ABSTRACT
Scheduling of lectures at a large university or college is a complex and specialized task with many rules but no ideal solution and no formal model and quality measure for the result. This prevents the use of any automated scheduling system, requiring each lecture to be scheduled by humans. This paper describes the MetroNG application—a visual interactive tool for multiparametric multidimensional scheduling, which we created to address the complex issues of scheduling at universities.

Categories and Subject Descriptors
H.5.2 [Information Interfaces and Presentation]: User Interfaces—graphical user interfaces (GUI); I.3.6 [Computer Graphics]: Methodology and Techniques—interaction techniques; J.1 [Administrative Data Processing]: Education

General Terms
Design, Human Factors

Keywords
interactive scheduling, data visualization

1. INTRODUCTION
The main problem when creating a schedule of university lectures lies in checking and preventing collisions—one teacher should not teach two lectures at the same time, students should not visit more than one lesson at the same time, etc. Moreover, there is a lot of soft or hard constraints, e.g. a teacher cannot teach in a particular time period or students are not happy to be taught continually 12 hours in a day. Therefore, creating a ‘good’ schedule is a tough task; some IT support for the whole scheduling process is required. Unfortunately, no algorithms for creating a good schedule can be created since no formal criteria and evaluation function exist. The creation of a schedule is so dependent on the vast experience and knowledge of the people involved, that a formal model of the whole scheduling process would take unreasonable time and resources to create and would probably be already outdated by the time it is finished.

In this paper we present a tool for supporting the whole process of creating a schedule. It uses a complex collision detection system, which is used to identify situations where a group of students is attending two classes scheduled for the same time. Such complex systems are required when scheduling lectures at a university, since—unlike lower levels of education—the groups of university students are very loosely defined. There is an M:N relation between groups and students compared to 1:N at the lower levels.

The rest of the paper is organized as follows: The next section describes the issues that are addressed by the tool, Sec. 3 describe the interactive layer of the application and shows some commented screen-shots demonstrating capabilities of the application. The final sections evaluate and compare the results to other relevant projects.

2. PROBLEM DESCRIPTION
The main purpose of the MetroNG application is to display the schedule and allow the users to modify it. The visualization part must cope with the huge amount of information present in the schedule. An average schedule of a college or university contains hundreds or thousands of events, spans several buildings with tens or hundreds of lecture rooms, and handles hundreds of teachers and many thousands of students. Since it is too much information to fit on one screen, the visualization must be highly interactive and provide the user with only that part of information that he or she is interested in at the moment.

Although the modification of the schedule is basically a simple drag-and-drop action, it is a part of a larger, very complex problem. While the whole decision making is left to the user (the user picks the time and room for each event), it is not sufficient to give the users the ability to schedule the events. To do the task efficiently, they need a lot of information beyond “what, when and where”. The most important information is whether they are creating a collision-free schedule, i.e. whether they are not asking someone to be at two different places at the same time.

2.1 Dimensions
The schedule is displayed as a two-dimensional grid. Traditionally, one axis corresponds to the days of the week and the other to the time of day. We never use this view in
MetroNG since it is too wasteful to use both axes for time. Instead, MetroNG supports several views that combine different dimensions and conform the most usual use cases. The dimensions used in these views include:

- **time** – the day of the week and time of day displayed as one linear ruler-like display. For an example, see the top part of the Figure 1. Note that it displays three lines – the day, the hour and the number of lecture (e.g., lecture no. 1 starts at 7:20 AM).
- **space** – the rooms where the events can be placed. The axis is structured into a three layer tree: status (“private” or “public”), buildings, rooms.
- **teachers** – the teachers that take part in the events (i.e., teach the lectures). A three layer tree structure is also used: sections – departments – teachers. This way, the user can group and hide/unhide the most relevant records for his particular decisions.
- **students** – the groups of students that attend the events. The groups are identified using *attributes* and naturally partially ordered by inclusion. The hierarchy of groups is presented to the user as a tree structure, allowing interactive selection of the displayed level of detail.

### 2.2 Collisions

A collision is a state when two events are scheduled in such a way, that someone supposed to attend both events would be unable to do so. There are four basic types of collisions: student, teacher, transfer, and space/time.

A student collision means that a group of students that attend an event are attending another event at the same time. The teacher collision is similar – one of the teachers that takes part in an event is teaching another lesson at that time. A transfer collision means that although the students or teachers attend no other event at that time, they attend an event in a different building before or after the time in question and would not be able to transport to the other building in time. Finally, a space/time collision means that there exists a time period when at least two events occupy the same room.

The first case (student collision) is the most difficult one. The problem is identifying which two events can be attended by the same students. There are two parts to this problem. First, the students are assigned to groups, but the structure of the groups is complex – formally, the set of groups is partially ordered by inclusion. In other words, the groups form a directed acyclic graph. For instance, there is a group of all students in the first year and a group of all IT students. But there is also a group of first year IT students, which is included in both of those groups.

The second part of the problem is similar; each scheduled event has a list of student groups that attend it. But the group hierarchy plays a role in this assignment as well. If the data states that first year students group attend a certain class, then the first year IT students group attend it as well. Put together, to check whether two events scheduled to the same time create a student collision, one cannot just check the list of student groups associated to each event to see if there is a same group in both lists. It is necessary to deduce whether there is any group of students that could attend both events based on the group hierarchy. For instance, if one of the events is for first grade students and the other for IT students, they do create a collision since there is a group of first grade IT students.

### 3. VISUALIZATION

The MetroNG system consists from two complimentary application. The MetroNG web interface provides support actions for the system and offers web access to the schedule for students and teachers. This web application will not be discussed further in this article. The second application is the MetroNG thick client application for the members of the scheduling committee and it is the subject of this paper.

After the startup, the main screen of the MetroNG application (see the Figure 1) is displayed. Apart from several dialog windows, this is the interface that the users work with. The menu is used mostly for configuration while the toolbar below it allows the user to switch the available modes. The layout and contents of the rest of the window depend on the currently selected mode.

#### 3.1 Application Modes

MetroNG supports several application modes; each mode de-facto displays several dimensions of the data and it is tailored for a specific class of events. The most common type of events is a regular lecture taught every week of the semester at the same time. The regular modes (S and L) display these regular events using a days-of-the-week time dimension as the X axis. These modes are used most of the time and an example is displayed in the Figure 1.

Another type of events is a block-oriented lectures; all working days in one week are dedicated to a single lecture for a particular group of students. This type of events is commonly used for practices and clinical education. The block mode (B) supports this type of events by using the weeks of the semester as the X axis. Each week is displayed as a single column – the day of the week and the time of the day is ignored.

T and TL modes are used for irregularly taught lectures, which are scheduled as set of one-time (not repeated) events. These modes allow the users to display schedule for each week individually (see the Figure 2). The views look almost identical to S and L modes, but the interpretation of what the user sees is a bit different – if a rectangle (an event) is displayed in one of the regular modes, it means that the event takes place at the corresponding time in some of the weeks (usually all of them) of the semester. If the same rectangle is displayed in T or TL mode, it means that the event is scheduled for the corresponding time of the selected week. If two events overlap in the rooms area in one of the regular modes, it does not necessarily mean that there is a collision in the schedule, since for instance one event is scheduled only for the first week and the other is scheduled for the fourth week. On the other hand, if this occurs in the T mode, it means that the events are scheduled into the same room, for the same time withing the same day of the same week (i.e. the same data) – there is a time/space collision.

These are the most important events and views used for the actual scheduling. There are three special-purpose application modes used for maintenance work and a lot of other events in the system, e.g. room reservations not related to lectures, students’ busy time, teachers’ preferences etc. Although these events are not directly maintained by MetroNG (they are maintained by the above mentioned MetroNG web application), they are displayed in relevant views so that users have all available information for their decisions.
3.2 Display Areas

Figure 1 displays the principal areas of the main application window. The text-oriented part on the left contains a sorted and filtered list of events together with their most important data fields, namely time and place, teachers and student groups involved, event status and some comments. The exact content depends on the selected mode – there are regular events, scheduling requests etc. It is possible to directly schedule or reschedule these events by dragging them to the graphical grid.

The graphical part on the right displays events in a grid-like way. All views share the same horizontal axis, but each has a different vertical axis, usually rooms, student groups, and teachers. Although these areas are used in all regular modes, their content is dependent on the horizontal axis bound to a particular mode (i.e. by the day of the week and teaching hours in the L or S mode).

These grid areas are complemented by textual detail area at the bottom. This area displays additional information on any object the user points to by the mouse. Even if the mouse points to overlapping or colliding events, the detail area displays the information for all of the events.

3.3 Axes

The areas are bound to vertical and horizontal axes; there are about 10 axes in MetroNG and their combinations make possible to display the data in the most useful way for particular tasks. Basically, the axes have tree or linear shape. From the structural point-of-view, the application modes are simply the aggregations of used axes combinations.

The tree axes hierarchically organize their items; users can expand or collapse desired branches. Especially the student axis is worth mentioning. For better comfort and lucidity, the application can be configured (by the user) to hide unnecessary or redundant branches of the attribute tree. For instance, the user can display the different years of study on the top layer. On the next layer, the hierarchy of study groups contains both the specialization of the study (e.g., IT, maths and physics) or language group (beginners, intermediate, advanced). But since the language groups are always scheduled within one specialization (IT - beginners, maths - beginners ...) the language branch is hidden and only the specialization branch is available. If the user picks a specialization, the language groups for that branch are then available at the third level of the tree.

The linear axes support no collapsing and hiding capabilities; instead nearly all of them stretch their size depending...
on the available space. The areas bound to such axes are stretched in the same way.

### 3.4 Collisions and Decorators

One of the most valuable features of MetroNG is the detection of collisions and prevention of collisions during the scheduling process. Existing collisions are displayed as hatched rectangles using contrasting colors, so that the user can immediately see a collision that is already present in the schedule. For an example, see the Figure 4. The figure also shows other decorators – graphical elements used to display important information about the events. For example, status flags that tell the user that an event is locked and cannot be modified.

Other important decorators are the color used as the background of the rectangle. Its meaning depends on the area where the rectangle is displayed. In the students area, the rectangles have the color same as the building they are scheduled to. In the room area, they have the same color (or multiple colored stripes) as the student groups that attend the event. Note that in the Figure 1 in the top (room) section of the view, the color of the background (color of the building) very often matches the color of the rectangle. This is an intentional side effect of this principle and the colors assigned to students and buildings. The displayed data are from a university with three sections and four buildings, where each section uses one (or two) buildings for most of their lectures. So the “green students” have most of their lectures in the “green building”. This whole setup makes it easy to spot anomalies, which are often significant in some way.

So far, we described graphical elements that help the user see the current state of the schedule. But there is another important group – features that allow the user to see possible effects of his or her actions (of changing the schedule). For example, see the Figure 5. When an event is selected (shown by a white background and thick black border), some parts of the view are covered with hatching of different colors and directions. It displays areas (in time and space) where there is a potential collision between the selected event and other events – that is, if the event was scheduled to that time and room, it would create a collision. Each type of collisions (see the Section 2.2) has a corresponding style of hatching.