universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„OCP: Visualization for Operative Curricula Planning at Universities"

verfasst von / submitted by

Raphael Sahann, BSc.

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Diplom-Ingenieur (Dipl.-Ing.)

Wien, 2017 / Vienna, 2017

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 066 926
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Wirtschaftsinformatik
Betreut von / Supervisor:	Univ.-Prof. Torsten Möller, PhD.

Erklärung zur Verfassung der Arbeit

Raphael Sahann, BSc.
Pergerstraße 9/1/2, 2500 Baden

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 1. Februar 2017

Raphael Sahann

Danksagung

Meine erste Danksagung gebührt Torsten Möller, der mich grandios durch die gesamte Arbeit begleitet hat und ohne welchen diese nicht in diesem Maße durchführbar gewesen wäre. Weiters möchte ich mich bei allen Experten für Zeit und Engagement bei den Interviews bedanken, besonders bei Martin Polaschek für seine intensive Teilnahme. Zuletzt möchte ich meiner Frau Sigrid danken, dass sie mir immer zur Seite gestanden hat, meiner Mutter — ohne sie wäre mein Studium nicht möglich gewesen — und allen Kollegen und Freunden, welche mir stets Rat und Antwort geben konnten.

Acknowledgements

My first acknowledgement goes to Torsten Möller, who superbly guided me throughout the whole thesis and without whom it would not have been possible to finish the thesis in the current extent. Further on I want to thank all experts for their time and dedication for the interviews, especially Martin Polaschek for his intense participation. Last but not least I want to thank my wife Sigrid, for remaining steadfastly at my side, my mother — without her studying would not have been possible for me — and all colleagues and friends for being on hand with help and advice whenever I needed it.

Kurzfassung

In dieser Arbeit kommt eine explorative Methode zum Einsatz, welche die Domain der universitären Planung im Bezug auf Studierendene und deren Quantifizierung untersucht. Verschiedene Entscheidungsträger die in den Planungsprozess involviert sind wurden identifiziert und interviewt. Die Resultate dieser Interviews werden als Übersicht aller Aufgaben und damit verbundener Probleme zusammengefasst um als Referenz für zukünftige Arbeiten in dieser Domain zu dienen. Die operative Semesterplanung wurde als Aufgabe für das Referenzprojekt OCP, ein Tool zur Informationsaufbereitung während der Planung, gewählt. Das Tool wurde iterativ, in einer stätigen Feedback-Schleife mit den representativen Experten, erstellt. Design-Entscheidungen während den unterschiedlichen Phasen, als auch die Entscheidungen und Details der finalen Implementation werden präsentiert und erklärt. Das Tool wurde von den Experten evaluiert und zeigt nicht nur große Effizienz in der Planung, sondern eröffnet auch neue Wege für die Exploration der planungsrelevanten Daten. Zuletzt werden die Erkenntnisse aus dem Design-Prozess präsentiert und weitere Teilbereiche mit Relevanz für die operative Planung als zusätzliche Startpunkte für zukünftige Arbeiten aufgezeigt.

Abstract

This thesis takes an exploratory approach to investigating the domain of operative student-related planning at universities. Several stakeholders of the planning process have been interviewed to identify the tasks they face in their work and the problems they encounter. A summary of all tasks and problems is given to establish a point of reference for future works in that domain. The task of operational semester planning was chosen to be supported by the OCP tool. The tool was created iteratively in a constant feedback-loop with the experts involved in operative planning. Design choices at the different stages are shown and the decisions and details regarding the design as well as the implementation of the tool are stated. OCP has been evaluated by planning experts of different fields of the university and could not only ease the process of planning, but also allowed experts to use it as a novel way of exploring and interpreting different aspects of the planning dataset. Finally the insights of the iterative design process are presented and more extended features of operative planning are stated as potential starting points for more future work in that field.

Contents

List of Figures

CHAPTER 1

Motivation

Student numbers have been increasingly growing over the past few years and trends show that there is no end of growth in sight. The rapid increase in students proves to be a difficult task for planners who are concerned with the amount of students that attend any given course at the university as well as the exploding costs of increasing numbers of students as more and more lecturers are needed. Finally some old curricula could not keep up with the large amount of students because of cost issues of hiring new lecturers, or simply room availability issues if special laboratories are needed for a course, in which space is a limited resource.

From the point of view of a data scientist the increase in student numbers is a welcomed sight, because it also yields an increase in available data to analyze and work with. To this day the amounts of data gathered by universities are not used to its potential, sometimes data is only gathered to be stored and may only show up in an occasional report. But the gathered data can provide so much insight into different aspects of the university which can also help tackling the problem of planning.

The rapid growth in student numbers was increasingly impacting the Faculty of Computer Science at the University of Vienna during the past few years, which was one of the factors, alongside the fact that the domain of Computer Science is rapidly evolving, for the decision to create new curricula for the faculty.

1.1 Personal Involvement

The Faculty of Computer Science of the University of Vienna released new versions of all of their curricula in the summer of 2016, which replaced the previous curricula in October of 2016. Two new specializations topics were added to the Computer Science Bachelor's curriculum, two previous Master's curricula were included as focus topics alongside two new focus topics into a new Master's curriculum, one Master's curriculum was discontinued and all other Bachelor's and Master's curricula were changed drastically. The changes went so far that some courses can either be chosen during the Bachelor's or the Master's study, which was previously impossible.

These changes in the curricula were implemented in a process which lasted more than two years and all professors of the faculty were involved in it. To coordinate and manage the process a joint working group was created, which was called *Curriculare Arbeitsgruppe*[1] (short *CurrAG*). In addition to the faculty's professors I was asked to join this working group, because at the time I was working as an Assistant for the Head of Students Programme, who was heavily involved in creating the new curricula.

At the beginning smaller focus teams were established for different "categories" of courses — this included e.g. Business Informatics, Mathematics, and other similar groupings. The respective courses were discussed internally within these small groups and once in a while the whole CurrAG would meet and the results from the focus groups were presented. After a couple of iterations the final courses started to manifest and it was time to categorize them and put them into place in a semester in the curriculum based on their complexity, prerequisites and possible lecturers. But these were not the only factors to comply — there are also budgeting calculations, legal issues and the problem of fitting into the modeling capabilities of the registration system of the University of Vienna. All these factors needed to be accounted for by few people planning the final curricula — including the vice dean of teaching, the head of studies programme and myself — without the aid of a tool, because no tool existed at the time.

1.2 Glossary

The following abbreviations will be used over the term of the paper:

SWS - measure of cost
SWS is a German acronym for *Semesterwochenstunden* which translates to weekly teaching hours per semester. It is an abstract measurement of cost of lectures, because more weekly hours per lecture directly relates to the increase of the total cost of that lecture.

ECTS - measure of required effort
ECTS is the acronym for *European Credit Transfer System.* It is a measurement of student effort, used to compare the scope of lectures. One ECTS stands for 25 hours of work for a student to complete a given course. Based on this universities all over Europe can judge whether the course a student took at another university was only an overview of a certain topic, or did deal with it in detail.

SPL - position in the faculty
SPL is short for *Studienprogrammleiter*, which literally translates to Director of

[1] translates to *curricular working group*

Studies Programme. The holder of this position is responsible for managing and controlling all courses a faculty has to offer — thus allowing all students to take the courses they need to complete their studies.

CurrAG - curricular working group
A working group formed to tackle the creation of new Bachelor's and Master's curricula for the Faculty of Computer Science. All professors of the faculty and a few other participants were part of this group, which lasted over two years until all curricula were finalized.

STEOP - part of the first semester of each Bachelor's programme
The STEOP — a German acronym for *Studieneingangs- und Orientierungsphase* — is a block of lectures in the first semester of a Bachelor's curriculum, which is mandatory for each student in order to advance to higher semesters.

CHAPTER 2

Related Work

There are currently no known computer science publications in the particular domain of operative planning for universities, but plenty of work tackling similar situations for different domains.

2.1 Scheduling and Timetabling

A whole research field is dedicated to the problems of scheduling and timetabling. A common problem among many publications in the field is regarding the timetabling of courses or examinations. Havås et al. [6] tackle the problem of course scheduling by modeling it into an integer programming problem and visualizing all outcomes to easily identify the preferred one. Komijan and Koupaei [9] create a binary mathematical model for the problem of examination scheduling as well as course scheduling, allowing multiple exams to be in the same room and multiple instances of the same course to happen in the same semester. A major restricting factor in these problems is always the availability of rooms, which is a problem this thesis chose to completely abstract from.

Boronico [3] creates a mathematical model to simulate student enrollment for courses and suggest a faculty schedule that minimize expected conflicting courses as well as integrating several other parameters. Finally Petrovic and Burke [11] present an overview over all commonly used techniques of dealing with the examination and course scheduling problem in universities, categorizing them into meta-heuristic, multi-criteria, case-based and self adaptive approaches.

All these scheduling and timetabling approaches, while somewhat related to the problem of semester planning do not tackle any of the specific problems faced in the domain of operative planning. They can rather be seen as complementary approaches to aid with creating a timetable based on the plans for the current semester, because scheduling relates to the distribution of the courses within a week and also the distribution of courses within a given set of rooms, whereas operative planning deals with the amount of expected students within a course in terms of planning the number of groups of an individual course to provide enough groups for all students.

Scheduling and timetabling can therefore be automated and the main focus in the scheduling community is finding efficient ways of calculating all possible scenarios based on the given restrictions (overlap, availability of rooms, available time slots, etc.) and visualizing the feasible solutions to make the decision process easy. Operative semester planning on the other hand does not need to take rooms and time slots into account — in fact this thesis purposely excludes the task of room planning — and rather focuses on the behavior of students and the prediction of demand for each individual course.

2.2 Visual Data Analysis

The domain of this thesis involves large quantities of data, but is also dependent on expert knowledge, because the prediction of human factors is not quantifiable as an input to a fully automated system. The field of Visual Data Analysis tries to make large datasets accessible to human users and help them gain insight into the data.

This thesis uses different visualization and filtering techniques shown by Keim [7] to make the data accessible for the user in an efficient manner, therefore respecting the visual analytics mantra by Keim et al. [8]:

Analyse First -

Show the Important -

Zoom, Filter and Analyse Further -

Details on Demand

The mantra was followed throughout this thesis by categorizing the different visualization tasks according to Brehmer and Munzner [4] in meta-tasks and evaluating each task on its relevance in the mantra.

Tory and Möller [16] show that human factors have a significant impact on the quality of the solutions in visualization, therefore the whole process of this thesis is based on a human-centered design process, including experts in multiple iterations during the whole process to create a meaningful and well-perceived solution. The design study conducted by Lloyd and Dykes [10] concludes that an early and frequent involvement of experts as well as early prototypes with real data increases the amount of time spent building a system, but increases the quality of the result by a noticeable margin, thus justifying the additional work.

Design studies are a crucial part of the visualization contributions today. A design study is used to identify a specific problem and create a solution based on the needs of experts, usually in a confined domain — Booshehrian et al. [1] created a tool to aide decision making in the domain of fisheries management in Alaska; Borkin et al. [2] improved an established workflow in medical diagnostics of heart disease by introducing a simpler representation of the available data. Similar to these tools this thesis uses the design study methodology as presented by Sedlmair, Meyer, and Munzner [14] and the guidelines for design study contributions [13] to create a tool to help decision making in the domain of operative semester planning in a university context.

CHAPTER 3

Domain Details

3.1 Available Data

Lots of different kinds of data are currently collected by the University of Vienna, but most of the data is not used, or only condensed into static reports, which consist of large tables full of values, thus not really accessible and therefore are only used for very few tasks. Data is stored in various different locations across the infrastructure of the university and depending on the confidentiality of the information contained within the data, different datasets are only accessible for specific positions which need access to that data. This made the collection of all relevant data for the thesis a more complex task than initially anticipated.

The following sections give a short overview of the data which is stored by the University of Vienna at the moment.

3.1.1 Event-based Data

The data gathered during the operation of the University of Vienna has an event-based structure. Each data-point is captured at a concrete event and has a concrete duration. A group of a course is a discrete event which contains a unique identifier, a timestamp and a duration. It is related to the course, the lecturer holding the group and the room where it takes place. Exams are events with a shorter duration, but a similar structure.

Students interacting with events result in other events like the registration for a group of a course, being accepted in that group, dropping out of a course, writing an exam and getting the results of an exam. These events also contain the unique identifier of the student, a time point and event specific data such as the relation to the event and also the student.

3.1.2 Static Data

Static data does not have a time-dependent component. Personal information of students and lecturers is part of the static data, even though the inscription of a student at the university may be a time-dependent event (and is also stored like that), the date of birth, gender or other personal details are constant.

Another static dataset is the list of buildings, rooms, lecture halls and other physical structures.

3.1.3 Financial Data

The last dataset regards the finances of the university. This data contains salaries, rents, financial planning data and many other expenditures. The only relevant part for the domain of planning is the financial planning data, which provides a financial threshold for the planner. Financial data is classified information, not available to the author of this thesis and because of the lack of relevance for the domain will not be explained in more detail. It is just listed for completion purposes.

3.2 Operative planning

A crucial part in this domain is the planning of the following semester, which happens twice each year, shortly before the new semester is starting. During this phase the SPL has to make sure, that each course, which is stated in the curricula to be present during the semester, has to be organized to take place. It is not possible for a course stated in the curriculum to not happen within the given semester.

Miscalculations during planning can have two different negative outcomes:

- **Class size too small** — not all students that are eligible to attend the course can do so, thus effectively prolonging their studies by making them wait at least another semester before they can attend the course.
- **Class size too large** — groups of a course are not filled, existing groups need to be closed in order to be cost-efficient. Lecturers lose hours, thus their payment for that group and students already signed up for the group which is closed may have planned their other courses according to the time-slot of the now closed group and cannot attend other groups because of timing collisions or at least need to invest considerable effort for changing their plans.

Calculating predicts of the number of students in the following semester is not a trivial task, because the number is dependent on a large amount of unpredictable external factors as well as many internal factors, which makes predicting future student numbers a task which can only be performed by experts, who got extensive insight in the behavior of students as well as the structure of the courses. This insight helps them to estimate the internal factors within the faculty, whereas external factors can only be predicted through the experts personal experience.

CHAPTER 4

Requirement Analysis

The first requirements stemmed directly from the needs that emerged in the CurrAG. The goal seemed simple at first, but turned out to be dependent on too many factors to manage them in an easy and manual way.

For a Bachelor's curriculum a plan consisting of six semesters needed to be designed. The following list shows all factors which needed to be respected in order to create such a plan.

Direct Prerequisites Some courses are directly dependent on other courses and cannot be taken without having completed all prerequisites (e.g. to attend Software Engineering 2, Software Engineering 1 needs to be completed).

Indirect Prerequisites There are courses which teach "basic" knowledge, and such which give in-depth knowledge of a topic. Students can profit from knowledge of certain basics in order to complete a course, so it should be possible to attend the "basic" courses before continuing with "advanced" courses.

Lecturer availability Each lecturer is only capable of holding a certain amount of lectures during each semester and preferably half of this lecturers courses should take place in the summer semester, and the other half in the winter semester.

Student workload A Bachelor's curriculum consists of 180 ECTS, which should be distributed equally over all six semesters (=30 ECTS per semester). While small discrepancies are possible, no single semester should contain considerably more ECTS than another semester.

Cost Typically there are two types of courses — courses with no continuous assessments like lectures take place once a week and if students want to attend it they can visit the lecture, but they can also choose not to. These courses are graded by completing a single exam at the end of the semester. The other type of course is the continuously assessed course — students are required to attend this course each week, participate in it and are graded based on this participation (e.g. assignments, presentations). These courses are limited to a certain number of students per group, which therefore requires more groups if more students attend such a course. Each group also needs a lecturer, which in the end translates to additional costs for the faculty, the more students attend a continuously assessed course.

Its a known fact that a higher semester is directly correlated to a lower number of students, because throughout the course of study students drop out. Thus a continuously assessed course in a later semester is less expensive than the same course in an early semester.

While some of the requirements are easily met, others are not as obvious and need to be recalculated with every change made to the curriculum − moving a continuously assessed course from the fourth to the second semester might double the number of groups needed, effectively doubling the cost of that course.

The capability of calculating these factors, showing the differences in costs and also the related workload of the involved lecturer were the basis on which the very first prototype (described in section 5.1) was created.

After the finalization of the curricula by the CurrAG, in which the prototype was generally well received, another effort was made to add features and create further designs. These designs were then analyzed by different experts of the domain.

4.1 Expert Interviews

4.1.1 Methodology

The following interviews were part of the conducted design study within this thesis, which followed the general rules of the design study methodology as proposed by Sedlmair, Meyer, and Munzner [14]. A design study is a human-centered design process, which seeks to interact with domain experts on a regular basis to let the needs and the knowledge of these domain experts influence the whole design process. The basic process revolves around creating a prototype, showing the prototype to the domain experts, gathering feedback from the domain experts and incorporating this feedback in the creation of the successive prototype. Each iteration yields another prototype, which continuously converges towards a final tool based on the needs of the domain experts. Several iterations took place to finalize the tool presented in section 5.3.

The first iteration of expert interviews did not only look for feedback to the design-prototypes, but aimed to understand the different views of the stakeholders of the domain, get insight in their work and the problems they are facing, thus finding possibilities of generally aiding them.

4.1.2 Vice Dean of Teaching

The first expert is the Vice Dean of Teaching of the Faculty of Computer Science, head of the research group Visualization and Data Analysis and also the supervisor of this thesis. He was also head of the CurrAG and therefore involved in planning the path through the semesters of the new curricula, using the first prototype mentioned in the section before, as well as in section 5.1.

His viewpoint on this topic is a general one, which does not focus on a specific detail, but is looking to get an overview of what happens within the faculty, regarding students as well as lecturers. The main goal is through the means of comparison and analyzing to be able to comprehend how and why certain things happen — in order to plan ahead and manipulate influential details to make the whole faculty run smoothly.

One of his visions involves analyzing typical paths that students actually take to complete their studies and clustering the paths based on different background information of the students (e.g. schools visited prior to studying) or grouping by the similarity of the paths to find cohorts of "similar" students, evaluate the success of these cohorts and possibly make changes to the current curriculum in order to prevent failure of a specific group.

He is also interested in detecting overall trends within the data, creating scatterplots and other visualizations to compare different factors and just gain insight by visually exploring the data in general. A tool for exploring the available data should be separated in different views based on the task currently focused on — e.g. overall curricula statistics, operative student data, or financial calculations.

4.1.3 SPLs and other planning-related positions

SPLs are renowned for their in-depth knowledge of all courses within their respective faculties, because they are in charge of the operative planning of each semester and therefore need good understanding of the dynamics between courses, lecturers and students.

All experts interviewed in this section showed great enthusiasm for the aims of this thesis and immediately agreed in participating in further interviews, which was finally the reason the field of operative planning was chosen to be the main focus of this thesis.

Computer Science

The SPL of the Faculty of Computer Science is responsible for all operative planning within the faculty. He was also part of the CurrAG and involved in planning the specifics of each semester of the new curricula, therefore also using the first prototype.

His view is very student centric — his main focus is providing enough space in each course to guarantee all students with a major in Computer Science, that are eligible of attending (fulfilling all prerequisites) to be able to do so and not being forced to skip a semester because a course was full.

His typical workflow of planning the following semester consists of the following steps:

1. He determines all courses which will take place in the following semester — based on whether a course is held in both semesters, only during the summer or winter semester, only bi-annual, or even just on demand.

2. After figuring out all courses that will take place in the following semester he compares the registration counts and positive exam results of the previous instance of the course to the number of students in the current semester to estimate the amount of eligible students in the following semester.

3. By calculating the ratio of number of students in the last instance of the course and the number of students currently attending courses he can make a prediction on how many students are going to register for the course in the following semester. The number of exams taken in the course in relation to the number of exams positive can give him extra insight in edge cases in which the prediction is not completely clear, because based on the number of students that failed the course, the number of students who will possibly re-attend can also be estimated.

4. He finally sets the number of groups needed per course according to his estimations and tasks the course coordinators to provide this amount of groups for their course.

To aid his planning process, the visualization which splits the courses according to the semester plan into six different semesters (for a Bachelor's curriculum) is not particularly helpful, because he thinks about courses in terms of winter semester courses and summer semester courses and therefore needs a possibility to see an aggregation depending on the respective semester. The first prototype introduced the limitation of only viewing a single curriculum at once, which was a design choice at that time to clearly differentiate between different curricula. The therefore introduced additional task of switching between different curricula does not contribute to the planning process, because he needs to plan all courses of the following semesters, regardless of the curriculum they are held in.

Combining courses into their respective modules, as stated in the curricula, is also seen as an unnecessary complication, because courses need to be planned with regards to their groups, not their modules. Modules can include continuously assessed and not continuously assessed courses which is not useful for planning, as the latter do not need a size estimation. It is also possible for two courses to take place in different semesters even though they are parts of the same module.

A useful view for planning must not only show the current and upcoming semester, but should also include the two semesters prior to the current one, in order to make predictions based on the previous year. Adding additional future semesters is not needed, because usually only one semester is planned at a time.

Curricula Planning At the time of this interview the final domain of the thesis was not chosen, therefore the task of planning a new curriculum, as happened within the CurrAG was also discussed. Regarding this task a few issues in the field of faculty interdependent courses were identified. The first potential flaw of such a system could potentially be the existence of mutually used courses between faculties, because different restricting parameters from either faculty might occur, which would not be noticeable by the planner, who is usually only a member of one single faculty. If the course is held in an equal share from both faculties the distribution of rights between the two planners from each faculty could become a problem — e.g. what metric should be used to determine which planner can modify certain parameters.

A more general problem in planning a new curriculum is the distribution of courses over summer and winter semester for individual lecturers, because any lecturer could hold lectures at other faculties as well, which are not calculated into the total workload of a lecturer, when only viewing faculty internal courses.

Other general remarks

- The concept of a module coordinator, or also course coordinator, dictates that the coordinator – usually a full professor – is responsible in finding qualified instructors for the needed number of groups. If additional lecturers are needed the coordinator can hire external lecturers to help out with teaching. Even though the concept is generally known and helpful for the planner, it is not formally defined by the University of Vienna and therefore not used throughout all faculties; the general way of thinking shifts towards this concept though.
- A person-centered view for grasping the whole extent of the activities of lecturers in the own faculty and other faculties could be useful to determine availability or general workload of a lecturer.
- If it is not possible to access personal data from other faculties to determine the external workload of lecturers it could also be possible to simply import the courses held by internal lecturers on other faculties as constraints for planning, thus incorporating each lecturers actual workload.
- Planning a course which has groups in both semesters versus a course which is held in only one semester is crucial knowledge in a planning situation, but irrelevant from a financial point of view, because the amount of students attending a course and thus the amount of teaching hours needed to the cover it stays constant.
- A more strict chain of dependencies could lower the general dropout rate of courses, which would make it possible to get more exact calculations on the estimated number of students in a particular course, because only students who fulfill the whole chain could register for the course, and each of those students who can register is more likely to also do so.

Economics

The Head of the StudiesServiceCenter Business, Economics and Statistics is responsible for the operative planning of the Faculty of Business, Economics and Statistics.

His way of planning a future semester does not differ much from the way described by the SPL of the Faculty of Computer Science, so I will the differences in approach and other differences to the way previously mentioned, rather than describing the whole process once again.

He uses the Kognos tool provided by the University of Vienna to get an overview of the work-hours for each lecturer, but critiques that the Kognos tool only provides reports, which makes it solely a reactive tool to analyze data in the past.

The way of providing data in the form of individual courses for the past three and the upcoming semester in a combined view which shows all courses of the semester seems the most appropriate for him, because it represents his way of thinking about courses.

He also stated that it would make sense for him to compare the planned semester to the real semester once it is over, to evaluate the accuracy of the prediction and allow fine-tuning of certain courses which may have been off with their prediction to achieve a higher overall prediction quality.

The tabular prototype is seen as a tool for analysis of current and historical data – e.g. comparing student performance in different courses and using this information to tweak individual courses, but this is not always a factor that can be determined by just looking at the numbers and the lecturer of the course usually can assess the status better through his experience. Therefore his main focus is also the operative planning process and not the tweaking of courses.

After his planning process is finished he needs to report his plannings to the dean of the faculty, who has to approve it before it will get into effect. The tool therefore should allow a tabular export of the planned data, which can be presented to the dean, because the dean does mostly care about the overall key performance indices.

Another aspect mentioned which always poses a problem in his daily work is the urgent replacement of a lecturer for a single lecture, e.g. in the case a lecturer gets ill and he needs to find somebody to hold the course on the same day.

Chemistry

The head of the Department of Organic Chemistry, the Vice Dean of Teaching of the Faculty of Chemistry as well as the SPL of Chemistry are all combined in a single person. This does not only make him a very busy man, but also responsible for the controlling as well as the operative planning of the Faculty of Chemistry.

His way of operative planning does also coincide with the previously described way. Because the Bachelor's and Master's curricula in chemistry are almost completely separate, it could make sense to be also able to view them independently during planning.

The Bachelor's curricula in chemistry are separated into four different blocks, and each part is dependent on the previous block.

1. First semester: STEOP
2. Second semester: general practical courses
3. Third to sixth semester: in-depth practical courses and lectures based on the field of the practical course
4. The Bachelor's thesis

All practical courses are held in both the summer and the winter semester, whereas the lectures only happen once each year. This is dictated by the need for space in the laboratories to do applied chemistry during practical courses and the limited number of laboratories available. His operative planning is therefore also centered around the availability of rooms for particular courses.

He identified two different problems which can lead to bottlenecks during the semester. Both deal with the limited amount of available space in relation to the amount of students which need to attend a critical course. The first bottleneck appears because not all students who need to attend a space-restricted course can do so, which forces them to wait until the next semester to reattempt attending this course. New students trying to grab a place in that course and the ones waiting from the prior semester add up, thus creating an even greater demand than before. The second bottleneck is similar, but even harder to detect, because it features students who fail the course and need to retake the same course in the following semester — the result is a course which will overflow in the following semester, even though it did not in the prior one.

In this regard he would appreciate a tool which makes it possible to detect bottlenecks before the appear to try and take preventive measures or even adapt the curriculum slightly to cope with the problem. It would also help to set a fixed maximum number of groups for certain courses to make it possible to highlight courses that are likely to exceed the maximum capacity.

Prospective students and drop-out rate Another problem while planning early semesters involves the inability to predict the number of new students signing up for a Bachelor's programme in chemistry. This number is highly fluctuating each year, because of different surrounding factors — not succeeding in the acceptance test in medicine or pharmacy seems to be a common reason to start a Bachelor's degree in chemistry, because different aspects of the chemistry curriculum also count towards the initially attempted studies. Another implication of this is the fact that most of those students will retry the acceptance test in the following year and those who succeed will discontinue their studies in chemistry in favor of the other field of study. Currently efforts are being made to reduce the drop-out rate.

A different anomaly happened during a particular year, where the number of registrations for a laboratory-dependent course suddenly tripled in comparison to the previous semesters. It turned out that a course with a similar content, which is mandatory in the Bachelor's programme of Pharmacy, had its time-slot changed to Friday at 5:00 p.m., so students chose to attend the course at the Faculty of Chemistry, because it had a more convenient time-slot.

From the view of a vice-dean he stated that the tabular prototype would be a good tool to evaluate current operations by the means of comparing lecturers based on the percentage of positive exams in their courses, or amount of hours covered, finding groups which are not completely filled and monitoring overall student success.

History

An Assistant at the StudiesServiceCenter Historical and Cultural Studies is responsible for guiding a combination of several SPLs in that field. He is responsible for operative planning, coordination and administration of courses as well as the sign-up system in the Historical and Cultural Studies.

The process of planning a future semester in the Historical and Cultural Studies differs substantially from the previous workflows. Planning starts with a process where lecturers propose topics they want to teach in accordance to the field they are working in. These proposals then need to be categorized to fit the topics in the curricula and grouped together to form modules, which are then added to the list of courses for the next semester. If two similar topics would not be contained within the same module a student could take both courses, learn the topic once and get credit for both courses.

The proposals usually do not cover the whole range of topics set in the curricula, so after the proposed courses are arranged, the missing topics from the curricula are added to the list of needed courses. Based on this list the required number of groups to cover the demand for each course is estimated by calculating student numbers from similar topics in previous semesters (similar how other SPLs estimate future demand).

The estimated demand for each course is then published on a website, where every lecturer can select which groups to cover based on personal knowledge and capacity. After a set period of time the selection closes and a meeting is held with all lecturers. During the meeting lecturers can swap courses and empty courses are assigned to lecturers until each lecturers contingent is fulfilled. Remaining groups are then advertised to the public and external lecturers are hired to completely cover all courses.

Structures within the faculty still follow a historic hierarchy due to the age of the faculty — more than 150 years. These hierarchies inherently bring a different range of problems, with which the planner needs to work.

- The higher a professor is in the hierarchy, the fewer courses he wants to hold. Even though there is a requirement to teach, there is no superior position that would force the professor to fulfill the requirement.

- The courses offered from the professors sometimes do not match any of the modules of the curricula, so they sometimes need to be grouped within a module, which hosts a wider range of topics — the problem with that is the fact that two students who both finished such a module could have a completely different knowledge due to the highly different courses within the module.
- Professors can "bring" external lecturers to their courses, which then need to be employed, even though the course does not require a second lecturer.
- Some topics are avoided by internal lecturers, because they are very demanding to teach or need a lot of time, which leaves those topics to external lecturers. External lecturers fluctuate each semester, thus making the quality of these topics vary greatly, because the realization is highly dependent on the current lecturer.

The main limiting factor in planning is the assignment of courses to lecturers and persuading lecturers to run their assigned courses. An useful way of persuading lecturers would be to show them the current status of the planning situation and also letting them compare themselves with other lecturers to make them realize that the amount of teaching is actually spread constantly throughout all lecturers. A visual representation which makes it easy to understand the demand in the future semester could persuade lecturers more easily than just telling them that they are needed.

4.1.4 Accounting and Finance

The group of Accounting and Finance is responsible for the cost calculation of curricula and the controlling of held courses. They try to calculate the worst-case scenario from a financial perspective for each faculty and help the vice-rector of teaching and students estimate the needed budget for each faculty.

The group is not directly involved in neither curriculum planning, nor in operative semester planning, but is indirectly restricting both processes by financial planning and providing a financial ceiling to the planners to work with, because money is a limited resource and needs to be spent according to actual needs, instead of being spent on miscalculations. Therefore the group spends a lot of time calculating planned curricula and evaluating the spent money on whether it was used purposefully, or could be used in a more effective manner and presents suggestions accordingly.

Calculating curricula Financial cost planning of a new curriculum is calculated based on a worst case scenario, which only takes into account strict dependency chains, ignoring everything else. Even if a course is planned to happen during e.g. the fifth semester, but it would be possible for a student to already enroll in this course in the second semester, the course is calculated as if all students in the second semester would actually do so.

Therefore it would be useful to depict dependency chains in the tool and have a maximize-algorithm to calculate the highest possible cost. A simulation of all possible scenarios and a plot of the resulting distribution could be useful as well. Such simulations should also be easily comparable to actual values.

A visual arrangement of courses based on their dependency chains in a graph-based view could help understanding dependency chains, which are currently only defined in textual form in the curricula. The current way of financial planning is limited by the capabilities of Microsoft Excel, which can neither depict dependencies, nor simulate different scenarios efficiently.

The two factors which play a role in financial planning is the estimated number of students in a course and the type of the course. Based on the type of a course a different way of planning is required, because some courses need special labs, computers, staff, or other specific requirements. The number of students is directly related to the cost of a course, because the more students attend a course, the higher amount of SWS is needed.

Controlling The second big task of the Accounting and Finance group regarding the operative semester planning is controlling. Controlling monitors the results of the operative semester planning, looks for bottlenecks and helps the vice-rector to see whether increasing allowed semester hours — thus increasing the budget — is justified or could be avoided by planning differently. It also examines the workload of lecturers, current load compared to the planned amount, the general efficiency (e.g. two groups of a course, which are both only half-filled could be combined to a single group which is completely filled) and historic data of a lecturer to get insight on the ways lecturers are managing their courses. The insight on who already held a course or just this persons general involvement in a subject could be helpful for controlling as well as operative planning.

Controlling also looks into the number of students for each course to distribute rooms and plan exams and warn lecturers if their courses could cause bottlenecks in future semesters, due to overflow of students or too little room to accommodate all students. To counteract bottlenecks and make the flow of a curriculum more evenly distributed the Accounting and Finance group is monitoring students and analyzing their progress and success. A current study is trying to find out whether the success in the acceptance test is linked to the success in the overall subject. Another current project from the EU is also targeting the success of students by nudging — luring students to take exams in the right order instead of postponing particular courses or exams. The university of Nottingham, Edinburgh and TU Graz are but a few involved in the project and the University of Vienna is currently running a similar process in the field of economy.

The sequence of passed courses for each student and the time-frame this happens in is a very interesting field, but highly dependent on the specific topic, because a great number of local factors play an important role on the success of a student.

It was finally stated that the Accounting and Finance group is currently in dire need of all its resources and therefore is not able to participate in further interviews.

4.1.5 Teaching Affairs and Student Services

The group of Teaching Affairs and Student Services is responsible for modeling courses whenever a curriculum changes and adding them to a global dependency database in order to make them available for services of the university like the student enrollment service, which has to make sure a student fulfills all dependencies of a course before registering in it.

A typical workflow in this group is extracting courses and modules from curricula — everything is stated in a descriptive, human-readable textual form — and entering them into the database, including all topics, dependencies and other defining factors.

The process of creating a new curriculum consists of four steps:

1. Planning a new curriculum done by the faculty which is in charge of that curriculum.
2. Creating a text-document describing the new curriculum, all modules contained within, dependencies, other structures as well as descriptions of all courses and the legal frame in which the curriculum operates.
3. Acceptance and publication of the new curriculum by the senate of the University of Vienna. A curriculum does not become effective before its official announcement by the senate.
4. Entering the details of the curriculum into the database of the university.

The Student Services group is closely working together with the senate, advising the senate during the acceptance process of a new curriculum by checking whether the drafts of courses in the new curricula are possible to be represented by the means of their dependencies. After the senate accepts the curriculum it tasks the Student Service group with reproducing all contents of the new curriculum by entering them into the database. The input data for this step is the curriculum in the form in which it is published — a human-readable, legal text-document.

They estimate that about fifty percent of his group's work is related to reading text documents, gathering information and restructuring the data to fit into the schema of the database.

A tool which allows faculties to already create curricula in a digital manner and then outputs a fully written curricula text could make both the task of creating the curriculum and transferring it to the database less tedious. The tool could also be used to visualize dependencies and even predetermine whether a concept is possible to be represented in the database or not. It could also be connected to a server, where questions of the faculty whilst planning new curricula would be posted along with real examples and his group could directly answer based on those examples.

Another task is the switch from an old module to a newer version. This involves the knowledge how many students are currently taking courses in that particular module, estimation how many students are going to take the module in future semesters and whether a change in the module involves a change in dependency chains. If a dependency is also changing with the module a temporary solution has to be in place until all students who fulfilled the old dependencies have also finished the old version of the module, or maybe courses of the newer module can be attended if students only fulfill the old dependencies. A solution according to each specific situation has to be found and its consequences need to be estimated and included into the operative planning — e.g. if a course is available for another two years before it is fully replaced by the new version, and the new version has different dependencies it has to be taken into account that possibly all students, who finished the previous dependencies, but not that module will try to attend it at the last instance, which may result in a spike in attendance for the last iteration.

They also clarified a structural convention which states that modules, which can consist of one or more courses, can only have dependencies to other modules, whereas courses within a module can be dependent on other courses within the same module. It is not possible for a module to be dependent on a course, or for a course to be dependent on a different course from another module. This is restricted by the representation of the two entities in the database.

Special thanks A special expression of thanks goes to Stephan Prechtl, who – due to his position — was able to provide all operative data needed to create meaningful prototypes for this thesis and explained the structure of the data and the different kinds of data tables available which was crucial information for the design of the prototypes.

4.2 Problem Analysis and Resulting Tasks

The following subsections describe the different categories of problems that were identified by analyzing the input from the experts and define tasks based on the problems.

4.2.1 Curricula planning

The problem of curricula planning involves the following entities: modules, courses and lecturers.

It is trying to find an optimal solution for the new curriculum, which takes all external and internal dependencies into account. Automation of this problem is not possible, because the limiting factors are often not comparable and could also form circular dependencies, which can only be solved through human estimation of the gravity of not fulfilling a particular dependency. Also human and political factors have to be included, which can not be easily modeled to be calculated by a computer.

Derived tasks

- Create new curricula optimally adjusted on external and internal factors

- Change existing curricula
- Visualize the gravity of changes by comparing to the former curriculum
- Give a comprehensible overview of a curriculum

4.2.2 Operative semester planning

The problem of operative semester planning involves the following entities: courses and groups.

Planning future semesters is dependent on historical data of the courses as well as the knowledge of the planner, who should be able to predict changes in student numbers, demand and staff. It is crucial that all students who are eligible to attend a course also get the possibility to do so, because otherwise bottlenecks could appear, or students could discontinue their studies because of lost time.

Derived tasks

- Predict the number of groups needed for each future course to include all eligible students
- Visualize details about different courses or even groups (e.g. failed students)
- Calculate the number of ineligible students registering for a course
- Schedule courses to prevent overlap with other courses according to the courses a predicted student might take
- Allocate rooms to accommodate all courses

4.2.3 Data exploration and evaluation

The problem of data exploration and evaluation involves the following entities: students and lecturers.

Data exploration and evaluation is a capability most wanted by vice deans, who need to gain insight into the processes in their faculties and are evaluating individual lecturers to confront them with facts about their work in personal meetings, as well as individual students or groups of students to monitor different success rates within the faculty.

A reliable overview about things happening in the faculty can also help in situations where spontaneous events disrupt the normal business — e.g. finding urgent replacements for ill lecturers.

Derived tasks

- Explore all available data
- Evaluate individual lecturers
- Monitor overall student success
- Keep a general overview of all processes happening at the faculty
- Determine trends within the faculty

4.2.4 Financial planning

The problem of curricula planning involves the following entities: courses and financing.

By calculating worst case scenarios for each new curriculum an estimation of the costs inherent to it can be made. This calculation is then used as a guideline on whether a curriculum is deemed too expensive and needs to be changed, or stays within the budget and can therefore be approved. This calculation of costs is not generic, as it needs to adapt to each individual option present in a given curriculum and can therefore be very complicated to predict in certain cases.

Derived tasks

- Calculate costs of curricula before they are realized
- Give recommendations on how to adapt a curriculum to make it more cost efficient
- Visualize outliers (e.g. very expensive courses, etc.) to give insight on the factors that make them become outliers
- Aid budget planning by presenting comprehensible calculation results (management-friendly views)

4.2.5 Controlling

The problem of controlling involves the following entities: courses and lecturers.

Controlling deals with the evaluation of courses that took place during the previous semester. It compares the actual key performance indicators to the planned values for a specific course, thus evaluating its efficiency. If a course is deemed inefficient the vice dean of the responsible faculty is informed and needs to remodel the course. This in-depth evaluation of all courses also shows possible future bottlenecks before they become an issue.

The second entity which is evaluated are lecturers, who are also scored based on performance indicators and reported to the heads of the responsible faculty if the indicators are significantly below average.

Derived tasks

- Analyze the cost effectiveness of current courses
- Evaluate lecturers and their methods of operation
- Predict bottlenecks or possible inefficiencies
- Cluster courses based on various factors to gain insight and easily filter certain courses (e.g. percentage filled)
- Show historical trends for courses, lecturers, curricula or even faculties

4.2.6 Dependency modeling

The problem of dependency modeling involves the following entities: modules and courses.

Dependency modeling is another limiting factor of creating a new curriculum, because an inconsistency emerging in the curriculum makes it impossible to be represented in the dependency database, thus not making it available for any of crucial systems — e.g. the platform where students can enroll in courses. If a curriculum is consistent regarding all dependencies it is transferred into the database and therefore made available to all services of the university. The Teaching Affairs and Student Services group is currently manually entering dependencies into a database which are only stated in a textual representation within the curricula.

Derived tasks

- Derive an automated way of dealing with dependencies within curricula
- Create the dependency representation of a curriculum in the database of the university
- Give recommendations on how to adapt a curriculum to make it possible to represent in the dependency schema
- Help operative planners understand conceptual discrepancies

4.3 Main Focus

The first prototype of this thesis was focused on the tasks related to curricula planning, but after finishing the requirement analysis it became apparent that this task was not well suited for this thesis. Operative planning — a problem more prevalent and therefore more likely to yield enough feedback and thusly good results — was chosen as a main focus. Several different factors encouraged the switch of focus:

1. Only one expert had a single task in mind where he could actively test the curriculum planning prototype
2. All experts related to operative planning independently reported the need for a tool to aid semester planning
3. Curricula planning turned out to be a minuscule task in the daily business of any faculty, and in the rare case it was happening a whole task force was in charge of tackling the task, thus reducing the relevance of simplifying it even further
4. The thought of providing feedback for a tool that seemingly yielded less benefit than the effort of giving the feedback was not particularly motivating for experts
5. Experts who expressed a need for a tool immediately saw a benefit in said tool and were happy to participate in further interviews and provide continuous feedback, a crucial criterion for the method of this thesis

CHAPTER 5

Design Concepts

5.1 Prototypes

The first prototype, which was used during the CurrAG to calculate costs of the new curricula looked a lot like the prototype in figure 5.1 but without the compare section on the left, and instead a table which contained all calculated results. Beneath the sixth semester on the bottom was another table, which had each lecturer involved in the curriculum listed with all courses and total sum of students and hours she or he had to cover. This prototype was only used during the creation of the curriculum and the appearing issues were tackled in the next design iterations. There was no way of analyzing or comparing separate courses, the results were displayed as static tables and multi-user capabilities were simply not there — changes made by a client were written to the database directly and other clients could only see those changes if they would reload the page; otherwise changes were just overwritten.

The benefits of this prototype were still enough to use it during the planning, because only excel was available as an alternative, where each change would require a new calculation by hand.

The second iteration began with exploring possible scenarios and creating different solutions to tackle the problems at hand. This was done on paper, using the Five Design-Sheet methodology [12] and resulted in the low-fidelity prototypes of the Semester-View and the Tabular-View prototype, seen in figures 5.1, 5.2, 5.3 and 5.4.

5.1.1 Semester-View Prototype

The semester-view prototype is composed of three different sections, which are connected with each other. The top-left section is the selection and filter area — it controls the view of the bottom-left main section. In the drop-down selection all planned curricula are available to choose from, and beneath it the view of the main section can be switched.

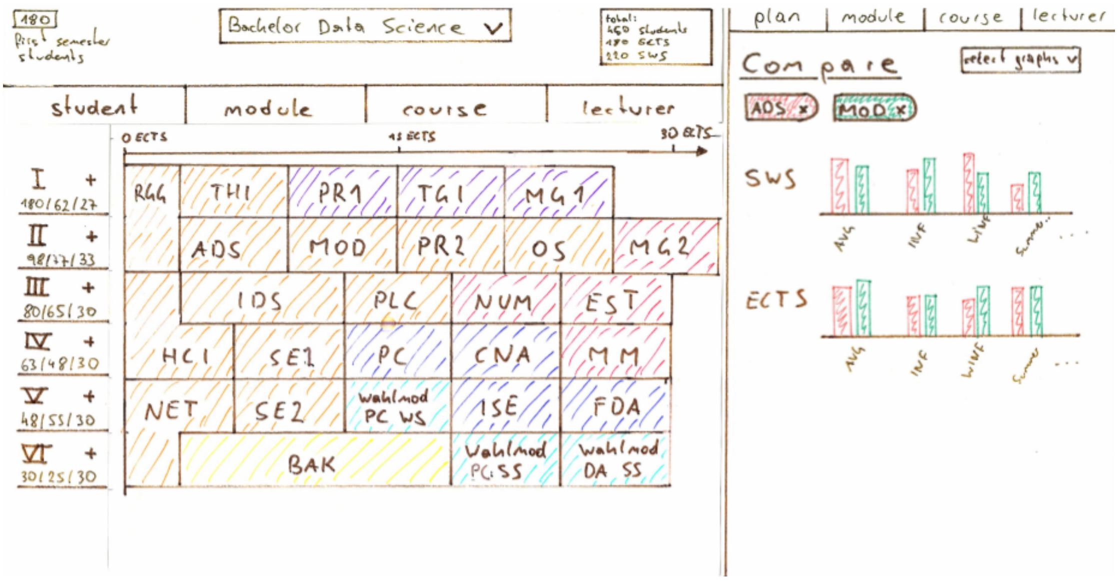

Figure 5.1: Semester-View prototype. The top left part shows a selection of different curricula as well as views; the bottom left part represents the curriculum based on the different semesters (currently in the student view); the right part is the compare section

In the main section a typical semester plan is shown, which contains all modules in the curriculum, sorted based on the semester they appear in the intended path of the curriculum. On the left-hand side of this view the semester numbers are depicted in roman numerals and the overall calculated figures for amount of students, SWS and ECTS for the semester can be seen at first glance beneath each semester index.

By selecting different views in the filter section, the x-axis or the granularity of the data presented in the main view changes. The view in fig. 5.1 shows the "student" option, which depicts modules – combinations of courses – as granularity and ECTS on the x-axis. The "module" option keeps the granularity, but changes the x-axis to SWS to show effort for each semester; the "course" option switches the granularity from modules to single courses, still with SWS on the x-axis; and the "lecturer" option, seen in fig. 5.2, plots the amount of SWS per semester for each lecturer involved in that semester. On the bottom the total amount of SWS for each lecturer over all semesters is shown in a one-dimensional distribution line, which can be clicked or circled to select one or more entries in the above semester view.

The compare section (on the right-hand side in figure 5.1) provides the possibility to compare different courses, modules, lecturers or whole curricula with each other based on their computed quantitative measures. Choices to compare can be added by dragging elements from the main view to the compare view, and based on the compared elements different graphs can be chosen to contrast different measures.

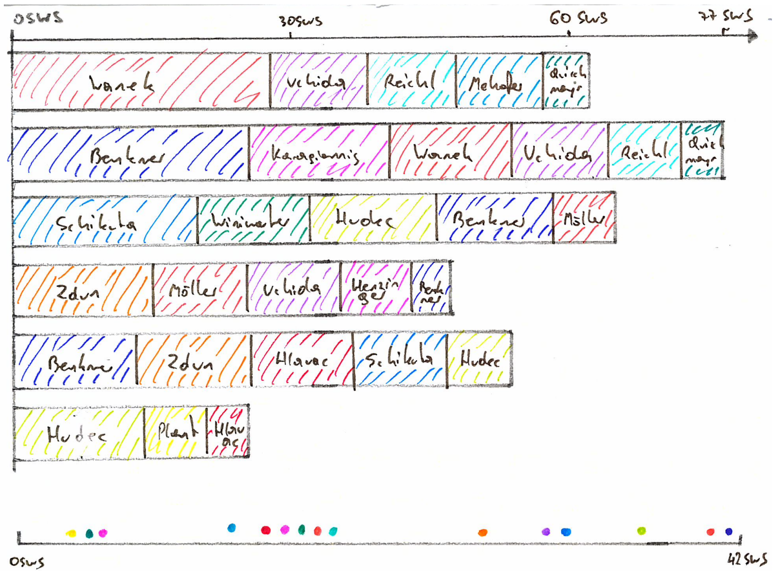

Figure 5.2: Lecturer view of the main section of the semester-view prototype (roman numerals on left-hand side not shown here)

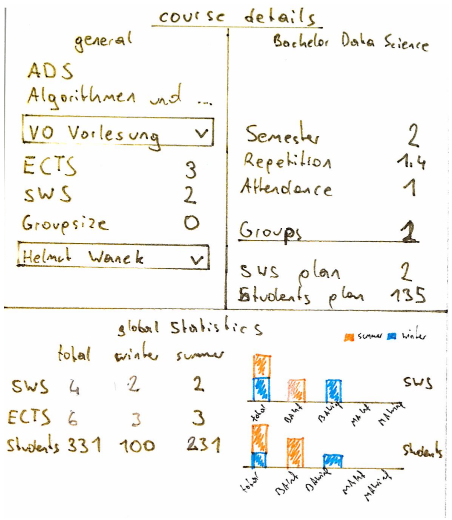

Figure 5.3: Course details pop-up of Semester-View and Tabular-View Prototype

Detail View The detail view, seen in fig. 5.3, is a pop-up view which is used in the Semester and Tabular-view prototype to show all related details of a course. It is shown when the user clicks on a course and overlays the current screen by darkening the screen and displaying centered on top of the darkened screen. It is divided in three sections, the top-left section shows general information about the course, which is constant across all curricula; the top-right section shows curricula specific details about the course (which can vary because a course can be used in different context throughout different curricula) and the bottom part shows a global statistic of the course, summing up all measures of the course over all curricula.

5.1.2 Tabular-View Prototype

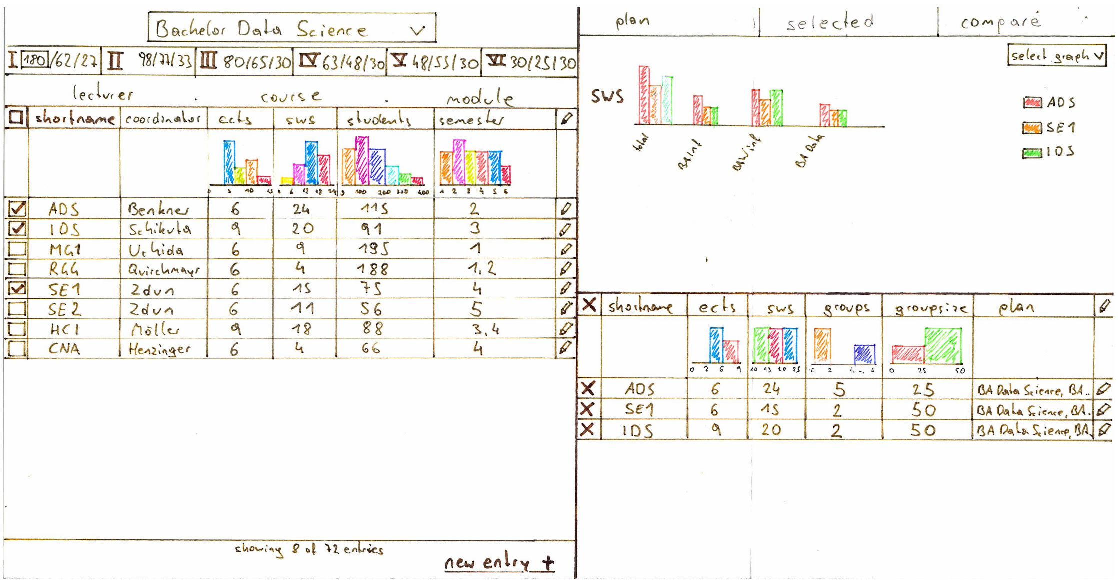

Figure 5.4: Tabular-view prototype. A curriculum selection can be seen on top of the left half, under which the complete data view is located; the comparison section is located on the top part of the right-hand side; a mirrored view of the data view to depict the current selection is located in the bottom right part

The Tabular-View prototype was designed with the fact in mind that the current planning happens fully within Microsoft Excel, but many tasks have to be made by hand in several tedious steps. It therefore seeks to mimic the functionalities of Microsoft Excel, but automates the common tasks and enhances the usability.

As seen in figure 5.4 the Tabular-View is split into two sections. The left part is the main section, the right part is the selection view. On the top of the main section is the curricula selection as well as the roman numerals for the semesters with the computed quantitative values for each semester, as already known from the Semester-View prototype.

Below this selection is the data-table which shows all courses from the current curriculum based on the settings in the table header. The data can be filtered, aggregated and sorted by right-clicking the table header. A histogram of the current view is shown in the first table row, which can be used to select groups of courses. The aggregation also allows for grouping of all courses within a semester, or all courses aggregated by the respective lecturer of the course.

Values in the data table are editable, as long as it is possible from the aggregation view, and single courses can be edited by clicking the pencil icon in the rightmost column. Rows can be selected by clicking the checkbox in the leftmost column. Selected rows are shown in the selection view, the right section of the prototype.

The top part of the selection view is similar to the compare section of the Semester-View prototype, where different curricula, modules, courses or lecturers can be compared based on their quantitative measures. The possibility to contrast two measures directly by plotting them in a scatterplot is an additional feature in this compare view, which was not present in the Semester-View prototype. This is enabled by right-clicking the table header to select a measure and then choosing "compare with..." to select a compatible second measure to draw the scatterplot.

The bottom part of the selection view has the same capabilities as the data table from the main section, but is only showing selected entries, in contrast to the main section which shows the full dataset. It can be used to show different aggregations or values of selected rows by filtering and aggregating independently from the main section. It also features the histogram in the first row to provide an overview of the selected data.

5.2 Prototype Feedback

The most frequent feedback to the Semester-View Prototype was that detailed information could only be accessed by opening another window (detail pop-up), which resulted in an interruption of the current workflow. Some other options which are not presented in detail in this thesis are also shown in separate pop-ups and the experts were not comfortable with dealing with so much different windows to access basic information.

Secondly the experts could not quite figure out what they would profit from comparing different elements based on their computed measures against each other, because that was not a task they were trying to achieve in any way. The details of a single course were more important than the comparison between two or more, because most of the courses cannot be compared to each other in a meaningful way, because each course has its own relevance independently of other courses. This observation made the compare view on the right-hand side (see fig. 5.1) obsolete.

The last part which was criticized was the strict semester-wise separation on the y-axis. For some tasks it was deemed more useful to have a separation between summer and winter semester, and for planning purposes it was even needed to show the semesters from the prior year. A switch between the six semester, two year and summer/winter separation was suggested.

In the Tabular-View Prototype the main criticism was the fact that the actual planning was already complicated to do in Microsoft Excel, so why continue bothering with the same restrictions in this new interface. This could be partly induced by the order in which the prototypes were shown to the experts during the interviews, because the

Tabular-View was always shown second, so the experts knew there was a more intuitive way of showing the data.

Another feedback was that the capabilities already existed in Microsoft Excel and therefore it would result in additional work to change the working environment without any major improvements over their current solution. The main use of the Tabular-View would be comparison and analysis of existing data, which was already possible in Microsoft Excel without a great hassle and the real problem of planning was not tackled with this solution, because it has similar limitations as Microsoft Excel.

It also showed that the grouping and aggregating of the data was not intuitive to some experts, which let them see even less improvements to their current workflow.

The general feedback from all experts – except from the Accounting and Finance department – was that the Semester-View Prototype generally depicted their own way of thinking about the courses and made it clear and easy to understand what they were looking at. They were excited to work with a tool which makes the vast amount of data accessible.

5.3 Final Design Choice

The final design incorporates different elements from the previous prototypes, but focuses on a different topic of planning, namely the operative semester planning instead of curricula planning. The design can be seen in figure 5.5 and consists of four different areas, which will be described in the following sections.

5.3.1 Filter section

The filters section can be seen on the top left of the tool in figure 5.5 (B) and control which courses are shown in the main section. The top row is an automatically generated row of checkboxes for each course type available in the underlying data and each ticked type of course is shown in the main section; all boxes are ticked by default.

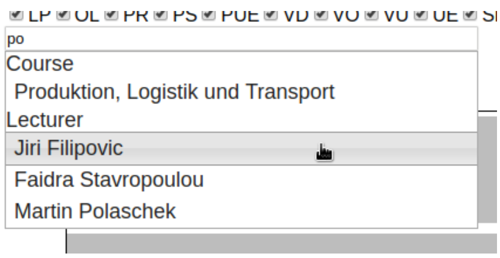

Figure 5.6: Autocomplete of searchbox with separation of results between courses and lecturers

Below the course type checkboxes is a text field to filter based on the text a user enters. It also features an autocomplete mechanism (see fig. 5.6), which shows suggestions separated in the categories "courses" and "lecturers". This filter is also capable of filtering courses which happen in a specific semester by simply typing the semester-identifier (e.g. "2016S" for the summer semester of 2016), but this is not featured in the autocomplete suggestions.

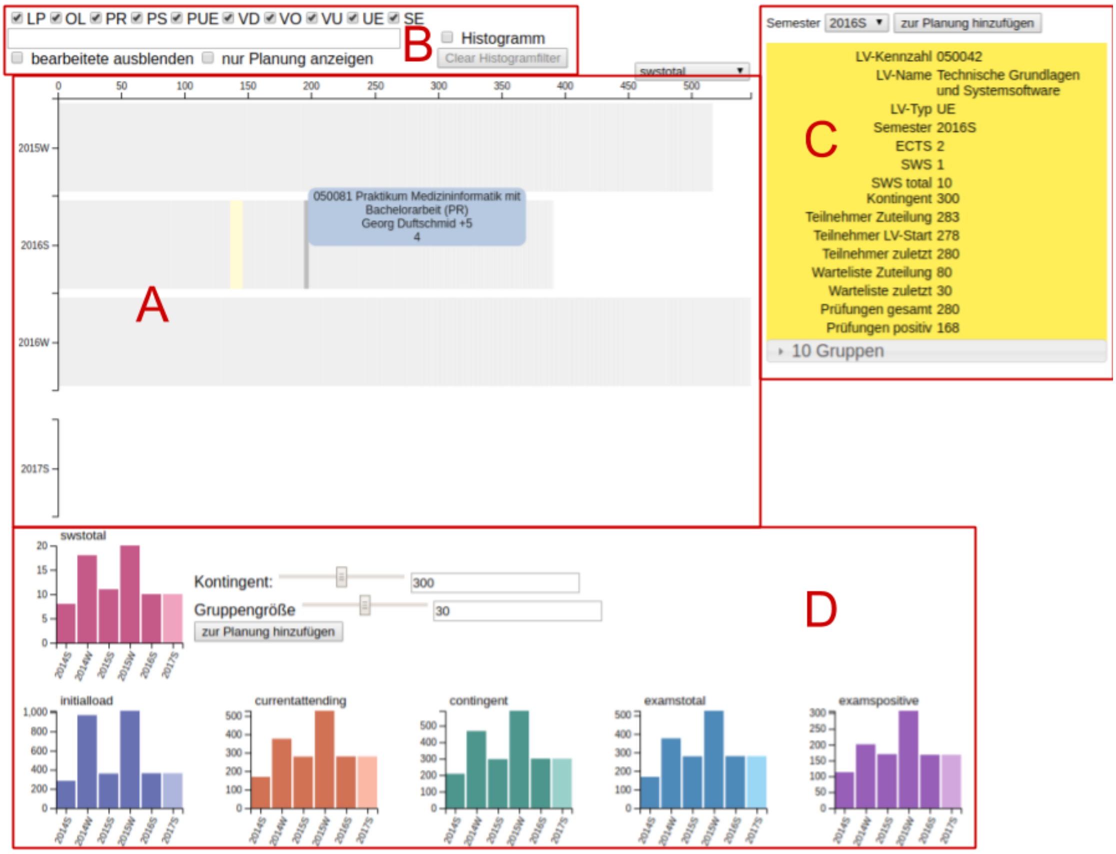

Figure 5.5: Overview of the final tool: Section (A) is the main area, (B) shows the filters, (C) is the detail-view and in (D) is the prediction section with the historic course data

The two checkboxes below the text field are used to filter only courses which are currently in the planning semester and to hide courses which are already marked "done" in planning.

Right from the text field is another checkbox which switches the main section between the general view of the stacked bars and a histogram view. The "clear histogramfilter" button below this checkbox becomes active after the user clicked a histogram-bar, thus filtering only courses in that specific range of the histogram. Clicking this button removes the currently active histogram filtering.

5.3.2 Main section

The main section is the major focus point of the user (see fig. 5.7) — it is used for data exploration, overview, and planning. It shows a visual representation of all courses of the chosen faculty of the past three semesters as well as the following semester. The y-axis of the view can be set by defining configuration variables whilst setting up the tool and can be easily adapted by changing the configuration on the backend, but it does

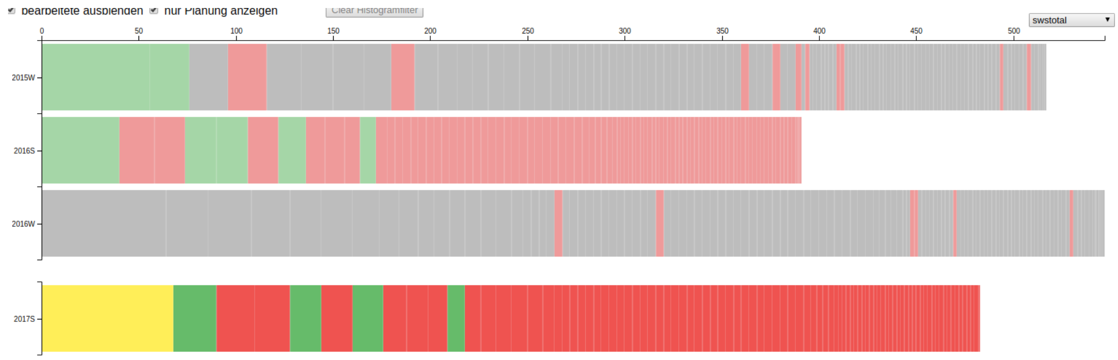

Figure 5.7: Main section of the tool showing the past three semesters and the fourth semester on the bottom, which is currently being planned

not change during the usage of the tool. The x-axis can be changed to any numerical value present in the data, either static or computed. This is done by changing the value of the drop-down selection in the top right corner and immediately transforms the view to plot the new setting.

Four stacked bars are present in this view, the first three bars depict the gathered data from the current semester (third bar), and the two semesters prior to the current one (first and second bar) and are close to each other, and the fourth bar represents the next semester, which is currently being planned (planning view).

Each bar is made of multiple segments, which represent individual courses in the respective semester. This representation was chosen because it is the closest representation of the expert's mental model of a semester. Each semester contains all courses of the chosen faculty, because a separation between curricula was not necessary, as each active course needs to be present in the following semester regardless of the curricula it is contained within. The width of each segment represents the corresponding value to which the x-axis is currently set and courses are sorted descending in width within the semester. Possible values for the x-axis are:

1. planned number of students
2. number of students attending course at the beginning of a semester
3. number of students attending course at the end of the semester
4. number if students on the waiting list at the beginning of the semester
5. number if students on the waiting list at the end of the semester
6. total number of exams written in course
7. number of exams written in course that had a positive result

8. SWS of a single group of a course
9. accumulated SWS of all groups of a course
10. ECTS of a single group of a course
11. accumulated ECTS of all groups of a course

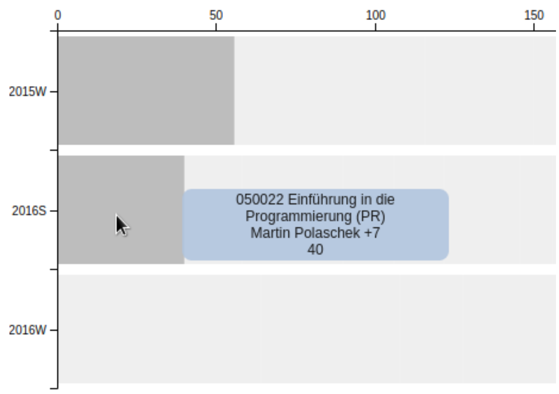

Figure 5.8: Mousing over a segment in the main section

Mousing over a segment in the main section highlights all instances of the same course in all other semesters as well as showing basic information about the course in a tooltip. This information contains the course identifier, name of the course, type of the course, lecturer and currently chosen x-axis value, as seen in figure 5.8.

The colors of the sections shown in figure 5.7 provide an overview of the current situation at the first glance. A grey section is currently not active, neither in focus, nor present in the current planning. Red sections are currently present in the planning view and not yet marked "done". The yellow section has been clicked last and therefore is currently in focus — its details are shown in the detail view and the prediction section and actions like marking done or changes of predictions result in changes on the yellow highlighted course. Green sections are – like red sections – also currently in the planning view, but already marked done.

A planner might choose to hide all "done" courses from the main view by selecting the respective filter and thus has only to focus on the red sections. After all planning is done only a few grey courses are left in the main view, which therefore are easy to grasp and filter for potential missing or left out courses.

Another option is the "histogram" view which can be activated in the filter section on the top of the page. It creates a histogram of each semester based on the value selected in the top right drop-down selection. The number of bins for the histogram can be set in the configuration on the backend which defaults to five. Each bar in the histogram view (see fig. 5.9) contains a label in the bar to show the number of elements in the respective

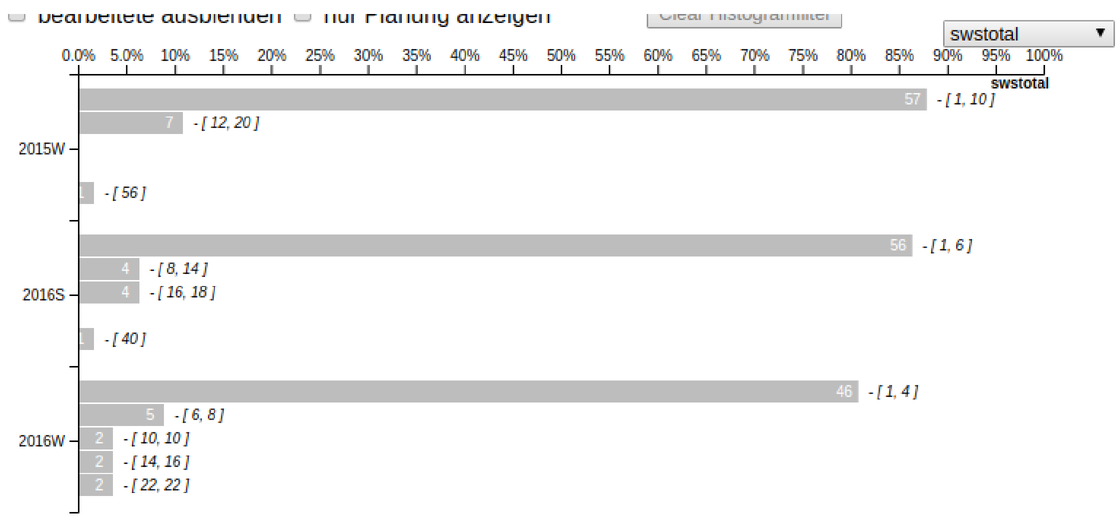

Figure 5.9: Histogram view of the main section

section of the histogram and an interval after the bar, which shows the minimum and maximum value contained in the section.

Mousing over a bar shows a tooltip containing the names of up to three courses contained in the bar, as well as the exact percentage represented by the bar. By clicking a bar a filter is applied which reduces the courses shown in the main section to only the courses present in this section of the histogram and switches back to the stacked bar view. By selecting the histogram view again multiple iterations to narrow down the results can be made. The applied filter can be reset by clicking the "clear histogramfilter" button below the histogram selection checkbox.

5.3.3 Highlighted Course View

The highlighted course view is shown whenever a segment in the main section is clicked. As seen in figure 5.10 it displays all relevant static and calculated data of the chosen course. On the bottom is a menu which can be opened to show the different groups of a course and each group contains its own respective values as well as the lecturer responsible for the group.

Above the highlighted course view a single row is shown which enables the planner to choose a semester from the main view and add all courses in that semester to the planning view. This was implemented based on the expert feedback, which stated that the first step of planning usually consists of copying the previous semester and then adapting the courses according to the needs of the next semester.

Clicking a segment in the main view does not only show the details of the chosen course, but also loads the course as a prediction in the prediction section.

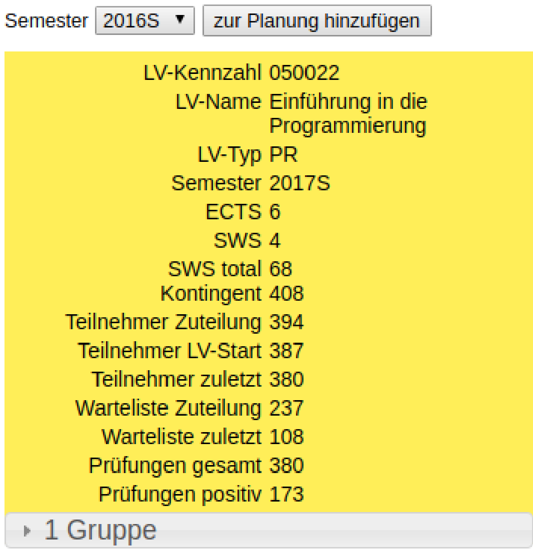

Figure 5.10: The highlighted course view displays all relevant factors of the currently selected course

5.3.4 Prediction section

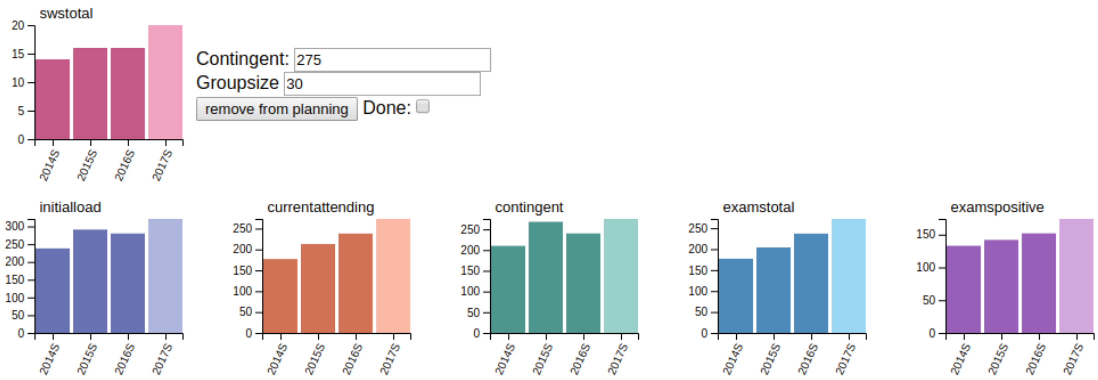

Figure 5.11: Planning section with selected course. Predicted values are represented by in a lighter color in each chart.

The prediction section (fig. 5.11) consists of six bar charts and sliders to change the prediction according to the needs. The bar chart in the first row shows the predicted SWS as a measure of cost, the five charts in the bottom row show the total amount of students, which is why they are displayed in the same row.

Historic data about the course is displayed in a solid color, the last bar in the lighter color shows the value of the prediction. The prediction is based on the number of planned places within the course. Each value is calculated independently based on their dependency;

some calculated values are based on other previously calculated values, thus cascading the dependencies. Initial load — the amount of students trying to attend the course at the start of the semester — is calculated by dividing the initial load of the previous iteration of the course by the previously planned number of seats, and multiplying it by the currently planned number of seats (called "contingent"). The value current attending — the amount of students attending the course at the end of the semester — is calculated the same way as the initial load. The total amount of exams is calculated depending on the amount of students attending at the end of the semester, and the amount of positive exams depends on the total amount of exams.

The amount of SWS is dependent on the groupsize and the planned number of seats, because a new group needs to be created until every student gets a place. By changing the values of the contingent and the groupsize with the two sliders in the top row the prediction can be adapted to fit the planners estimation.

The prototype had all bars as a grouped bar chart, grouped by semester, combined in a single view. This design did not show trends in the individual values throughout the semesters, because the semesters were too far apart from each other visually to recognize trends and correlations efficiently. Thus the final design features individual charts for each value to grasp trends and easily compare similar values with other semesters. When hovering over a bar in either chart a line gets drawn at the height of the hovered bar in each chart to ease comparing values even more. The hovered bar and all bars in the same semester are additionally highlighted and their respective value is shown within the bar to also allow comparison of different values (shown in fig. 5.12).

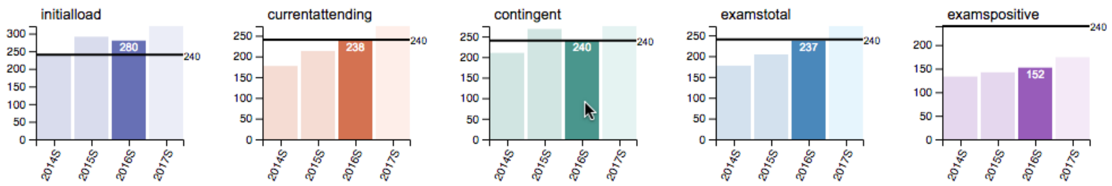

Figure 5.12: Hovering over a bar displays a line at the height of the bar in each chart.

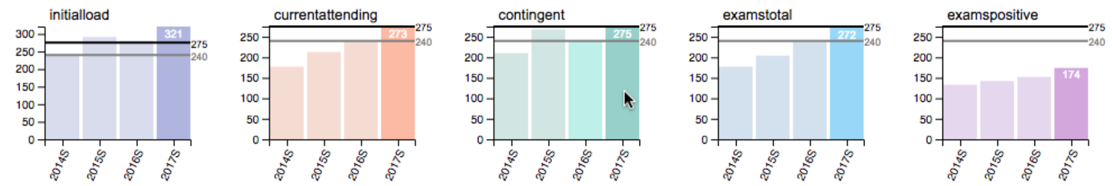

Figure 5.13: Clicking on a bar adds a static line to all charts (grey) which can be used to compare the chosen value with the hover line (black).

Clicking on a bar keeps (see fig. 5.13) the clicked bar highlighted (in a different highlighting color shade as the hovering highlight) and permanently adds the line at the height of the clicked bar in a dark gray color to all charts. This can be used to compare two different values from different semesters by clicking the first bar and then hovering over the second one.

5.4 Design Conclusion

The final design uses many different aspects of the prototypes which received positive feedback and improves or removes parts which received negative feedback.

The main view is plotted as a stacked bar, because that view has the closest resemblance with the mental image of a semester for an SPL. A static y-axis fits the purpose of planning and does not introduce additional complexity, whereas the x-axis needs to be flexible in order to cater all needs of exploring and planning. Granularities beyond the simple course view were omitted, because the general approach in planning a semester did neither regard modules nor lecturers as separate entities.

A positive feedback was received regarding the histogram row in the tabular prototype; it was therefore chosen to incorporate this view into the main section as well and provide the capability of filtering by selecting regions in the histogram.

Comparing two or more courses was not regarded useful, which made this functionality obsolete and freed space on the right-hand side to statically display course details without opening a separate pop-up view, thus not switching the view completely and interrupting the planners work.

Grouped bars were replaced by small individual charts to create a better overview and represent trends in the data more obviously.

Color coding was initially based on the color scheme of the semester plan, soon changed to a wider range of colors to better differentiate between individual courses. The wide range of colors proved to be hard to grasp for the users and did not ease differentiation, so it was replaced by a gradient which should help color code the width to better visualize differences between two values in the main view. It turned out to be rather insignificant, because the few "large" courses at the beginning of the gradient had the effect that all "smaller" courses still had a similar color. The final color scheme is based on the task of planning itself — it shows the courses which are currently in the planning view and the status of the respective course, to quickly give an impression of the progress of the planning.

CHAPTER 6

Implementation

The first prototype was written in JavaScript with the help of the QueryUI to manipulate elements. Its database was a MySQL database, which was accessible via PHP handlers for all basic CRUD functionalities. All calculations due to changes and the resulting statistics were generated locally at runtime on the user's client browser and instantly written to the database afterwards. This resulted in a couple of issues:

- The speed of the application was limited by the size of RAM and power of the CPU of the client, or even the part of processing power the browser was capable of handling.
- Each new calculation — thus each action where data changed — required an update in the database, which blocked the application until the data was written. Therefore a large number of successive updates took a considerable amount of time for the database to process, which was an issue because relatively small changes from the user could result in a large amount of changes in the database.
- The tool was not usable for more than one user at a time, because updates in the data were not pushed to other clients, but only loaded when the client refreshed.
- Updates issued from different clients on the same object simultaneously were not handled, so only the latest change on the object was stored in the database.

Even though these issues could be circumvented in the course of the CurrAG where this prototype originated, it was clear that the resulting tool needs to tackle all issues in order to guarantee consistent data handling and deliver correct results.

JavaScript - Web-based scripting language for dynamic contents, executed in the browser on the client

PHP - a stateless server-side script language used for creating dynamic web pages

jQuery and jQueryUI - a JavaScript framework for creating interactive user interfaces (e.g. mouse related movements, animated resizing, etc.)

D3 - a JavaScript framework for data visualization, used to plot graphs and related visualizations on web pages

Node.js - server-side JavaScript runtime environment using Google's V8 JavaScript engine, capable of handling large number of synchronous connections and supplying real-time communication

socket.io - JavaScript library utilizing the WebSocket protocol for event-driven real-time communication

MySQL - relational database management system

Figure 6.1: A short reference of technologies used in the implementation of this thesis

6.1 Backend

The backend is completely written in Node.js rather than PHP to be able to handle multi-user access, push data to clients and be easily scalable to larger amounts of concurrent users. Structurally the backend is designed using the hierarchical layers design pattern [15] which separates different parts of the code in independent units (see Fig. 6.2).

6.1.1 Data Access Layer

The data access layer is responsible for all interactions with the database. At program start it loads the current dataset into the program memory by storing it in internal class variables. It is also responsible for creating any missing tables in the database if needed (e.g. first server-start).

After loading the data a separate thread is started, which is responsible for writing changes to the database. It accomplishes that by monitoring all changes and writing them to the database each full minute (time can be configured at server-start in millisecond-intervals). This timed writing behavior results in limited disk access, thus not flooding the disk with large amounts of write operations when multiple changes on the data appear in short succession. The interval in which it updates the changes can be seen as a trade-off between disk performance and reliability — the time set is also the maximal interval from which data can get lost in case of a server crash.

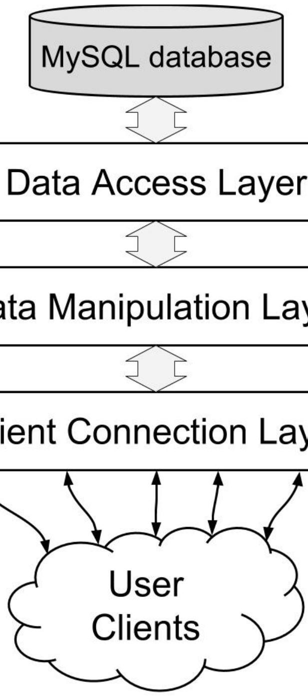

Figure 6.2: Layer structure of the Node.js backend server

The data access class provides Getter and Setter for all class variables to allow the data manipulation class access to the current data.

6.1.2 Data Manipulation Layer

The data manipulation layer handles all calculations and other manipulation options needed to further process the data. Every statistic, metric, clustering, re-grouping is done in this class. The idea behind centralizing the data manipulation is a simple one — there can only be one version of each manipulation mechanism, whereas if the manipulation would be done on the client-side, it could happen that older or cached versions of algorithms would be used.

It also tackles the problem of limited RAM or CPU for large data calculations on the client-side — the calculation is now only limited to the server's specifications, which is better suited for large scale calculations due to its architecture and can easily scale to provide the required amount of RAM and CPU to achieve the optimal performance.

Every change of data is stored by accessing the setters of the data access class and distributed to the users by pushing the changes to the client connection layer.

6.1.3 Client Connection Layer

The client connection layer implements the socket.io library to handle communication between clients and the server. Socket.io is a JavaScript implementation of the WebSocket protocol [5], which allows for a two-way communication — allowing the server to push updates to the clients directly. The WebSocket protocol basically works like a regular socket protocol; it opens a port to the server on which the communication happens, only if the webpage is closed the socket gets closed. This enables real-time updates without the need to reload the page or sending continuous requests.

In the client connection class a pool of opened sockets is managed, and all requests are forwarded to the data manipulation class alongside the socket of the requester. It also provides three interfaces for the data manipulation layer to access:

1. **send to single socket**
   used to answer single requests to a user (e.g. GET)

2. **send to all sockets**
   used to push updates in the data relevant to all users

3. **send to all sockets except one**
   used to push updates to all users except the one who issued the update

The data manipulation layer can then decide which of the above options applies and send the updates to all relevant users.

6.2 Frontend

No actual data manipulation happens in the frontend; it is an interface to view the data, which is also capable of creating new data.

Due to the nature of JavaScript the frontend is not strictly confined into classes as such, but still structured in different components by the means of the model-view-controller pattern [15].

6.2.1 Data Handler

The data handler component (see Fig. 6.3) acts as controller in the frontend — it connects to the server to send and receive data and sets up all possible user interactions after receiving data. This connection is achieved by starting a socket.io client for communication and defining all necessary socket handlers.

Any selections made by the user are then forwarded to the filter and restructuring component.

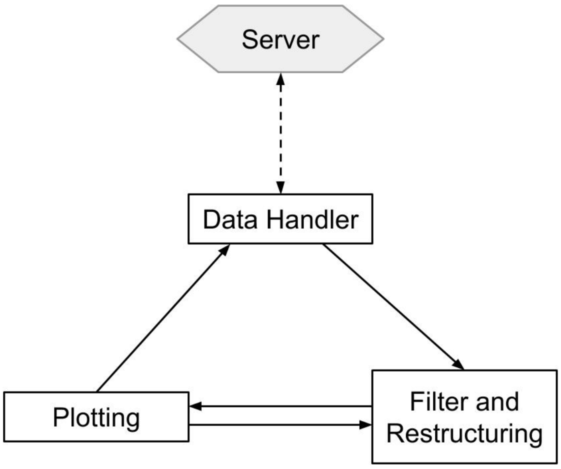

Figure 6.3: Component structure of the frontend JavaScript application

6.2.2 Filter and Restructuring

Any filtering applied by the user is received by the filter and restructuring component, which acts as the model in the MVC-pattern of the frontend. Data is never changed in this component, but only gets fitted to comply with the needs of the selected view — a stacked bar chart does need a different structure in the data to get plotted than a histogram, or a regular bar chart. All fitting and filtering happens in this component before the data gets handed to the plotting component.

6.2.3 Plotting

The plotting component receives the data and uses D3 to generate different diagrams based on the data. These diagrams incorporate events (e.g. "onclick", etc.) which get forwarded to the controller if they generate new data, or get sent to the filtering component if they change the range of data viewed. This is a slight break with the basic MVC-pattern, but needed in this context because of the dynamic nature of the diagrams generated.

CHAPTER 7

Evaluation

7.1 Methodology

Four domain experts took part in the evaluation, which were also part of the design process. Each evaluation took place individually. At first they were given an introduction into all the functionalities of the OCP tool which lasted for approximately half an hour. After that they were given the tasks of of selecting particular courses based on the name of the course as well as the lecturer, estimating the needed contingent for the following semester and adding those courses with the right estimations to the planning view. After completing these tasks they had some time to interact with the tool, which they used for general explorations of their data-sets. The interactions with the tool were supervised, such that the usage was monitored and appearing questions while using the tool could be answered.

After about an hour of using the tool the experts were interviewed regarding their opinion of the interface for its usability as well as the possibility of including the tool into their current workflow.

7.2 Results

The interface of the final prototype was generally regarded as easily understandable and intuitive to use. All users mentioned that they had never seen a tool with capabilities like this before. Four of the experts mentioned that it astounded them to have so much information compressed in a single interface, which is still easy to understand as well as to use. Three of the four experts from the field of planning asked whether they could gain immediate access to the tool because they wanted to use the information in one way or another in their current planning steps. On the negative side three users mentioned that a better naming concepts for the different depicted parameters could be found, two participants mentioned a missing filter between Bachelor's and Master's courses and one expert found it counter-intuitive that a task which is highlighted and switched to "done" would stay yellow, rather than switch green.

An additional request was also made by two experts, which stated that they wanted to group courses based on the content of the lecture. Content of a lecture is an additional classifier which is stated within the curricula and has an ID for each content.

When asked whether the experts could see themselves using the tool in their current workflow of planning each one of them found it highly plausible. They came up with different scenarios based on their own respective workflow.

- The SPL of Computer Science is going to use the tool to copy the previous semester, adjust the contingent if needed and hand the resulting list of courses, their contingent and groupsize to the StudiesServiceCenter to gather missing details from the lecturers and enter the results into the database of the university.
- The SPL of Business, Economics and Statistics as well as the SPL of Chemistry both would split the resulting list between the different departments of their faculties and then send each department the part of the list they are responsible for. The feedback from the departments then concludes the planning and a member of the StudiesServiceCenter can again enter the courses into the database of the university.
- The SPL of History would build an interactive website based on the list of planned courses resulting from the tool, where each individual group is represented. Lecturers would then get a certain amount of time to inscribe themselves for all groups they plan on holding in the following semester, not being able to choose from groups already taken. A first come, first served principle would therefore be established, which could further motivate lecturers to think about planning for themselves.

As shown all experts are keen on integrating the tool and the results from the tool into their workflow and are currently using the tool for extensive testing purposes.

7.3 Conclusions

Two of the experts are currently still using the tool in their daily workflow which shows a really positive reception of the tool altogether.

A different level of detail and aggregation based on the content of a lecture is the most requested feature which will further be discussed in the future work section of this thesis.

A better way of expressing the possible values of the dataset (e.g. more easily understandable names for variables) and a more intuitive color pattern might help the overall interface in terms of understandability and usability.

All experts have different ways of incorporating their planning results from the OCP tool into their workflows, which makes the introduction of an API to access the results a very useful addition to the tool, because each SPL can process their resulting data individually according to their needs (e.g. putting it on a website).

CHAPTER 8

Discusson

8.1 Lessons Learned

The main issue encountered during this project was right in the beginning, where the tool started not as a design study, but was created in a hurry to compensate a need in planning a new curriculum in the CurrAG. There was an obvious gap in the workflow of designing a new curriculum, which could obviously be filled with the tool, because it was purpose-built to do so.

Different issues emerged during the usage of the quick prototype in the CurrAG which were already stated in the previous chapters, so the next step was refactoring the tool to include multi-user features, data consistency checks, better visualizations and usability featured. After quite some time went into the refactoring it was decided that the tool could be the basis for a master's thesis.

It was decided that a design study was the methodology to be used for the thesis and a design process to create enhanced prototypes as well as different representations to accomplish the task of curricula planning were designed. After the final iteration of an extended design period using the five design-sheet methodology [12] the first expert interviews were scheduled. To this point the only expert involved was the supervisor of this thesis, who was also part of the CurrAG.

The first iteration of expert interviews discovered a fact which did never occur to us during all previous steps — even though the domain experts liked the tool we created and appreciated its design, nobody could really use it, because it filled a gap in a workflow, which was irrelevant to the daily work of the domain experts. Only a single domain expert, the SPL of chemistry, saw a chance of him using it "in about a year's time", because he was planning on reworking one of the master's curriculum at his faculty.

After scrapping the first idea and reworking the tool to tackle an actual need in the daily business of an SPL and therefore following the design study methodology I discovered that Sedlmair, Meyer, and Munzner [14] already identified the very problem encountered in the early stages of this thesis as a reoccurring pitfall while conducting design studies.

Design lesson A considerable amount of time was spent during this thesis trying to come up with a different view for the data to present a choice of different visualization methods to the domain experts — the tabular view prototype was created, which was fully capable of planning a semester distribution in a new curriculum. No domain expert involved in operative semester planning actually preferred this view in comparison with the semester view, which displays individual courses as rectangles in a grid representing the different semesters — a view which mimics the way a semester table is usually displayed. Identifying the mental model of the situation beforehand could have saved the effort of creating completely different views of the data, and the time could have been spent on developing different other views based around the semester grid. Based on this observation I would recommend to start with even more low-fidelity prototypes than shown in the Prototypes section of the Design Choice chapter — basically rough sketches of designs with neither a deeper meaning to individual elements in the sketch nor considerations about fitting the full functionality into the sketched design. After this first iteration with the plain sketches the design space would be narrowed down to start the second iteration with the five design-sheet methodology [12] to fit all possible features in the given design space.

8.2 Future Work

8.2.1 Improving the tool

Further improvements as already stated in the Final Evaluation section can be made to enhance the planning capabilities and representation of the courses in the OCP tool. A design change that groups courses by the content of the lecture is also possible, but is a non-trivial change to the current tool. It introduces another layer of abstraction on top of the course view, because a lecture content can be present in different courses within a particular semester.

8.2.2 Agent based model and simulation

To get a better understanding of the interactions happening within a curriculum I want to create an agent-based model and simulation of the whole curriculum and all its participants. This requires to simulate all courses, lecturers and individual students and put them in an environment where modeled students can act as real students could. By fitting simulated students as closely to real students as possible through incorporating real student data the results of the simulation should be able to create realistic future scenarios.

This kind of simulation can be used to evaluate worst-case scenarios in the current curricula, simulate the impact of changes to the curriculum and help predict student behavior for the following semester to detect possible bottlenecks before they appear in reality.

Feeding the results from the simulation into the operative planning tool can lead to more precise predictions and less manual tweaking effort while using the tool.

8.2.3 Financial planning

Calculating cost of any curriculum, regardless of it being already realized, or currently in planning is a complex task. A tool which not only helps the calculations, but also contains a possibility to make finances more easily comprehensible, shows the reasons for different costs and automatically determines whether this cost is justifiable or rather should be looked at in detail could vastly improve the workflow of the finance and accounting group.

Another benefit of an easily understandable tool is the fact that planners can also use at least parts of this tool to evaluate their plans financially before completing them. This would reduce the friction between planners and finance and help both workflows.

8.2.4 Different analysis components

It is basically possible to choose any of the tasks stated in section 4.2 of this thesis and base some kind of future work on it, as all tasks are currently unsolved problems which domain experts are facing in their daily work.

As one of the tasks to base a future project on I want to highlight the task of data exploration. The supervisor of this thesis is not directly involved in operative planning, but in his function as a vice dean of teaching in constant touch with the same data. His different view on the dataset results in a need for a tool to perform exploration, analysis and evaluation tasks, which include a lecturer centric view, grades and the fill-factor of courses amongst many others. It is therefore planned to create a tool with different capabilities fitted to the needs of a vice dean instead of the needs of a SPL.

CHAPTER 9

Conclusion

This thesis provides a detailed explanation on the domain of university planning, specifies stakeholders, identifies both problems faced within the domain as well as tasks that emerge from those problems. It can therefore be used to identify possibilities for research topics and build a foundation of knowledge for new studies in the same field.

Furthermore a single task from the domain was chosen as focus for creating a design study. Through several iterations an optimal design solution for the problem of operational semester planning was found and a tool was created which aids the workflow of SPLs whilst planning a future semester. Different designs were evaluated and the decision on the final design was documented, as well as the details of the implementation that were needed to create the tool. No public prototype of the final tool is available, because the tool incorporates sensible real world data.

OCP is targeted at domain experts related to planning, therefore is based on their mental model and aids them throughout the whole process of predicting accurate student numbers. Its main view provides a large range of filtering and sorting options which result in different visual representations of the overall dataset to effectively pinpoint certain segments of the dataset. Selected courses are shown in much detail which also includes historical data of the course.

A typical workflow in the tool consists of (a) selecting a course, (b) predicting load of that course for next semester, and finally (c) adding the prediction to the final result. Automated predictions are created based on the values of the previous semester, so that the domain experts can either reuse the planning from the last semester, or tweak the prediction according to their estimations. By statistically predetermining predictions the task of domain experts calculating predictions on their own is completely circumvented, thus reducing it to only tweaking existing predictions.

Evaluation showed a high level of functionality, the expert users stated that the insight into the historical values of courses and the easy way of tweaking pre-calculated predictions had a very positive impact on their workflow both in time it takes to come to a prediction as well as accuracy of the prediction. They also found that OCP gave them novel ways of viewing and exploring their data, giving them new insights and a better overview.

Bibliography

[1] Maryam Booshehrian et al. „Vismon: Facilitating Analysis of Trade-Offs, Uncertainty, and Sensitivity In Fisheries Management Decision Making“. In: *Computer Graphics Forum*. Vol. 31. 3pt3. Wiley Online Library. 2012, pp. 1235–1244.

[2] Michelle Borkin et al. „Evaluation of artery visualizations for heart disease diagnosis“. In: *IEEE transactions on visualization and computer graphics* 17.12 (2011), pp. 2479–2488.

[3] Jess Boronico. „Quantitative modeling and technology driven departmental course scheduling“. In: *Omega* 28.3 (2000), pp. 327–346.

[4] Matthew Brehmer and Tamara Munzner. „A multi-level typology of abstract visualization tasks“. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2376–2385.

[5] Ian Fette. „The websocket protocol“. In: (2011).

[6] Johan Havås et al. „Modeling and optimization of university timetabling-a case study in integer programming“. In: (2013).

[7] Daniel A Keim. „Information visualization and visual data mining“. In: *IEEE transactions on Visualization and Computer Graphics* 8.1 (2002), pp. 1–8.

[8] Daniel A Keim et al. „Challenges in visual data analysis“. In: *Tenth International Conference on Information Visualisation (IV'06)*. IEEE. 2006, pp. 9–16.

[9] Alireza Rashidi Komijan and Mehrdad Nouri Koupaei. „A new binary model for university examination timetabling: a case study“. In: *Journal of Industrial Engineering International* 8.1 (2012), p. 28.

[10] David Lloyd and Jason Dykes. „Human-centered approaches in geovisualization design: Investigating multiple methods through a long-term case study“. In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2498–2507.

[11] Sanja Petrovic and Edmund K Burke. *University Timetabling.* 2004.

[12] Jonathan C Roberts, Chris Headleand, and Panagiotis D Ritsos. „Sketching designs using the Five Design-Sheet methodology“. In: *IEEE transactions on visualization and computer graphics* 22.1 (2016), pp. 419–428.

[13] Michael Sedlmair. „Design Study Contributions Come in Different Guises: Seven Guiding Scenarios“. In: *Proceedings of the Beyond Time and Errors on Novel Evaluation Methods for Visualization.* ACM. 2016, pp. 152–161.

[14] Michael Sedlmair, Miriah Meyer, and Tamara Munzner. „Design study methodology: Reflections from the trenches and the stacks“. In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), pp. 2431–2440.

[15] Walter F Tichy. „A catalogue of general-purpose software design patterns“. In: *Technology of Object-Oriented Languages and Systems, 1997. TOOLS 23. Proceedings.* IEEE. 1997, pp. 330–339.

[16] Melanie Tory and Torsten Möller. „Human factors in visualization research“. In: *IEEE Transactions on Visualization and Computer Graphics* 10.1 (Jan. 2004), pp. 72–84.