[PED12c] PEDAV das Softwarehaus für Schulen GmbH & Co. KG. *Das Handbuch zu Untis Multiuser*. PEDAV das Softwarehaus für Schulen GmbH & Co. KG, Mai 2012.

[PED12d] PEDAV das Softwarehaus für Schulen GmbH & Co. KG. *Das Handbuch zu Untis Multiuser*. PEDAV das Softwarehaus für Schulen GmbH & Co. KG, Mai 2012.

[W3C12] SQL Data Types, November 2012. `http://www.w3schools.com/sql/sql_datatypes.asp`.

[Wik12] Comparison of relational database management systems, November 2012. `http://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems`.

Nomenclature

ACL	Access Control List
BAU	Construction (Bauwesen)
CMS	Content Management System
CoS	Course of Studies
CRUD	Creation, Reading, Updating, and Deletion
ERD	Entity-Relationship Diagram
FM	Facility Management
HRM	Human Resources Management
KMUB	Hospital and Medicinal Techniques, Environmental and Biological Technology (Krankenhaus- und Medizintechnik, Umwelt- und Biotechnologie)
MNI	Mathematics, Natural and Information Sciences (Mathematik, Naturwissenschaften und Informatik)
QM	Quality Management
SuK	Social and Cultural Sciences (Sozial- und Kulturwissenschaften)
THM	University of Applied Sciences Central Hessen

Declaration of Authorship

I hereby declare that I have written this thesis without any help from others and without the use of documents and aids other than those stated. I have mentioned explicitly all used sources of information and I have cited them correctly according to established academic citation rules.

Place, date	Signature