Computer Science Education in Secondary Schools –
The Introduction of a New Compulsory Subject

PETER HUBWIESER, Technische Universität München

In 2004 the German state of Bavaria introduced a new compulsory subject of computer science (CS) in its grammar schools (Gymnasium). The subject is based on a comprehensive teaching concept that was developed by the author and his colleagues in the years 1995-2000. It comprises mandatory courses in grades 6/7 for all students of grammar schools and in grade 9/10 for the students of the science & technology track of this school type. In grade 11 and 12 there are elective courses that qualify for an optional graduation exam in CS. The first students that have attended the course in total graduated in 2011. This paper describes the whole project in the form of an extensive case study that is guided by the Darmstadt Model, which was developed as a category system for computer science education in secondary schools by a working group at the ITiCSE 2011. This case study is the first (nearly) all-embracing discussion of the whole project that describes the long way from the original concept to the first graduates as well as the internal structure of the subject and the first results.

Categories and Subject Descriptors: K.3.2 [Computer and Information Science Education]: Computer Science Education, Curriculum

General Terms: Human Factors, Design

Additional Key Words and Phrases: Compulsory subject, Object-oriented modeling and programming, Objects-first, Darmstadt model

ACM Reference Format:
Hubwieser, P.: DOI =

1. INTRODUCTION

In their ACM/CSTA study [Wilson et al. 2010] Wilson et al. investigated the implementation of the ACM-K12-Standards [Tucker et al. 2006] in the US states. At p. 24 the authors draw the following conclusion:

“Our research has shown that most states are focused on lower-level skills instead of deeper computer science concepts and capabilities. However, this is not enough in the 21st Century“ (p. 24)

This statement is underlined by the very different grade of adoption of the standards of the three categories concepts, capabilities and skills that have been defined by the US National Research Council [US National Research Council 1999]. The standards of the category skills (100%) are clearly dominating, compared to capabilities (60%) and concepts (37%).

Author’s addresses: P. Hubwieser, Technische Universität München, TUM School of Education, Fakultät für Informatik, Boltzmannstr. 3, 85748 Garching, Germany.
Peter.Hubwieser@tum.de. www.ddi.tum.de

© ACM, 2012. This is the author’s version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in ACM Transactions on Computing Education (TOCE), (VOL 12, ISS 4, (November 2012)) http://doi.acm.org/10.1145/2382564.2382568
Additionally, the same research detected that there seems to be a popular confusion about the proper subject area for the implementation of computer science education:

“Forty states count an upper-level computer science as an unconstrained general elective credit. This means that computer science courses satisfy no particular category and thus fall into the ‘general credit toward graduation’, as would a course in any random subject. Nine states allow computer science to count as a mathematics credit. Seven as an advanced mathematics credits. Only one state (Georgia) counts computer science as a science credit. Six states defer the mapping of course credits to individual school districts whose resulting categorization of computer science courses ranges widely”(p.44).

The situation in Europe, at least in Germany, is quite similarly confused. In 2007 the German business association BITKOM conducted a survey that showed that 78% of the parents were supporting a compulsory subject of computer science in the grades 5-10 of German secondary schools [Forsa 2008]. In the face of this public opinion it seems amazing that until now only very few of the 16 German federal states have implemented such a subject, e.g. Saxony, Bavaria or Mecklenburg-Vorpommern [Starruß 2010].

In my opinion, one of the reasons for the inactivity of the educational administrations of the other states might be the permanent confusion about the intentions of a subject “computer science”. While the intention of BITKOM probably lies on vocational education in order to educate future IT-professionals, the parents might be worried about the dangers of the Internet, the CS faculties of the universities might want to recruit gifted students, while the educational administrations might aim mainly to improve the level of general education in school. Additionally, there is a continuing disagreement about the possible learning content of such a subject: learning to program (if at all, in which language), acquiring user skills of standard software, searching information in the Internet or learning to solve complex problems? Apparently we are missing a theoretical framework that might help to clarify these questions and to mediate in these debates.

The development of such a framework was the goal of the working group “Informatics in secondary Education” (WG ISE) at the ACM-ITiCSE conference 2011 in Darmstadt, Germany. At the end of the conference the group presented the “Darmstadt Model” as a first proposal for such a framework [Hubwieser et al. 2011]. The long-term goal of WG ISE is to investigate the different conceptions, circumstances and consequences of CSE in secondary schools and to trigger an international discussion about this topic that is based on theory and evidence instead of opinions and beliefs.

In this paper, I will discuss the recently introduced compulsory subject of Informatics in Bavarian secondary schools (more precisely in two types of schools) in the form of a case study that is guided by the Darmstadt Model in the sense of a theory. Some of this information was already published in the report of the WG ISE [Hubwieser et al. 2011], but in much less detail and, as we were not able to discuss the situation in the different countries adequately there, among information about other countries without any references to the specific country. Against this background, this paper has three different intentions. Firstly, it illustrates the structure and the categories of the Darmstadt Model, applied to a specific situation. Secondly, it might serve as a guidebook for the design and implementation of an intended subject of computer science, presenting a clearly defined, high-level oriented education strategy, following the objects strictly first approach. This study might demonstrate under which preconditions such a strategy could work. Thirdly, this report might be regard-
ed as the concluding and comprehensive description of the largest project of my life, aiming to implement computer science as a compulsory subject in the secondary schools of my home state.

The project had started in 1994 with the development of the first idea of the teaching concept, published in [Hubwieser and Broy 1996], and was finished in a certain regard in 2011 by the graduation of the first age cohort that has attended the whole course. Although some facets of this project have already been published, e.g. aspects of the teaching approach in [Hubwieser et al. 1997], [Hubwieser 2003], [Hubwieser 2004] and [Hubwieser 2006] or the learning objectives in [Hubwieser 2008], this paper could represent an integrative framework that compares, extends and integrates all these previous publications about it.

As the term Informatics in the educational context is understood quite differently in several countries [Hubwieser et al. 2011], I will use computer science (CS) instead throughout this paper, referring to the definition in [Wilson et al. 2010] p.24:

“An academic discipline that encompasses the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society.”

Additionally, computer science education will be abbreviated by CSE. At the end of this introduction I have to apologize to all English speaking readers for the frequent citation of German publications. As though this might be regarded as natural for a German project to a certain extent, I have tried to find as many English substitutes as possible. At the end I have constrained the German sources to cases that were absolutely inevitable.

2. THEORETICAL BACKGROUND

Although there is a vast amount of theoretical work that is relevant for this project, I will restrict the discussion at this point to the theories that decide the structure of the paper or that are very important for several sections. The theoretical background that is relevant to single sections only will be presented there in order to keep the logical structure of the paper as clearly arranged as possible. Additionally, the report of WG ISE [Hubwieser et al. 2011] provides many references to theoretical background that might be interesting for this topic.

2.1 The Darmstadt Model

Starting from the so-called Berlin Model [Heimann et al. 1965] (in English see [Uljens 1997]), the WG ISE performed a qualitative text analysis of five selected case studies about CS in Secondary education. It turned out that this model was not sufficient for the description in some aspects. The central issue was that the top level categorization of the Berlin model (Preconditions, Decision Areas and Consequences) is strongly dependant on the influence range of the relevant persons (e.g. the authors). Therefore, the WG ISE extracted the two variables Berlin Model Top Category and Level of Responsibility/Range of Influence into a new dimension each. Additionally, some relevant and important categories were missing (e.g. curriculum issues) in the Berlin Model and were consequently added to the new system. The result of the WG ISE was a new three-dimensional category system, which was named Darmstadt Model in honor of the location of the ITiCSE conference 2011:

- Dimension 1 (Level of Responsibility/Range of Influence) determines the decision level of the regarded stakeholders. According to the position of the respective person in the school-system, the following subcategories are suggested: Student/Pupil, Classroom, School, Region, State, Country, and International.

- Dimension 2 (Berlin Model Top Category) is formed by the categories of the first level of the original Berlin Model: Preconditions, Decision Areas and Consequences.
Dimension 3 (*Educational Relevant Areas*) describes issues that are directly relevant for educational activities. It comprises the remaining subcategories of the original Berlin Model that have turned out to be relevant in our context (e.g. *Intentions*) and additionally several other categories that had emerged during coding (e.g. *Educational System*). In table 8 in the appendix all categories of this dimension are listed and additionally categorized according to dimension 1 and 2 in the context of our case. Additionally, the last column refers to the section of this paper where the respective category is discussed.

Regarding the theoretical background of those categories, I have to refer to the report of the WG ISE [Hubwieser et al. 2011].

At this point I have to remark that the Darmstadt Model is in a very early development stage. The denominators of the three dimensions might be improved in order to be more precise. Also the categories are not well-defined yet, as the quite poor intercoder agreements of the coding iterations showed [Hubwieser et al. 2011]. Nevertheless, we are convinced that the model could be very beneficial in the future. As already stated in the introduction, this report might be regarded as an illustration and first application test of the Darmstadt Model.

### 2.2 Object Orientation

The object-oriented paradigm was defined e.g. in 1990 by Rosson and Alpert as a combination of the four aspects *communicating objects, abstraction, shared behavior* and *designing with objects* [Rosson 1990]. In order to define the concept of *object-orientation*, James Rumbaugh stated in 1991, that object orientation would have to encompass four aspects: *identity of objects, classification, polymorphism* and *inheritance* [Rumbaugh and Blaha 1991]. In 1992 Henderson-Sellers identified the following three main conceptual components: *encapsulation – information hiding, abstraction – class·object, inheritance – polymorphism* [Henderson-Sellers 1992].

Armstrong conducted a very interesting review of 239 publications and presented a list of the 20 aspects of object-orientation that were addressed most frequently in the definitions of object orientation. The five most frequently used were *inheritance, object, class, encapsulation* and *method* [Armstrong 2006].

Discussing the *Objects-first* paradigm, Lewis stated:

“A distinction must quickly be made between initially *writing classes that define objects, and using objects* defined by preexisting classes. It is sometimes suggested that if students do not write multiple classes and methods initially, they are not being indoctrinated into an object-oriented approach. Most educators agree, however, that using objects is the appropriate first step.” [Lewis 2000]

In 2008, Bennedsen and Schulte conducted a very interesting survey about the understanding and implications of *objects-first* among introductory programming teachers. They contacted about 700 authors, teachers and SIGCSE members and received 298 at least partly filled out questionnaires. The content analysis led to the suggestion of the three categories *using objects, creating classes and concepts* (“involves the teaching of the general principles and ideas of the object-oriented paradigm, focusing not just on programming but on creating object-oriented models in general”). The authors deduced three common sequences of *objects-first* courses: “

1. Using objects, followed by
   a. creating classes, followed by concepts.
   b. concepts, followed by creating classes.

2. Creating classes, followed by concepts.” [Bennedsen and Schulte 2008]
In her doctoral thesis Ira Diethelm defined the strategy strictly models and objects-first that should start with the usage and manipulation of objects before the class concept is introduced [Diethelm 2007b], corresponding to the sequences 1.a and 1.b according [Bennedsen and Schulte 2008].

3. RELATED WORK

Similar to the presentation of the theoretical background above, I will constrain the references to related work at this point to publications that could be regarded as case studies of similar projects. Related work that concerns only single aspects of the project will be referred in the related chapters below.

There are several publications describing national initiatives, e.g. by A. Tucker, who presented and explained the situation in the US and the ACM K12 [Tucker 2010] or the UK [Glendinning and Low 2010]. Frequently these initiatives come from eastern European states, e.g. as described by V. Dagiene in her papers about CS education in Lithuania [Dagiene 2005], [Dagiene 2008]. In the last years several more or less detailed discussions of computer science education in secondary schools have been written, e.g. from Austria [Micheuz 2005], [Micheuz 2011], Poland [Gurbiel et al. 2005], [Syslo and Kwiatkowska 2008], [Syslo 2011], New Zealand [Bell et al. 2010], Israel [Armoni et al. 2010] or Italy [Calzarossa et al. 2011].

Regarding the US, the Computer Science Teachers Association of the ACM (CSTA) had described the situation in 2005 [Stephenson et al. 2005]. This was followed by the already mentioned report of [Wilson et al. 2010], which investigated the implementation of the ACM Curriculum in the US-states.

4. THE BAVARIAN PROJECT IN A NUTSHELL

In the following sections I will present the most important facets of the project, aiming to give an overview before all relevant aspects are discussed in detail according to the structure and categories of the Darmstadt Model (see section 5).

4.1 The Project

The work on the project started in 1994 at the Technische Universität München (TUM). At this time CSE in German secondary schools, implemented mainly as elective courses, was declining from the first hype that had started at the end of the 70ies. According to the teaching approaches that have been developed until this time there was a choice between courses that focused on the application of standard (particularly office) software on the one hand or on programming (usually in Pascal at that time) on the other hand. While the first choice did not satisfy the requirements of general education that is demanded of a regular subject in German secondary schools, the second choice was too technically oriented and therefore the students were disappointed very often at the end by the missing of application context for their programs. As a consequence, the enrollment for the courses decreased dramatically in these years.

Therefore, we started to developed a new teaching approach that emphasized the application of object-oriented modeling techniques that had been developed shortly before e.g. by [Rumbaugh and Blaha 1991] or [Booch 1994]. The first results of this development were published in 1997 [Hubwieser et al. 1997].

The approach was illustrated by a set of exemplary lessons (modules), that described the learning objectives, intended knowledge elements, teaching methods and media in detail [Hubwieser 2000], see table 1.

After some experimental projects that had started in 1997, in the year 2000 the government of Bavaria decided to introduce a new compulsory subject of Informatics (synonymously CS from now on) in all Gymnasiums, which represent the most demanding of its three types of secondary schools in Germany (see section 5.1), similar
to grammar schools in the UK or academic high schools in the US. We will use the term grammar schools from now on.

<table>
<thead>
<tr>
<th>M1</th>
<th>Novice course (Fundamentum):</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1a</td>
<td>Object-oriented modeling of documents</td>
</tr>
<tr>
<td>M1b</td>
<td>Introduction to algorithms</td>
</tr>
<tr>
<td>M2</td>
<td>Graphical representation of information</td>
</tr>
<tr>
<td>M3</td>
<td>Data modeling and data base systems</td>
</tr>
<tr>
<td>M4</td>
<td>State modeling</td>
</tr>
<tr>
<td>M5</td>
<td>Functional modeling, data encryption</td>
</tr>
<tr>
<td>M6</td>
<td>Object-oriented modeling and programming</td>
</tr>
<tr>
<td>M7</td>
<td>Recursive data structures</td>
</tr>
<tr>
<td>M8</td>
<td>Formal languages</td>
</tr>
<tr>
<td>M9</td>
<td>Computer networks</td>
</tr>
<tr>
<td>M10</td>
<td>Limitations of computability</td>
</tr>
</tbody>
</table>

Table 1. The modules of the original project

This was extended later [Hubwieser 2007b] by two additional modules (see table 1a).

<table>
<thead>
<tr>
<th>M5a</th>
<th>Functional modeling and spreadsheets</th>
</tr>
</thead>
<tbody>
<tr>
<td>M11</td>
<td>Integrative project: InfoBanc</td>
</tr>
</tbody>
</table>

Table 1a. Additional Modules

The subject started in 2004 in grade 6. Consequently, the first age cohort of students that had attended our course graduated from grammar schools in 2011. At this time about 750.000 students had attended “our course” (more precisely the course that was created according to our concept by the Bavarian school administration) in any grade. A comparison of table 9 in the appendix and the original list of modules (see tables 1, 1a) shows that our original concept was realized quite closely at the end.

4.2 The Course of Lessons

Although all relevant aspects of the teaching concept (e.g. intentions, content, methods) will be discussed systematically and in detail below according to the categories of the Darmstadt Model, I will give a short overview here, so that the reader might get an impression about what the students are learning in the course at this point already. All details concerning e.g. learning objectives, teaching methods or software systems are explained in section 5.

As described in [Hubwieser 2003], the compulsory courses in grade 6 and 7 start with the application of the concepts object, attribute, method and class in the context of vector graphics, worked out by the students using e.g. OpenOfficeDraw©, MS Powerpoint© or CorelDraw©, see figure 1.

Figure 1. Object Models of Vector Graphics
In a second step, the concept of aggregation (in the sense of e.g. “paragraphs containing characters” at this point) is introduced in the context of word processing (figure 2).

![Image of object models of text documents]

**Figure 2. Object Models of Text Documents**

Following this, the students work with recursive aggregation, applied to file systems, where objects of the class *Folder* may contain other objects of this class (figure 3). This leads to the construction of folder trees as a representation of hierarchical structures. In the next step, the trees are generalized to graphs by the application of references in the context of hypertext systems.

![Image of hierarchical information represented by folder trees]

**Figure 3. Hierarchical Information represented by folder trees**

At the end of grade 7, the students work out their first programs, using a virtual robot, e.g. the *Robot Karol* (www.schule.bayern.de/karol), which is regarded as the only instance of a class *Robot*, with some “built-in methods” like *forward*, *turnLeft*
etc. “Programming” is understood as creating new methods for the class Robot at this point.

In the 9th grade, where CS is compulsory in one of the four education tracks of grammar schools in Bavaria (science & technology track), the students learn to apply the concept of function. They construct functional models (data flow diagrams), where the information processing units are restricted to functions. The models are implemented on spreadsheet systems like MS-Excel© or OpenCalc© (see figure 4), which has been described in detail in [Hubwieser 2004].

![Figure 4. Functional model implemented in a spreadsheet system](image)

The greater part of the school year in grade 9 is dedicated to (object-oriented) data modeling. The students work with the basic concepts of data base systems: record, table, query, using a relational data base system like MS Access© or MySQL©. They design entity relationship models of more complex systems that consist of several tables, connected by relationships, and implement their models using relational data base systems.

The students of grade 10 consolidate their object-oriented knowledge by “real” object-oriented programming, designing object-, class- and state-models and implementing them with a suitable object-oriented programming language, which currently will be Java in most of the schools. Additionally, they learn to apply the concepts of subclass, superclass, inheritance and polymorphism.

In the elective courses in grade 11, the object-oriented modeling and programming concepts are extended by the recursive data structures List, Tree and Graph. Afterwards the students are introduced to basic concepts of Software Engineering, which they apply within the context of a large programming project: collaborative workflows, milestones, phases of software development, coordination of concurrent processes, combination of several different modeling views, software patterns, test, verification of the completeness and correctness of the system and documentation of the project.

In the second year of the elective course, the students are introduced in several important subject areas of computer science in grade 12: Formal Languages, Parallel Programming, Assembler Programming and Limitations of Computability.

4.3 Publications

Up to now, our curriculum proposal was described in several publications. In [Hubwieser et al. 1997] we described the principle that the curriculum should be dominated by certain modeling techniques. Following this, we presented in [Müller and Hubwieser 2000] the first outline of the curriculum for the three relevant age levels (11-13, 14-16, 17-19 years). Together with Norbert Breier, I explained the Information-Oriented Approach [Breier and Hubwieser 2002] our concept is based on. In
I have described the object-oriented concept for the first age level in detail. This was followed by Hubwieser 2004, where I have outlined the first part of grade 9 (functional modeling). An overview of the curriculum for the first two age levels up to grade 10 was presented in Hubwieser 2006. In Hubwieser 2007a and Hubwieser 2008 I have discussed specific issues concerning the part of object-oriented modeling and programming that fills grade 10 and the first months of grade 11. First results of the project were presented in Mühling et al. 2010 and Steer and Hubwieser 2010.

5. APPLICATION OF THE DARMSTADT MODEL
Following the short overview in section 4, we present the details of our new compulsory subject according to the structure of the Darmstadt Model (see section 2.1), which is constituted of three dimensions:

- dimension 1: Level of Responsibility/Range of Influence,
- dimension 2: Berlin Model Top Category,
- dimension 3: Educational Relevant Areas.

In consequence, the presentation should be three-dimensional also, which is hardly representable in continuous text. Therefore, I decided to structure the presentation of our project according to the 3rd dimension (Educational Relevant Areas) of the model, because this contains the most relevant categories regarding this case study. Consequently, the following subsections will be labeled (more or less) with the top-level categories of this 3rd dimension. The first exception is Research, because this, on the one hand, is an issue all over this paper and, on the other hand, is particularly important for the results in section 6. It is apparent that some of the categories should be changed or renamed in order to fit better for this case study. Nevertheless, I will keep the original terminology of the Darmstadt Model. However, according to logical interdependencies between the categories, I will change the order of some categories as well as the level in the hierarchy of others. Additionally, I will combine some of them to a common chapter. Although this might seem quite arbitrary at a first glance, it suggests itself according to the interdependences in the context of our case study. Table 8 in the appendix keeps track of the original categories of the Darmstadt Model and how the sections of this paper cover each of them.

The respective values of the 1st and the 2nd dimension of the Darmstadt Model will be discussed at the beginning or at the end of a few of the following subsections as far as they are particularly relevant there. Yet, seeking not to bore the reader, pure repetitions of statements about the same values of dimension 1 (mostly state-wide) and 2 were omitted in most subsections. All values of all dimensions can be looked up in table 8 in the appendix. Extending the original scale of dimension 1, I have added the value “none” to the 1st dimension, meaning that we had no influence at all on the respecting category. Please note that this has to be decided from the case that a stakeholder has an influence that does not reach far enough, e.g. a statewide influence on the national education system.

5.1 Educational system
As the work on this project was performed at a university, there was no direct influence on the school system at the first glance. Nevertheless, due to the close cooperation with the department of school of Bavaria, the level of responsibility, representing the value of the first dimension of the Darmstadt Model, was statewide in most cases. Concerning the subcategory Enrollment (of Educational System according dimension 3 of the Darmstadt Model), this issue is a precondition in certain respect, e.g. under-
stood as the enrollment of students in the Science & Technology direction of study, where CS is a compulsory subject in grade 9 and 10. On the other hand, the quality of our CS course (and hereby our work on it) might strongly influence the enrollment of students as CS majors at the university. Thus, enrollment should be regarded as a consequence also. Some results regarding both aspects are presented in section 6.

Similar to the US, Germany is a federal republic, formed by 16 partially independent states. Bavaria is largest of these states regarding area, while Nordrhein-Westfalen (NRW) is the largest regarding population. While NRW has 17.9 million, Bavaria has 12.5 million inhabitants. Bavaria is famous for its beautiful landscapes (e.g. the Bavarian Alps) on the one hand, but also for its successful high-tech industry with some very innovative companies (e.g. Siemens, Audi, BMW) on the other hand.

Each of the 16 German states runs its own quite different education system. The Bavarian system is tripartite (see figure 5) in the lower secondary stage, consisting of three school types (English terms according to [German Federal Office for Migration and Refugees])

- secondary general school (Hauptschule),
- intermediate secondary school (Realschule),
- grammar school (Gymnasium).

Bavaria offers approximately 5,500 Schools that are attended roughly by 1,880,000 students. Currently there are 405 grammar schools that comprise the grades 5–12 and qualify for enrollment at a university directly, 352 intermediate secondary schools (grade 5-10) and 1,147 secondary general schools (grade 5-9).

![Figure 5. The Bavarian school system (FOS = Fachoberschule).](image)

All students in grade 1-4 attend one of the 2,418 elementary schools. Depending on their performance in grade 4, some of them are qualified for grammar school, some others only for intermediate general school. The rest have to attend a secondary general school. In 2007 for instance, 37.1% of the students that had graduated from primary school switched to grammar school, 22.3% to intermediate general school and 39.4% to a secondary general school [ISB 2009]. In summary, about one third of every age group attends a grammar school, more or less selected as the most gifted
students. The average number of students per class in grammar school was 27.6 in 2007/08.

Interestingly, girls are overrepresented at grammar school: While 49.2% of the students at primary schools are female, there are 52.0% girls at the grammar school, but only 45.5% at the secondary general school. This illustrates the fact that in Germany girls are generally more successful in school, as all indicators show. Regarding immigration background, we find opposite trends: in elementary school (Grundschule), 12.0% of the students have immigration background (they or their parents have immigrated), most of them fail qualify for grammar school, where only 6.1% have immigration background compared to 20.6% at the secondary general school.

The Bavarian schools are not free to decide individually neither about the subjects they teach nor about the number of lessons that are given in each subject per week. These specifications are prescribed by law in a certain catalog of subjects (called Stundentafel). In consequence, this catalog has to be changed in order to integrate a new subject. The catalog of subjects for the grammar school does not offer many choices to the students up to grade 10, apart from four different education tracks: science & technology, foreign languages, economy, music & arts, each with its own catalog of subjects. Most of the grammar schools offer 2-3 directions of these.

In grades 11 and 12 of grammar school, the students are offered a certain, limited choice of courses. They have to attend compulsory courses in Religious education or ethics, German language, mathematics, history & social studies and sport. Additionally, they have to attend courses in at least one continued foreign language and one natural science (physics, biology, chemistry). They have to choose either a second foreign language, a second natural science or CS, either geography or economy & laws, music or art. Additionally, they have to attend two specific seminars: one project seminar that prepares them for a course of study at university and a shows possible careers in industry, the second seminar is intended to introduce them into basic principles of scientific work.

The quality management of schools is delegated to the Bavarian State Institute for School Quality and Educational Research (ISB), which trains, coordinates and controls the external school evaluation teams that are continuously evaluating all schools. Additionally, the ISB is monitoring the learning progress of the students by voluntary age group tests and comparative studies that are performed in grade 2, 3 and 8 (see section 6).

5.2 Examination/Certification

The Bavarian school system offers four different graduation levels. The lowest level (Qualifizierter Hauptschulabschluss) can be achieved by graduating from a secondary general school. The second level (Mittlerer Bildungsabschluss) can be acquired by graduating from a intermediate general school or from the M-track of secondary general school. The third level (Fachabitur) is delivered by grade 12 of a Fachoberschule (FOS). It qualifies for enrollment at a University of Applied Science. The highest level is represented by the graduation from grammar school or FOS 13, which qualifies for enrollment at a university.

The graduation from grammar school requires the passing of a centralized examination (Abitur) and entitles the student to enroll at any university. From the students that have graduated e.g. in 2003 from grammar school, 82.5% have enrolled at a university until 2009.

The Abitur comprises three written and two oral exams. The written ones are centralized over Bavaria and mandatory in German language and mathematics, while the third subject is elective with certain restrictions, similar to the two subjects for the oral exams.
As our concept strongly influenced the nation-wide standards for the graduation from high school in CS [Sekretariat der ständigen Konferenz der Kultusminister der Länder in der Bundesrepublik Deutschland 2008] we had we had a nation-wide influence on this category in some respect.

5.3 History of CS in Bavarian Schools

In any case it seems prudent to have a close look at relevant historical experiences before you change a school system, in our case at the history of CS and ICT at school. Our presentation follows [Forneck 1990], who has presented an excellent discussion of those “early years” of CSE in Germany.

In Germany, during the 60ies the schools started to teach CS. In admiration of the impressive achievements of the first mainframe computers (e.g. the calculation of the space trip to the moon) the first approach was focused on hardware, proposing to teach assembler programming, Boolean algebra and formal languages.

Induced by the evolution of CS into a serious scientific discipline, a second approach was developed. It stressed the systematic development of algorithms, which seemed to promise valuable educational effects.

Following this approach, the general educational reform of the 70s [Robinson 1967] suggested to postulate that each programming task should have an application context. Unfortunately, this demand turned out to be hardly realizable, mainly because it led to a significant increase in complexity of the algorithms.

Inspired by the increasing propagation and the dramatic drop in prices of software applications at the beginning of the 80s it was suggested to teach the usage of standard software instead of programming skills. This fourth approach produced the concept of Basic Education in Information Technology that was introduced as an integrated approach (mostly within the subjects mathematics, economy and German language) all over Germany during the 80ies. Unfortunately, the students did not get any deeper understanding of basic concepts of hard- and software systems thereby, which would be required for a competent usage as well as for the assessment of the chances or risks of these technologies [Liessel 1994].

Caused by the development of the Internet, the information-centered approach followed, which claimed that CS as the science of information should be taught as well as Physics as a the science of energy and Chemistry as the science of matter. Following this approach, the students should learn mainly how to deal with extensive or complex information [Breier and Hubwieser 2002].

At the time we presented our first concept, there were several types of CS courses at the Bavarian grammar schools. In grade 7-11 most Gymnasia offered elective courses that were additional to the compulsory program. Therefore, the teachers were free to design the courses. The majority decided to teach programming, some trained user skills. Another type of CS course was integrated in Mathematics as an elective element, concurring to several special fields of Mathematics, e.g. complex numbers or descriptive geometry. This course was designed according the algorithmic approach, as well as the elective course in grade 12 and 13 which could be attended additionally.

In our concept we have tried to integrate some intentions, knowledge elements and methods of all these approaches, as far as they have turned out to be suitable: we took the very fundamental principles of the mostly spread hardware concepts (von-Neumann-principle) and the formal languages from the hardware-oriented approach, the principle of systematic algorithm development and discussion from the algorithmic oriented approach, the principle of teaching in context [Cooper and Cunningham 2010] from the application oriented approach, the integration of standard software (but aiming at understanding their basic principles) from the application oriented
approach and finally the goal of managing complex information from the information oriented approach.

5.4 Policies
Two state-wide policies had a substantial influence on our project. The first one was an extensive structural reform project that should improve the grammar school in its original, nine-year spanning form (G9/2). The work on this project had started in the late 90ies, while the concept was scheduled to go into operation in 2003. As there were many, partly concurring proposals for this new form, the Bavarian administration launched several different experimental projects, amongst others the Mathematical-Scientific Gymnasium Type 2 (MNG2) and the European Gymnasium Type 3 (EGy3), that were scheduled to run from 1997 to 2003. About 20-30 schools participated in each of these. Following our proposal, it was decided to incorporate a compulsory subject of CS in both of these experimental projects. As a result of these experiments, we received very interesting feedback from students, teachers and parents. Based on these experiences, the government decided in 2000 to introduce a new subject of CS in the new “regular” G9/2 that should comprise compulsory courses in grade 6, 9, 10, 11 and elective courses on two different levels (basic/advanced) in grade 12 and 13.

The second, even more crucial policy was the reduction of the grammar school from 9 to 8 grades that was decided very unexpectedly in autumn 2003 by the Bavarian government, only two months after the new G9/2 had been put into operation in grade 5. The planning and preparation work on this new type of grammar school (G8) had to be finished already in autumn 2004, when the G8 should start in grade 5 and 6 simultaneously. The new subject of CS was incorporated in its current form that is displayed in figure 6.

As a consequence of the shortening of the grammar school by the introduction of the G8, two age cohorts would graduate nearly simultaneously from the grammar school: the last age group of G9 in April 2011 and the first one from G8 in June 2011. Therefore, (at least) the Bavarian universities would face the big challenge to master approximately the double number of freshmen as usual in autumn 2011.

5.5 Organizational aspects of subject - Degree of compulsion
As the stakeholders of any existing subject would never give up teaching time in favor of another subject voluntarily, the integration of our new subject was only possible within a total rearrangement of the catalog of subjects. By the occasion of the extensive structural reform project (see section 5.4) of the Bavarian grammar school the catalog of subjects had to be changed extensively in 2003 (G9/2), which offered the chance to incorporate a new subject. The catalog of subjects was changed again in the in the context of the reduction of the grammar school in 2004 (G8, see section 5.4).

Figure 6 shows the resulting structure of the new subject and the years it has started in the different grades. In grade 6 and 7, CS is incorporated formally in the combination nature & technology (NuT) that comprises the subjects biology (in grade 5 and 6), physics (in grade 7) and CS (in grade 6 and 7). Nevertheless, all three subjects are taught separately in a prescribed number of lessons per week by teachers with a university degree in the respective subject.
In the science & technology track, the students have to attend CS as a compulsory subject in grade 9 and 10. In average about 50% of the students choose this direction. In grade 11 and 12, an elective CS course might be chosen instead of a second natural science or a second foreign language. In the final examination, CS can be chosen for written as well as for oral examination.

In 2008, a compulsory subject information technology was introduced at the Bavarian intermediate general schools. It was derived in parts from our concept for the grammar school and has a modular curriculum that prescribes different sets of modules for the different directions of study at intermediate general school. However, this paper is focused on our original concept for the grammar school.

5.6 Teacher Qualification

In general teacher education in Bavaria is located at the universities. The teacher students are educated and examined for one specific school type only. For all types of secondary schools, the teacher students have to pass two centralized examinations. The first one qualifies for the subjects to be taught, the second one for Pedagogy and Psychology. There are certain restrictions for the number, extent and combination of the subjects that depend from the school type. Concerning Gymasium, the students have to choose two subjects that are implemented on an equal level in their course of study. computer science can be combined with Mathematics, Physics, English or Economics & Law.

The introduction of every new subject has to overcome a typical initial problem: without a respective subject there is no need for teacher education programs, while without properly educated teachers it is not possible to start a new subject. The need of CS teachers caused by our new subject was estimated to add up to approximately 1200 persons (at least 3 for each grammar school). We solved the initial problem described above by a circular strategy. Long before the subject was decided, we had launched in 1995 a 2-year lasting in-service training program at two universities that was successfully attended by 94 teachers. Simultaneously, we installed a regular course of study for CS teachers that started 1997. This was followed by the large SIGNAL program (2001-2006), that was initiated and coordinated by the author and that educated about 300 teachers at 5 universities all over Bavaria. The subsequent program (FLIEG), which is focused on self studies, was attended by about 80 teachers until now. Additionally, the Bavarian administration started programs to attract people with Diploma or Master degree in CS to switch to teachers profession. As our teacher survey in 2009 showed (see section 6), about 50% of the responding “active” CS teachers had already a degree in CS at that time. Taking into account the number of students that attend the subject currently (see table 5) and the number of gradua-
tions from our teacher education courses, we estimate that there are about 1300 persons teaching CS currently, among which about 700 should have a degree in CS.

In 2007 the German Kultusministerkonferenz (the national board of the 16 secretary of state that are in charge for the schools) started to work out Standards for Teacher Education. The author was invited to act as one of the two members of the commission for CS. The standards were decided and published in 2008 [Sekretariat der ständigen Konferenz der Kultusminister der Länder in der Bundesrepublik Deutschland 2008]. In this respect we had a nationwide influence regarding educational standards.

5.7 Techno-economic Development

Concerning the technical circumstances, the last JIM-survey [Medienpädagogischer Forschungsverbund Südwest 2009] in the German state of Baden-Württemberg showed that 100% of the households of 12-19 year old students own a computer or a laptop and 98% have Internet access. Since Baden-Württemberg is structurally very similar to Bavaria, we can assume that in Bavaria the same proportions will hold. As the grammar schools attract predominantly the children of parents from the upper social ranks (as all PISA studies show), we can assume that all its students have a computer and Internet access at home.

In 2009 about 90% of the boys and girls used Internet more than once a week, the average weekly time of Internet usage was about 134 min. [Medienpädagogischer Forschungsverbund Südwest 2009], 47% of this time was used for communication, 18% for games, 14% for searching information and 23% for entertainment (e.g. music). The only real big difference in regard to these percentages between boys and girls was gaming: boys spend 24% of their Internet time for games, girls only 8%. We can draw the conclusion that our students are very experienced users of computers and Internet, and additionally that games as an application area for CS has a very different attraction for boys and girls.

5.8 Personal factors of the students: Age, Gender, Social and Immigration Background

As our subjects starts in grade 6, where the majority of the students are 11–12 years old, we have to take note that some of them might not have reached the stage of Formal Operations according to [Piaget 1963], remaining still in the stage of Concrete Operations and therefore requiring concrete objects for rational deliberations, being not yet capable of hypothetical and deductive reasoning. Recently Spiel and Glück performed the Competence Profile Test Deductive Reasoning with 418 students that were 7-12 years old. The result was that none of them had reached the stage of Formal Operations totally [Spiel and Glück 2008]. Thus we have to take into account that the majority of the students will not have reached the last Piagetian stage in grade 6 and 7. As a consequence, CSE has to stay close to concrete examples and low-abstraction levels in grade 6 and 7.

Concerning the gender aspect, we have a slight majority of young women at the Bavarian grammar schools (51-52%), which are more successful additionally (2.5% of the boys, but only 1.5% of the girls have to stay down a year). Unfortunately, the girls tend to avoid the science & technology direction of study: it was chosen by 65.8% of the boys, but only 34.4. % of the girls in 2009 [ISB 2009]. Consequently, in grade 6 and 7 we have slightly more girls than boys, while in grade 9 and up there are be substantially more boys in the CS courses.

Concerning the immigration background, which might be important e.g. for selecting a teaching context that is motivating equally for all students, we have to state that students with immigration background in the 1st or 2nd generation are underrepresented at the grammar school: 32.7 % of an age group without such a back-
ground attends a grammar school, while only 17.9% (1st generation) respectively 15.3% (2nd generation) with such a background does so.

5.9 Family Socialization, Public opinion

Following the international Norton Online Living Report '09 [Symantec Corporation 2009], 55% of German parents insist on a weekly media-free day at home, during which their children do not watch TV, go online or use the computer. This is the highest rate of all countries surveyed, compared with a rate of 20% globally. According to the Norton Report, Germans are by far the most risk-averse people compared to other nations, with just over one in four (26%) reporting the risks of using the Internet outweigh the benefits. In average, 5% of adults in the countries surveyed dislike learning about new websites or Internet communication tools, with the highest levels of dislike being reported from Germany (12%) and Japan (12%). One in five parents in Germany have caught their child doing something online they don't approve of, but only 9% have reprimanded their child for such behavior. Yet, parental confidence in Germany is extremely high - 81% are confident that they would know what their child is looking at online. The children report a different story, as one-third of the kids reports that their parents would not know what they are doing online. This tells us that German parents are relatively critical concerning the new information and communication technologies, but in large part don't really control the Internet activities of their children.

CS education has to respond to this public opinion by incorporating proper media education and by covering the technical and logical foundations of digital media, aiming to help the students to apply the media in consciousness of dangers and drawbacks.

5.10 Curriculum Issues

In Bavaria, the structure and the elements of the curricula for all subjects are decided by the government. For this purpose, a specific state institute, the Staatsinstitut für Schulqualität und Bildungsforschung (ISB) was installed, which is in charge for curriculum development and quality assurance of the schools. Thus there had to be a formal curriculum design process before our subject could start. The total curriculum for the G8 consists of three levels: at the top level the goals of the school type and the overall structure of the educational process is defined. Level 2 describes the goals and characteristics of each subject and level 3 the knowledge that is taught in each subject and grade. Learning objectives were to be specified only as “basic knowledge” per grade, which resulted in about 10 objectives per subject and grade.

The work on the curricula started in 1997 with the design of the curricula for the two experimental projects MNG2 and EGy3 (see section 5.4) and was finished in 2008 with the curriculum for the regular grade 12. Looking for knowledge elements that we should incorporate, we inspected several existing curricula very accurately. In 1994 the UNESCO offered its curriculum for CS in secondary schools [van Weert and Tinsley 1994], which was renamed to “Information and Communication Technology in secondary Education” in 2000 [van Weert 2000] and extended to “Information and Communication Technology in Education” in 2002 [van Weert and Anderson 2002]. In 1993, a special task force of the ACM presented its Model High School Computer Science Curriculum [Pre-College Task Force Comm. of the Educ. Board of the ACM 1993]. A decade later, the ACM K-12 Task Force Curriculum Committee published its “Model Curriculum for K-12 Computer Science” [Tucker et al. 2006], which defined also a set of standards (see section 5.15). This proposal was revised very recently [Tucker et al. 2011]. Table 9 in the appendix displays the result of our curriculum design process.
5.11 Intentions: Learning Objectives and Competencies

Concerning the intentions of our project, we have to distinguish between the intentions of the whole project and the intentions about the learning outcomes of the students. The former were quite simple: we wanted to install a successful new subject of computer science. This intention has always started at the students’ point of view: we aim to help them to make their living (and their money) in a society that is dominated by modern information and communication technology.

Concerning the learning outcomes of the students, things are more complicated. According to [Anderson and Krathwohl 2001], we regard learning objectives as a combination of a certain type of knowledge (factual, conceptual, procedural or meta-cognitive knowledge) and a certain intended cognitive process (remember, understand, apply, analyze, evaluate or create). Anderson & Krathwohl also proposed a categorization of learning objectives according to their granularity:

- **global** objectives: “Complex, multifaced learning outcomes that require substantial time and instruction to accomplish”,
- **educational** objectives: derived from global objectives by breaking “them down into more focused, delimited form”, and
- **instructional** objectives, with the purpose “to focus teaching and testing on narrow, day-to-day slices of learning in fairly specific content areas”.

Regarding the global objectives, we have to look at the legal prescriptions of our state. The German constitution determines that the federal states are responsible for their own educational system. The constitution of Bavaria defines the overall goals of the schools in Art. 131 (translated by the author):

“[..] The most important educational goals are awe for God, worship of religious conviction and for the dignity of the human being, self-control, sense of responsibility, readiness to help and open-mindedness for everything that is true, good or beautiful and sense of responsibility for nature and environment.”

In Art. 2 of the Bavarian law for schools (BayEUG) some goals are set that are particularly relevant for CSE (translated by the author):

“The schools have the particular mission to enable students for independent judgment and independent and autonomous acting in society, to educate them towards the responsible use of freedom, to promote equal rights of women and men and to work towards the removal of existing disadvantages, to enable the students for the equal perception of their rights and duties in family, state and society, to prepare for professional life and profession, to support the choice of career and to encourage in particular girls and women to extend their occupational spectrum.”

Art. 9 defines the educational mission of the grammar schools as a combination of extended general education, which is demanded by the universities with the general preparation for a vocational education instead of university.

In combination with the intentions at level 1 (the project) and the circumstances of life in the current information society, we derived the following global objectives [Hubwieser 2007a]. The students should:

- acquire the capability of independent opinion and responsible acting in the Information Society,
- be able to act responsibly and efficiently in a world of work and profession that is ubiquitously penetrated by IT,
– master efficiently the tools and understand the limitations, chances and risks of information technology,
– learn the responsible, efficient usage of information technology based on knowledge of the theoretical foundations and basic principles of the systems,
– master complex systems, particularly being able to describe their structure and behavior and communicate about them in a competent way,
– be prepared for the application of information technology in other school subjects, and
– be able to chose their career based on a sufficient knowledge of the possibilities and principal limitations of future IT developments.

Concerning educational objectives, we have explained in [Hubwieser 2007a] that our CS lessons are very demanding, e.g. the students have to

– analyze problems in order to represent them properly by an object-oriented model,
– evaluate alternative models in order to choose one of them, and
– create models and programs out of their models.

This shows that the educational objectives reach the most difficult cognitive process dimension create according to Anderson-Krathwohl [Anderson and Krathwohl 2001].

Finally, discussing the instructional objectives, we have shown that a quite simple object-oriented program with a suitable context (e.g. simulating a traffic light), easily demands 40 or even more instructional objectives in order to be understood by the students [Hubwieser 2007a].

During the work on our project, particularly during the first decade of the new century, it has become much more popular to represent the intentions of educational projects by competencies, although this might be contradictory to the original character of the latter, being learning outcomes more than objectives. According to Weinert, a competence in human and social sciences is

"...a roughly specialized system of abilities, proficiencies, or skills that are necessary to reach a specific goal. This can be applied to individual dispositions or to the distribution of such dispositions within a social group or an institution" [Weinert 2001].

Following the definition of the OECD,

“a competence is the ability to meet a complex demand successfully or carry out a complex activity or task” [OECD 2003].

As we have already discussed in [Hubwieser et al. 2011], it is hard to distinguish if a certain description of an ability denotes a competency or a learning objective in many cases. This seems to be a question much more of the point of view than of the described ability. While competencies describe outcomes of the learning process that are to be applied in complex, real world situations, learning objectives represent the intentions of the teachers or planners of the project. Nevertheless, the final goal that students acquire a certain competency might be an intention before the learning process starts.

In order to demonstrate the contribution of our CS course to general education, we refer to the *The Definition and Selection of Key Competencies* (DeSeCo), a framework that was published by [OECD 2005]. The educational level of courses will be assessed in Europe according to the *European qualification framework* (EQF), presented by the European Union in 2008 [European Parliament 2008]. It distinguishes 8 levels of competence, starting from level 1: “work or study under direct supervision in a struc-
asured context” up to level 8: “autonomy, scholarly and professional integrity and sustained commitment to the development of new ideas or processes at the forefront of work or study contexts including research”. The level of competencies in our central substantial field of object-oriented programming can be described by the empirically founded Competence Model for Object-Interaction in Introductory Programming, presented recently by [Bennedsen and Schulte 2006]:

- Interaction with objects,
- Interaction on object structures,
- Interaction on dynamic object structures, and
- Interaction on dynamic polymorphic object structures.

Table 10 in the appendix shows the intended coverage of the competencies of these three models by our course. Of course it has still to be proven by empirical surveys that the students really reach these competencies.

5.12 Knowledge

In contrary to former approaches of teaching, the learning “content” is not supposed to be the most important aspect of teaching any more. Instead, the focus has shifted towards the competencies the students should acquire, as discussed already in section 5.11. In fact, the term “content” is quite treacherous, because it references to a learning metaphor that has become obsolete already many years ago. Thus, in my opinion it is more adequate to use the term knowledge instead. Nevertheless, the intended knowledge is a very crucial aspect still, even if it is gains its importance mainly as a component of learning objectives or, following [Weinert 2001], as one of several facets of competence, see section 5.11. After all, the relevant subject matter knowledge determines the substantial and logical structure of the teaching process.

As it is impossible to teach all the knowledge of CS in secondary schools, we had to select the knowledge elements that seem to be particularly valuable for our target group, with respect to the intentions we had defined specifically for our project (see section 5.11). In particular, we had to respect the primary objective of extended general education. For this purpose, we applied the concept of Fundamental Ideas of CS of A. Schwill:

“A fundamental idea with respect to some domain (e.g. a science or a track) is a schema for thinking, acting, describing or explaining which

1) is applicable or observable in multiple ways in different areas (of the domain) (horizontal criterion),
2) may be demonstrated and taught on every intellectual level (vertical criterion),
3) can be clearly observed in the historical development (of the domain) and will be
4) relevant in the longer term (criterion of time), and
5) is related to everyday language and thinking (criterion of sense)” [Schwill 1993].

Regarding the structure of relevant subject matter knowledge, we decided to chose the fundamental process of information processing as the central idea [Hubwieser and Friedrich 1998]. It is built around the concept of an Informatics System (IS), which we define as a combination of any hard- and software that is able to accept a representation of information in a machine readable form as its input, processes or transports this representation and presents it at the end in a different representation that might be interpreted by a human being or by other Informatics Systems (e.g. as embedded system), see figure 7. Generally, IS are characterized by three features
Empfehlungen der Gesellschaft für Informatik e.V. 1999: automation, interaction and interconnectness. As a consequence of the latter, in many cases the processing of data is distributed over several physical machines.

The processing or the transport is prescribed by certain rules that are called “program” or “protocol” [Hubwieser 2000]. Examples might be a spreadsheet that accepts the data in a table and calculates a graphical representation, a search engine in the Internet, a cellular phone or a robot.

![Diagram of information processing](image)

Figure 7: The fundamental process of information processing [Breier and Hubwieser 2002].

Following this scheme, we could arrange all relevant knowledge fields around the three stages of this process:

1. **Representation of information**: data as representations on which information processes operate: data models and data structures, memory technologies, variable concept, and modeling techniques for the representation of information about systems: control flow (algorithms, state and sequence charts), decomposition in subsystems, communication with its environment and between the subsystems (data flow, object and class models, component diagrams).

2. **Processing and transport of representations**: representation of processing prescriptions: algorithms, programs and programming languages, syntax, semantics, models of IS (as a description of the processing machines), application fields of IS, limitations of computability, costs (complexity, efficiency), chances and risks of IS, and interactions of IS with their environment in temporal, spatial, human and social context: history, development, operation, service, ergonomics, consequences on the private and professional life of users.

It is apparent that all possible knowledge elements out of this list can be divided into four categories. They might be applicable and important:

1. generally, i.e. beyond the limits of automatic information processing, e.g. modeling techniques, which can be applied to real world systems,
2. for all IS, e.g. algorithms, principal limitations of computability,
3. for a certain class of IS, e.g. concept of register machine, data structures of text processors or spreadsheets, principles of object-oriented programming, and
4. for a certain instance of IS, e.g. menu structure of MS Word 2010, how to fix a favorite URL in Firefox 3.0, syntax elements of Java 2.0.

For the acceptance of a knowledge element in our curriculum we demanded
- to belong to category 1 or 2 of the list above or to category 3, if the class of IS is very important for the user group (e.g. text processors) and
- to fulfill all the criteria of A. Schwills fundamental ideas with the restriction of the vertical criterion (2) to “several intellectual levels” instead of “every intellectual level”.

This criteria turned out to be very helpful during the long and anfractuous, 11-year lasting process of curriculum development, where many different persons, opinions and interests struggled. Finally, guided by this filter, we agreed to a list of knowledge elements that are listed in the curriculum in table 9 in the appendix. A comparison with our original conception [Hubwieser 2000] shows that many parts of the original concept made their way into the curriculum at the end (see table 9 in the appendix). In [Hubwieser 2006] I have illustrated the curriculum from grade 6 to grade 10 by demonstrative examples.

In our original proposal we had postulated that the covered knowledge
- should take the modern software development process as a guideline,
- focus on modeling activities instead of “instant coding” and
- abstract from specific details of hard- and software, particularly from specific software products or programming languages [Hubwieser et al. 1997].

Also, respecting Schwills claim that the object-oriented approach would be the most suitable for introductory courses [Schwill 1995], I had proposed in [Hubwieser 2000] a specific concept for grade 6 and 7 that was based on object-oriented modeling of standards software documents like graphics, texts or hypertexts. This idea was inspired by the prior work of U. Freiberger about the object-oriented modeling of word processors [Freiberger 1988-1990].

<table>
<thead>
<tr>
<th>Knowledge Element</th>
<th>Object-oriented concept</th>
<th>Examples, Tools, Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data structures of standard software</td>
<td>Object and class models of documents, attributes, methods</td>
<td>Graphics, Text, Hyper-text, Presentation</td>
</tr>
<tr>
<td>Algorithm</td>
<td>State and activity models, methods</td>
<td>Karol, Scratch, etc.</td>
</tr>
<tr>
<td>Function</td>
<td>Functional models</td>
<td>Spreadsheets</td>
</tr>
</tbody>
</table>
Data bases

<table>
<thead>
<tr>
<th>Records as objects, tables as classes</th>
<th>Relational data base systems</th>
</tr>
</thead>
</table>

Computer networks, Internet

<table>
<thead>
<tr>
<th>Parallel collaboration of objects, defined by protocols</th>
<th>E-Mail, Client-Server systems, network protocols</th>
</tr>
</thead>
</table>

Formal languages

<table>
<thead>
<tr>
<th>Syntax of programming languages and protocols</th>
<th>Java, BNF</th>
</tr>
</thead>
</table>

Machine level programming

<table>
<thead>
<tr>
<th>Object-oriented models of processors and computer systems</th>
<th>Assembly programming, Class PROCESSOR</th>
</tr>
</thead>
</table>

During the curriculum development it turned out that the object-oriented approach could even serve as a conceptual framework that contained all other knowledge elements that we had decided to be encompassed in the curriculum (see table 2).

Additionally, the object-oriented concept had the following advantages in our opinion:

- authenticity (common use in the professional software industry),
- closeness to the “real professional world”,
- choice of several sophisticated modeling techniques (UML) with different views, well-engineered syntax and semantics many suitable, free tools,
- enforcement to design a model before coding, because it is very difficult to write object-oriented programs without any model.

The most demanding challenge of the course design was the combination of modeling techniques with suitable software systems or programming environments that should allow the students to implement their models as soon as possible after they had created them. Thus, for every new modeling technique, we had to look for systems that were suitable to the specific views of this techniques and that would enable the students to implement their models immediately after they had created them. The result was the combination of modeling techniques and software or programming systems respectively. For example we found that functional models, as far as they consist of functions and data flows, could be implemented directly on spreadsheets. Table 9 of appendix shows the other results of this efforts.

Concerning the arrangement of the knowledge elements over the different grades we followed a radical strictly models and objects-first approach [Diethelm 2007a] that represents a combination of sequences 1.a and 1.b. according to [Bennedsen and Schulte 2008] (see section 2.2) by interlacing the teaching of concepts and the creation of classes. The students pass three consecutive stages of competencies [Hubwieser 2006]:

1. In the grades 6 and 7 the student manipulate objects, model standard software documents, e.g. vector graphics, texts or hypertexts, supplemented by some basic concepts of algorithms. Additionally, they learn the concepts of aggregation, association and create some methods.

2. In grades 9 an 10 the students model real world problems and implement their models on specific software systems, starting with spreadsheets and data base systems and finally using object-oriented programming environments. They create methods, associations and classes (implemented by data base records without methods). In grade 10 they start to create classes in general, followed by the concepts of inheritance and polymorphism.
In grade 11 and 12 the students learn important concepts out of specific knowledge fields, e.g. data structures, concepts of software engineering or theoretical CS.

This stages correspond in some respect to the levels that were presented more than 10 years later by the CSTA in their recent Computer Science Standards [Tucker et al. 2011]:

- Level 1: CS and me,
- Level 2: CS and community,
- Level 3: Applying concepts and creating real-world solutions
  3-A CS in the modern world,
  3-B CS principles, and
  3-C Topics in CS.

The \textit{strictly models and objects-first} approach was chosen out of three reasons: Firstly, at the time we had to set the cornerstones of our concept, programming was not very popular in German schools. As a consequence of the uncoordinated and often quite unprepared programming courses (mostly in Pascal or Basic) of the early 90ies, mostly taught by teachers without any degree in CS, (see section 5.3), the enrollment to these courses had dropped almost to zero. Therefore it seemed not wise to start the course with any form of programming or to set the focus of the courses on programming.

Secondly, as explained in [Hubwieser et al. 1997], we had proposed to set the focus of our courses on certain object-oriented modeling techniques. Therefore it seemed natural to start with modeling (of objects in this case) and implement those models in a second step, as usual in professional software development processes. At the end of the course the students should gain the conviction that it might be helpful in many cases to model a system without any implementation, while, on the other hand, programming without any modeling might be dangerous and unprofessional mostly.

Thirdly, demanded by parents and administration, the courses would have to cover user skills on common of software systems, e.g. drawing systems, word processors, spreadsheets or presentation software. On the other hand, this should take place at lower grades, because these knowledge and skills would be needed in other subjects. The problem was that the training of user skills did not meet the usual requirements on learning content (mainly concerning generality and abstraction) of the Bavarian grammar schools. Yet, in combination with the objects-first approach, we found a well-tailored solution to incorporate this into the curriculum. As explained already above, the basic idea was to have the students develop object-oriented models of the documents of those software systems and regard the produced documents as implementations of such models.

As far as we have evidence about the success of this conception, this is presented in section 6. Particularly from the teachers we received quite positive reactions, which might be indicated e.g. by the fact that our cluster analysis of their attitudes towards the curriculum [Mühling et al. 2010] showed that the largest cluster by far (44%) consist of “fans of the curriculum” or by the contentedness and the missing of any suggestions to change the conception according to the survey of the state institute ISB from 2011 [ISB 2012] (see section 6 for both surveys).

5.13 Teaching Methods and Motivation of Students

According to [Biggs 1992] the highest level of teaching focuses on “what the student does”: 
“This implies a view of teaching that is not just about facts, concepts and principles to be covered and understood, but about:

1. What it means to **understand** those concepts and principles in the way we want them to be understood.
2. What kind of **TLAs** (teaching/learning activities) are required to reach those kinds of understandings.”

In fact it was our main concern during the development of the subject and the textbooks to have an idea what “real learning activity” the student might show regarding the different curriculum topics. “Listening”, “understanding”, “reading” were not satisfying in this respect. More generally speaking, we tried to follow the principles of constructivism as far as possible [Ben-Ari 1998].

According to the approach of **cognitive apprenticeship** [Collins et al. 1989], to motivate students and to offer them examples of realistic complexity, we suggest the students to work on authentic problems and contexts [Cooper and Cunningham 2010] as far and as early as possible. Concerning the structure of the lessons we try to arrange all learning objectives in and around larger modeling and programming projects. The information input by the teachers should be as short as possible, usually not exceeding 5-10 minutes within each lesson. In application of the principle of anchored instruction [Cognition and Technology Group at V 1992], the projects should be defined by scenarios, that are told as stories, offering narrative anchors by this way. Intensive cooperation within task sharing working groups should offer sufficient occasions for social learning.

Because the first object-oriented program is intellectually very demanding for the students [Hubwieser 2007a], we suggest to respect the **Cognitive Load Theory** [Chandler and Sweller 1991]. As the **Intrinsic cognitive load** is quite high in our context, the **extraneous cognitive load** should be kept as low as possible by reducing all disturbing factors and keeping the working environment as simple as any possible. Therefore, we suggest to use fading worked examples according to Gray [Gray et al. 2007] for the introduction of programming concepts, starting with a fully worked out program fragment, followed by less and less worked out examples until the students are able to implement the regarded concept on their own from the scratch.

As the lessons should be dominated by student projects, we have proposed a certain structure of these projects [Hubwieser et al. 1997]. The lesson should start with an **introduction** of the problem, presenting the task as illustrative as possible, aiming at motivation, followed by a detailed but still **informal description** of the problem, using e.g. word-processors. After discussing the informal description the central part of the lesson begins: the formal **modeling phase**, applying didactically reduced modeling techniques from software engineering, e.g. data flow diagrams, state charts, class models or sequence charts. In most cases it will do to construct 1-2 models per project, depending on the characteristics of the problem. The type of the models will decide the appropriate **implementation** platform. The main goal here is to simulate, validate and illustrate the modeling results, not to train skills on specific software systems or sophisticated programming abilities. The last step is a critical **review** and discussion of the achievements, concerning alternatives, possible improvements, correspondences with problems solved already, as well as the social consequences of the produced solution.

### 5.14 Media

Following the principle of **holistic education**, we should try to use as many haptic media as possible, similar to the “algorithms unplugged” concept [Vöcking 2011]. There are many occasions in CSE that suggest such media, e.g. visualizing hypertext structures on a pin board by sheets of paper, representing the documents, and threads of wool for the links, or using a paper box to represent a variable, writing its
denominator (name) on the front and putting a ping-pong ball inside for the value. These media may also help to correct misconceptions [Weigend 2006].

Concerning the digital media, we faced the challenge to find a suitable software system for every topic of the curriculum that enables the students to implement the concepts they had learned, as already mentioned above. The result is displayed in table 9 in the appendix.

There are basically three categories of activities the students conduct by using these software systems:

1. producing documents using standard software or data base systems in order to create and manipulate objects, classes and their associations,
2. producing spreadsheet formula in order to implement functional models, and
3. developing programs (e.g. for robot systems or in Java) to implement algorithms, state machines and/or models of classes respectively.

Besides that several software systems were produced especially for this course, e.g. Object-Draw (not to be confused with the objectdraw library by [Bruce et al. 2006]) and EOS, both developed by Martin Pabst (www.pabst-software.de) or Robot Karol by U. Freiberger and O. Krsko (www.schule.bayern.de/karol).

Principally, we did not prescribe any specific software system or programming language. In contrary, the schools are free to use any software system they regard useful. Nevertheless, the BlueJ environment that was developed by M. Kölling (www.bluej.org) plays a very important role at least historically, because the teaching concept of the 10th grade was inspired heavily by its features. Particularly the BlueJ project shapes that comes with the software enables us apply the objects-first principle once again, this time on the “real” programming part of the course in the 10th grade.

The students start this part by manipulating the attributes of ready-made objects (e.g. circles or rectangles) and by calling their methods interactively in the BlueJ IDE. By this way they rediscover the same classes, attributes and methods that they have learned at the beginning of the course in grade 6. Additionally, the objects that are existing in runtime are visualized by BlueJ, which might help them to understand the crucial distinction between object and classes (see figure 8).

5.15 Educational Standards
As for all other subjects, the intended learning outcomes of CS courses should be defined by suitable standards instead of curricula. Nevertheless, it is still a long way to that goal. Regarding the current situation in Germany, there isn’t even a dedicated
subject of CS in most of the states, except e.g. Saxony and Bavaria (see above). As described by [Klieme et al. 2004], the agreement on educational standards is eased by the existence of suitable competency models, similar to the competency model of the [National Council of Teachers of Mathematics (NCTM) 2000] as a foundation of several standards in Mathematics. Currently the MoKoM project [Magenheim et al. 2010] aims to develop the first CS competency model in Germany that is based on evidence. As soon as an adequate competency model is available, a standardization process that is comparable to traditional subjects could start. In Germany the Kul-tusministerkonferenz (see section 5.6) is in charge for deciding about such standardizations, which already happened e.g. for Mathematics in grade 10. This process was described in detail by [Klieme et al. 2004].

In a first preliminary step, a quite arbitrarily composed group of teachers and researchers from the German Informatics Society (Gesellschaft für Informatik) worked out a proposal for standards [Gesellschaft für Informatik e. V. 2008]. Apparently these standards were inspired in many respects by our concept. Regarding the values of dimension 1 of the Darmstadt Model, we have influenced at least the unofficial standards in Germany. Therefore, we had a nationwide influence at this point in some respect.

An informative overlook of the current international situation concerning standards is given by [Micheuz 2008].

As far as the final graduation from German grammar schools (called Abitur) is concerned, there are formally decided standards for all subjects, called Einheitliche Prüfungsanforderungen in der Abiturprüfung (EPA). The currently valid EPA (decided in 2008) of CS was developed in due consideration of our course concept.

For the US, the CSTA has developed and published its K–12 Computer Science Standards in 2006 [Tucker et al. 2006] and revised in 2011 [Tucker et al. 2011]. In 2010 the CSTA investigated the adoption of these standards (version 2006) by the 50 US states, applying a quite liberal methodology [Wilson et al. 2010]:

“If the state standards made any reference to the general idea detailed in the ACM/CSTA standards, it was marked as adopted.”

The result was a very different implementation percentage, varying from 0% to 100% [Wilson et al. 2010]. Applying the same methodology to the curriculum of our Bavarian subject, we find the percentages that are shown in tables 3 and 4.

<table>
<thead>
<tr>
<th>Standard Level</th>
<th>CSTA Standards</th>
<th>Adopted in Bavaria in any grade</th>
<th>Adopted in higher grades than proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number</td>
<td>Number</td>
<td>Percentage</td>
</tr>
<tr>
<td>I Foundations of Computer Science</td>
<td>35</td>
<td>17</td>
<td>48.6%</td>
</tr>
<tr>
<td>II Computer Science in the Modern World</td>
<td>10</td>
<td>9</td>
<td>90.0%</td>
</tr>
<tr>
<td>III Computer Science as Analysis and Design</td>
<td>10</td>
<td>7</td>
<td>70.0%</td>
</tr>
<tr>
<td>Total</td>
<td>55</td>
<td>33</td>
<td>60.0%</td>
</tr>
</tbody>
</table>

It is apparent that our course is focused predominantly on the higher intellectual levels II and III, neglecting level I. In addition we have adopted about one third of the standards too late, i.e. in higher grades than proposed by the CSTA.

In 1999 the Committee on Information Technology Literacy of the US National Research Council presented its report Being Fluent with Information Technology. The Components of “fluency with information technology” were grouped according to three categories:

- Intellectual Capabilities,
- Information Technology Concepts, and
- Information Technology Skills [US National Research Council 1999].

Wilson et al. adopted these categories in their report from 2010 by grouping the CSTA standards accordingly [Wilson et al. 2010]. If we transfer this categorization to our project, we find the percentages that are displayed in table 4.

<table>
<thead>
<tr>
<th>Type</th>
<th>CSTA Standards</th>
<th>Adopted in Bavaria in any grade</th>
<th>Adopted in higher grades than proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number</td>
<td>Number</td>
<td>Percentage</td>
</tr>
<tr>
<td>Capabilities</td>
<td>22</td>
<td>7</td>
<td>31.8%</td>
</tr>
<tr>
<td>Concepts</td>
<td>19</td>
<td>18</td>
<td>94.7%</td>
</tr>
<tr>
<td>Skills</td>
<td>14</td>
<td>8</td>
<td>57.1%</td>
</tr>
<tr>
<td>Total</td>
<td>55</td>
<td>33</td>
<td>60.0%</td>
</tr>
</tbody>
</table>

This shows that our subject is strongly focused on conceptual knowledge, but quite weak in respect to intellectual capabilities, at least compared to the proposals of the CSTA.

6. RESEARCH AND RESULTS

In 2009 the Bavarian Gymnasia were attended by 285,518 students. Table 5 displays the number of students that have enrolled in our CS courses, according to current data [ISB 2009].

<table>
<thead>
<tr>
<th>Grade</th>
<th>Students of grammar school in 2009</th>
<th>Start of CS course</th>
<th>Percentage of students in CS course/exam</th>
<th>Number of students in the CS course/exam per year</th>
<th>Total number of CS students until now</th>
<th>Information source</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>47434</td>
<td>2004 (NuT)</td>
<td>100.0%</td>
<td>47434</td>
<td>332038</td>
<td>*</td>
</tr>
<tr>
<td>7</td>
<td>44752</td>
<td>2005 (NuT)</td>
<td>100.0%</td>
<td>44752</td>
<td>268512</td>
<td>*</td>
</tr>
<tr>
<td>9</td>
<td>42732</td>
<td>2007 (IN)</td>
<td>49.3%</td>
<td>21067</td>
<td>84268</td>
<td>*</td>
</tr>
<tr>
<td>10</td>
<td>41069</td>
<td>2008 (IN)</td>
<td>49.3%</td>
<td>20247</td>
<td>60741</td>
<td>*</td>
</tr>
<tr>
<td>11</td>
<td>35147</td>
<td>2009 (IN)</td>
<td>14.2%</td>
<td>5000</td>
<td>10000</td>
<td>***</td>
</tr>
<tr>
<td>12</td>
<td>33603</td>
<td>2010 (IN)</td>
<td>7.0%</td>
<td>2342</td>
<td>2342</td>
<td>**</td>
</tr>
<tr>
<td>Graduation exam in CS (Abitur)</td>
<td>31906</td>
<td>2011</td>
<td>2.4%</td>
<td>772</td>
<td>772</td>
<td>**</td>
</tr>
</tbody>
</table>

* Educational report of Bavaria 2009 [ISB 2009]
** Preliminary data from graduation statistics 2011, orally from the Bavarian administration
*** Estimated according to the sales figures of textbooks

The investigation of the actual learning outcomes of the students is very difficult in Bavaria, because the government is strictly prohibiting almost all surveys among students (except surveys that are proposed by cooperatives of reputable scientists or selected international large scale surveys, e.g. PISA), claiming that otherwise there would be too many disturbances of the daily work in school. Even in the principally accepted cases it takes a long time (and piles of paper) to get a permit for a proposed
survey. Therefore, we are restricted at the moment to official results that are published by the administration, to surveys among teachers or to surveys among graduates. Nevertheless we are preparing a large scale survey for the future.

Every year the Bavarian ISB (see section 5.1) conducts a voluntarily test about learning outcomes in grade 6. The participation of each class is decided by its teachers. The results show that the participation in the CS part of this test is similar to the Biology part (attended by 60-80% of all classes in Bavarian grammar schools). The average scores of the CS items are quite similar to Biology, also (see figure 9 for the results of 2011 [ISB 2010]).

On the other end of the course, the results from the first graduation exams in 2011 are quite spectacular, possibly due to the low number of students that had chosen CS as (optional) subject for the final exams (772 student or 2.4%, see table 5 and section 5.2). The average score in CS was 1.65, compared to an average of 2.27 over all subjects on a six-valued scale, where 1.0 is the best score. For the second generation of graduation exams in 2012, we expect a substantial higher number of participants and therefore a decrease of the average scores.

In 2007 we investigated, how the results of students of Bavarian grammar schools in the international Bebra contest have been influenced by the our new subject. We found that there were significant improvements compared to the students of two other German states with comparably high participation rates (the largest state Nordrhein-Westfalen and Brandenburg, which has a very different educational concept for CS), as far as our students had already attended our course as well as compared to Bavarian students that did not attend any CS course yet [Steer and Hubwieser 2010].

In 2009 we conducted a survey among all Bavarian CS teachers. At this time the first age cohort of students had just completed the 10th grade of our new subject. We contacted more than 1000 teaching persons, using the E-Mail lists of all Bavarian universities that offer regular teacher education programs in CS. 448 teachers have answered the online questionnaire more or less completely. As reported in [Mühling et al. 2010], we found four interesting clusters of teachers as far as their attitudes towards our objects-first concept is concerned. According to their specific preferences, we called them “office users” (25%), “fans of the curriculum” (44%), “anti-programmers” (13%), “traditional computer scientists” (18%). The second and largest
group has a particularly positive attitude towards the course and our object-oriented concept, while the teachers of the last group are less content of themselves and think less positively about the students’ attitudes towards the subject than the other clusters.

Table 6. Modal values of several indicators for the success of the subject ([Hubwieser and Mühling 2009])

<table>
<thead>
<tr>
<th>indicator</th>
<th>Grade 6</th>
<th>Grade 7</th>
<th>Grade 9</th>
<th>Grade 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satisfaction with the participation of the students*</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Students Attitude to CS*</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Completion of curriculum*</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Satisfaction with the CS course*</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Feedback from parents**</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

*1=totally satisfied, 2=predominantly satisfied, 3=more or less satisfied, 4= more or less dissatisfied, 5 = predominantly dissatisfied, 6=totally dissatisfied.

** 1= no feedback, 2=strongly positive feedback, 3=mostly positive, some negative feedback, 4=mostly negative, some positive feedback, 5=strongly negative feedback.

It seems that predominantly young teachers are involved in the new subject, as the modal value of the professional experience was “5–10 years”. While 46% of the teachers declared that they would have a university degree in CS, 42% admitted not to have one.

Apparently, the course is running quite successfully in the eyes of the teachers according to these results. Similarly, the motivation of the teachers seems to be quite high. On the question “Do you like to teach CS?”, 72% of the teachers responded “Yes”, only 4% “No”.

It was a matter of particular concern for us that the course would be accepted by the girls in the same way as by the boys. Therefore, we asked if the performance of boys or girls would have been better according to the teachers’ view. The result was quite positive, except for the apparent decrease of the girls performance in grade 10 (see figure 10).

Figure 10. Comparison of the performance of boys and girls as reported by teachers

Similarly to the latter, all other indicators (see table 6) were still positive, but nevertheless decreasing in grade 10. Apparently the introduction into object-oriented programming in this grade represents one of the most crucial points of our concept. This has to be improved in the future. The results of the survey can be downloaded from www.ddi.tum.de/forschung/lehrerbefragung-2009.
Most recently the Bavarian State Institute for School Quality and Educational Research (ISB) (see section 5.1) conducted a survey among all Bavarian grammar schools [ISB 2012]. According to the answers of 463 teachers, most if them are very content with the items of the new curriculum of CS (see Figure 11). The items of the curriculum are listed in table 9 in the appendix. Apparently the most problematic items (contentedness below 80%) are 7.2.2 (Exchange of information by E-Mail), 10.3 (Software project) 11.2.2 (Software engineering), 12.4 (Limitations of computability). We will have to investigate the reasons for these results in the future.

![Figure 11. Contentedness of the Bavarian CS teachers with the curriculum [ISB 2012].](image)

In October 2011 we investigated the educational background of the freshman at the Faculty of Informatics at our Technische Universität München, aiming to get information about the consequences of our project for the enrollment at CS faculties. As a consequence of the doubled number of students that had graduated from grammar school this year in Bavaria (first of G8 and last of G9 simultaneously, see section 5.4), our faculty could welcome approximately 1200 “official” freshmen, compared to about 750 in the year before. This number includes approximately 500 selected students from the old type G9, who had already attended a specific, extraordinary program (TUMTwoInOne). This program aimed to teach the first two semesters of the regular course of study already during summer, so that the attendees could switch to the 3rd
semester in autumn 2011. Nevertheless, they started their course of study “officially” in October, as all other freshmen. Therefore, we estimate that about 700 “actual” freshmen (who had not attended TUMTwoInOne) had enrolled at the faculty in October 2011.

Due to the organizational circumstances, we were able to present our questionnaire to about 450 of these, which resulted in 375 completed questionnaires, representing 53.5% of the supposed “actual” freshmen. The analysis of the data yielded the following distributions: 60.4% of the students had graduated from a school in Germany, 80.2% of those in Bavaria. Among the Bavarian students 56.5% had attended the new G8, 36.2% the old type G9. Among the graduates from G8, we found the educational background in CS that is listed in table 7.

<table>
<thead>
<tr>
<th>Part of CS course</th>
<th>Attended by*</th>
<th>Percentage of freshmen**</th>
<th>Percentage in general***</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-7 NuT</td>
<td>153</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>9-10 IN</td>
<td>120</td>
<td>78.4%</td>
<td>49.3%</td>
</tr>
<tr>
<td>11 IN</td>
<td>110</td>
<td>71.9%</td>
<td>14.2%</td>
</tr>
<tr>
<td>12 IN</td>
<td>105</td>
<td>68.6%</td>
<td>7.0%</td>
</tr>
<tr>
<td>Graduation exam in CS (Abitur)</td>
<td>81</td>
<td>52.9%</td>
<td>2.4%</td>
</tr>
</tbody>
</table>

* Number of responding freshmen from G8 that had attended the respective course part
**Percentage of those attendants among the 153 responding freshmen from the Bavarian G8.
*** Percentage of students in CS course/exam in G8 in general (see table 5).

Multiplying by the factor 1/0.535 due to the portion of freshmen represented by the responses on our questionnaire.

This figures show that the attendees of the CS courses are strongly overrepresented among the freshmen at our Faculty of Informatics. While in the G8 in general only 7.0% of the students of grade 12 had enrolled for CS, this percentage is almost 10 times as high (68.6%) among our freshmen. The percentage of students that had chosen CS as graduation subject looks even more dramatic (2.4% vs. 52.9%), where the overrepresentation factor is nearly 22.

As we could ask only 375 of the assumed 700 “actual” freshmen (53.6%), we can extrapolate that approximately 151 of the newly enrolled students had passed the graduation exam in CS at G8 (see section 11), while about 196 had attended our CS course for six years. This means that about 9% of all Bavarian students that had attended the total CS course at grammar school and about 20% of all students that had passed the graduation exam in CS (see table 5) had decided to enroll for CS as major subject at our University. As there are eight more Universities and additionally 13 Universities of Applied Sciences in Bavaria that offer CS as a major subject, we can suppose that nearly all students that had attended the total CS course have enrolled in CS as a major subject. This might be taken as an indicator that these students had liked the subject of CS at grammar school so well that a majority of them even had decided to chose a career in CS.

7. DISCUSSION

First of all, it turned out that the Darmstadt Model provides a very practical checklist for case studies about CS in secondary education. After completing this report according to the Darmstadt Model, I have the strong feeling that all relevant issues have been discussed, even though not in every detail. Nevertheless, there are some
changes to the Darmstadt Model that should be considered. First, the category Research should be refined and specified more precisely. Maybe it will form a fourth dimension on its own. Second, the dimensions Berlin Model Top Dimension and Level of Responsibility/Range of Influence should be renamed. Additionally, it is apparent that all categories should be defined closely according to literature.

As far as our project is concerned, most indicators presented in this paper show that our concept works quite well. Nevertheless, it has to be taken into account that it was designed for (and implemented in) a very special school type. The students of the Bavarian grammar schools are selected very early and quite rigidly, as only about one third of an age group is admitted. It is pretended that this selection is based on giftedness, abilities or talent. Besides the question whether these attributes are sufficiently constant and valid at all, the selection result is also strongly influenced by the social state and the educational background of the students, as several studies show, e.g. IGLU 2006 [Bos et al. 2008], PISA 2006 [OECD 2007] or the most recent Educational report by OECD [OECD 2011]. Due to this selection process, the Bavarian grammar school is spared from many problems that arise in the other school types. Before our concept could be applied to another school type that is attended by all students of an age group (e.g. the US high schools), it would have to be investigated thoroughly and, assumedly, changed substantially. Particularly the students would probably need much more learning time in other school types.

Nevertheless, the increasing problems in grade 10 (see section 6) show that even in the comparably “idyllic world” of our grammar school there are some weaknesses in our concept that have to be improved. In particular, the technical orientation of the programming part in grade 10 should be reduced and oriented more towards conceptual understanding instead of software production.

8. CONCLUSION AND FUTURE WORK

The experiences I have made in the 17 years I have worked on this project could be summarized in the following way: The most important preparations for the successful introduction of a new subject of CS are in my opinion

1) to define clear educational goals that are accepted by the whole society and
2) to describe the competencies the students should acquire and
3) to explain why this is a real benefit of them.

The focus should be on this benefit of the students as individuals, not on the benefits of the science of CS, universities or the society in general.

Well worked and explained exemplary teaching lessons are very helpful from the beginning. Very often misunderstandings arise from a different interpretation of terms or concepts between schools and university teachers, working at very different abstraction levels.

Our next short-term goal is to evaluate the 375 questionnaires we have taken from the freshman at our faculty (see section 27). Additionally to the questions about their educational background we have asked them

1) to describe their perception of CS as a scientific discipline and as a profession,
2) to give the people and incidents that had triggered their decision to chose CS as major subject, and
3) to draw a concept map about their perception of CS concepts.

Currently we are evaluating all these data, hoping that their analysis will yield interesting results, particularly in correlation with their educational background in CS. As far as the evaluation of the biographical data is concerned.
We are preparing a second survey among the CS teachers in Bavaria, similar to the survey from 2009 (see section 6). The focus will be on the evaluation of grade 11 and 12 according to the opinion of the teachers. Additionally, we are interested in the requirements concerning the competencies of CS teachers, similar as the COACTIV project investigated for math teachers [Max-Planck-Institut für Bildungsforschung 2011]. This research will take place within a large project about teacher competencies that is funded by the German Federal Administration.

9. ACKNOWLEDGMENTS

Many people were involved in the one or the other way over these years. Therefore, it is not possible to name all of them here explicitly, but nevertheless, I want to thank everybody who has contributed, particularly my co-authors of the textbooks. However, there are several people who are to be mentioned as those who have made the project possible at all. First of all I have to thank Manfred Broy, who hired me for the Technische Universität München, inspired me in many ways by his brilliant scientific work and supported the project by his resources and by several political initiatives. Secondly Peter Müller has to be honored as the deciding person in the administration. He took all the crucial decisions as far as he was able to do so and convinced his superiors as well as the Secretary of State to introduce the subject. His successor Dieter Götzl supported our work wherever he was able to do so. Ulli Freiberger inspired the concept of the novice courses heavily by his idea of object-oriented modeling of word processor documents [Freiberger 1988-1990]. He was always willing to help, in curriculum development as well as in teacher education. He developed the Robot Karol, spending long nights of coding. Among all my staff members, Ferdinand Winhard, Matthias Spohrer, Markus Steinert and Siglinde Voß spent most time and effort, from the first teacher education courses to the last textbook. Albert Wiedemann contributed many concepts and good ideas to the curriculum. Finally I have to thank Stefan Stöckle, who made our textbooks possible, coordinating the authors, contributing by excellent ideas and never giving up in trouble.

10. APPENDIX

Table 8. Assignment of the discussed issues according to the 3 dimensions of the Darmstadt Model

<table>
<thead>
<tr>
<th>Dim. 1: Range of Influence</th>
<th>Dim. 2: Berlin Model Top Category</th>
<th>Dim. 3: Educational Relevant Areas</th>
<th>Covered by section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dim 1: Range of Influence</td>
<td>Dim 2: Berlin Model Top Category</td>
<td>Dim 3: Educational Relevant Areas</td>
<td>Covered by section</td>
</tr>
<tr>
<td>P=Precondition</td>
<td>D=Decision, C=Consequence</td>
<td>Level</td>
<td>1</td>
</tr>
<tr>
<td>None</td>
<td>P</td>
<td>Educational system</td>
<td>5.1</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Organizational aspects of subject</td>
<td>5.5</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Degree of compulsion</td>
<td>5.5</td>
</tr>
<tr>
<td>Country</td>
<td>FC</td>
<td>Enrollment</td>
<td>5.1, 5.5, 6</td>
</tr>
<tr>
<td>State</td>
<td>P</td>
<td>School type</td>
<td>5.1, 5.5</td>
</tr>
<tr>
<td></td>
<td>Socio-Cultural related Factors</td>
<td>5.8, 5.9</td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>P</td>
<td>History of ICT and Informatics in School</td>
<td>5.3</td>
</tr>
<tr>
<td>None</td>
<td>D</td>
<td>Age</td>
<td>5.8</td>
</tr>
<tr>
<td>None</td>
<td>P</td>
<td>Gender</td>
<td>5.8</td>
</tr>
<tr>
<td>None</td>
<td>P</td>
<td>Social and Immigration Background</td>
<td>5.8</td>
</tr>
<tr>
<td>None</td>
<td>P</td>
<td>Family Socialization</td>
<td>5.9</td>
</tr>
<tr>
<td>State</td>
<td>P</td>
<td>Public opinion</td>
<td>5.9</td>
</tr>
<tr>
<td>None</td>
<td>P</td>
<td>Techno-economic development</td>
<td>5.7</td>
</tr>
<tr>
<td>State</td>
<td>Policies</td>
<td>5.4</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>P</td>
<td>Research and Funding Policies</td>
<td>Not relevant</td>
</tr>
<tr>
<td>State</td>
<td>P</td>
<td>Education Policies</td>
<td>5.4</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Enhancing Cooperation</td>
<td>Not relevant</td>
</tr>
<tr>
<td>State</td>
<td>P</td>
<td>Technical Infrastructure</td>
<td>5.7, 5.14</td>
</tr>
<tr>
<td>State</td>
<td>P</td>
<td>Financing Initiatives</td>
<td>Not relevant</td>
</tr>
<tr>
<td>State</td>
<td>P</td>
<td>Quality Management</td>
<td>5.1, 6</td>
</tr>
<tr>
<td>-------</td>
<td>---</td>
<td>-------------------</td>
<td>-------</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Teacher Qualification</td>
<td>5.6</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Teacher Education</td>
<td>5.6</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>CSE Teacher Education</td>
<td>5.6</td>
</tr>
<tr>
<td>State</td>
<td>P</td>
<td>Certification</td>
<td>5.6</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Training</td>
<td>5.6</td>
</tr>
<tr>
<td>State</td>
<td>C</td>
<td>Professional Experience</td>
<td>5.6, 6</td>
</tr>
<tr>
<td>State</td>
<td>C</td>
<td>Motivation</td>
<td>5.13, 6</td>
</tr>
<tr>
<td>State</td>
<td>C</td>
<td>Student</td>
<td>5.13,</td>
</tr>
<tr>
<td>State</td>
<td>C</td>
<td>Teacher</td>
<td>5.6, 6</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Intentions</td>
<td>5.11</td>
</tr>
<tr>
<td>Country</td>
<td>C</td>
<td>Standards</td>
<td>5.15</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Competencies</td>
<td>5.11</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>CSE</td>
<td>5.11</td>
</tr>
<tr>
<td>None</td>
<td>P</td>
<td>Interdisciplinary</td>
<td>5.11</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Learning Objectives</td>
<td>5.11</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Knowledge</td>
<td>5.12</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Computer Science</td>
<td>5.12</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>ICT</td>
<td>5.12</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Curriculum Issues</td>
<td>5.10</td>
</tr>
<tr>
<td>Country</td>
<td>D</td>
<td>Examination/Certification</td>
<td>5.2</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Teaching Methods</td>
<td>5.13</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>CSE</td>
<td>5.13</td>
</tr>
<tr>
<td>None</td>
<td>P</td>
<td>General Education</td>
<td>5.13</td>
</tr>
<tr>
<td>None</td>
<td>P</td>
<td>Extracurricular Activities</td>
<td>6</td>
</tr>
<tr>
<td>None</td>
<td>P</td>
<td>Contest</td>
<td>6</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Media</td>
<td>5.14</td>
</tr>
<tr>
<td>State</td>
<td>P</td>
<td>Technical infrastructure</td>
<td>5.7</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Textbooks</td>
<td>5.14</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Tools</td>
<td>5.14</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Didactical software</td>
<td>5.14</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Visualization software</td>
<td>5.14</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Unplugged Media</td>
<td>5.14</td>
</tr>
<tr>
<td>State</td>
<td>D</td>
<td>Haptic media</td>
<td>5.14</td>
</tr>
<tr>
<td>International</td>
<td>PC</td>
<td>Research</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Grade</th>
<th>Theme</th>
<th>Basic concepts</th>
<th>Implementation software</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>6.1</td>
<td>(Biology)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.2.1</td>
<td>Information and Representation</td>
<td>Representation of information, interpretation</td>
</tr>
<tr>
<td></td>
<td>6.2.2</td>
<td>Representation of information by graphic documents</td>
<td>Object-oriented modeling of documents: object, attribute, class, method, aggregation</td>
</tr>
<tr>
<td></td>
<td>6.2.3</td>
<td>Representation of information by text documents</td>
<td>Aggregation</td>
</tr>
<tr>
<td></td>
<td>6.2.3</td>
<td>Representation of information by multimedia documents</td>
<td>Animation by methods, combination of graphics, text and multimedia</td>
</tr>
<tr>
<td></td>
<td>6.2.4</td>
<td>Hierarchical information structures</td>
<td>Tree, root, leaf, node, edge</td>
</tr>
<tr>
<td></td>
<td>7.1</td>
<td>(Physics)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7.2.1</td>
<td>Network structures</td>
<td>Link, anchor, Internet, cyclic structures, reference, Hypertext</td>
</tr>
<tr>
<td></td>
<td>7.2.2</td>
<td>Exchange of information</td>
<td>E-Mail, attachment, mail server, principles of E-mail transfer</td>
</tr>
<tr>
<td></td>
<td>7.2.3</td>
<td>Basic concepts of algorithms</td>
<td>Representation of algorithms, control structures (sequence, conditional, repetition)</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>9.1 Functional modeling</td>
<td>Data flow diagrams, function, parameters, return value, concatenation, simple data types</td>
</tr>
</tbody>
</table>

Table 9. Overview: knowledge, concepts and media of our CS curriculum
9.2.1 Object oriented data models
9.2.2 Requirements on a data base systems
9.2.3 data protection, data security
9.2.4 Complex application example

10 10.1.1 Summary and consolidation of prior knowledge
10.1.2 States of objects and algorithms
10.1.3 Communication of objects
10.2 Generalization and specialization
10.3 Software project

11 11.1.1 Recursive data structures: Lists
11.1.2 Trees and graphs
11.2.1 Software engineering: concurrent projects
11.2.2 Software engineering

12 12.1 Formal languages
12.2 Synchronization of parallel processes
12.3 Basic functionality of a computer
12.4 Limitations of computability

<table>
<thead>
<tr>
<th>Competency model</th>
<th>Level/competency</th>
<th>Chapters of our curriculum (see table 9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeSeCo</td>
<td>1-A: The ability to use language, symbols and text interactively</td>
<td>6.2, 12.1</td>
</tr>
<tr>
<td></td>
<td>1-B: The ability to use knowledge and information interactively</td>
<td>6.2, 7.2.1, 7.2.2, 9.2</td>
</tr>
<tr>
<td></td>
<td>1-C: The ability to use technology interactively</td>
<td>6.2, 9.1, 9.2, 10.1, 10.2, 10.3, 11.1, 12.2, 12.3, 12.4</td>
</tr>
</tbody>
</table>

Table 10. The competencies that the students should acquire over the course.
2-B: The ability to cooperate

K1: Basic competencies:
  K1.1 System Application and Development
  K1.2 System Comprehension
  K1.3 System Development

K2: Informatics views:
  K2.1 External view and Functional Analysis
  K2.2 Internal view and Internal view and Internal view

K3: Complexity

K4: Non cognitive skills:
  K4.1 Attitudes
  K4.2 Social-Communicative Skills
  K4.3 Motivational and Vocational Skills

MoKoM

REFERENCES


