Exploring teachers’ attitudes towards object oriented modelling and programming in secondary schools

Andreas Mühling  
Fakultät für Informatik  
Technische Universität München  
Boltzmannstr. 3  
85748 Garching, Germany  
andreas.muehling@tum.de

Peter Hubwieser  
Fakultät für Informatik  
Technische Universität München  
Boltzmannstr. 3  
85748 Garching, Germany  
peter.hubwieser@tum.de

Torsten Brinda  
Didactics of Informatics  
University of Erlangen-Nuremberg  
Martensstr. 3  
91058 Erlangen, Germany  
torsten.brinda@cs.fau.de

ABSTRACT

In the year 2004, Bavaria (one of the 16 federal states of Germany) started an innovative subject “Informatics” in its highest type of secondary schools (Gymnasiums). It is comprised of a compulsory stage (grades 6–10), which is followed by an eligible course in grades 11 and 12. The curriculum of the course is based on the “objects first” approach. All relevant object-oriented concepts are introduced and used in the context of standard software, before the students write their first object-oriented program. In July 2009 the first class completed the compulsory stage. We evaluated the experiences of this first run in December 2009 by a large scale study about the teachers’ opinions about and attitude towards this new subject. In this paper we present some of the first results that partly were obtained using cluster analysis.

Categories and Subject Descriptors
K.3.2 [Computers science education]: Computer and Information Science Education—Computer science education; K.3.2 [Computers science education]: Computer and Information Science Education—Curriculum; K.3.2 [Computers science education]: Computer and Information Science Education—Information systems education

Keywords
Empirical study, Computer science education, Secondary schools, Secondary education, Didactics of Informatics

1. INTRODUCTION

While there is a broad consensus that Informatics or computer science has to be taught at universities over nearly all courses of study within specific lectures, the dispute about how to teach it at secondary schools is still going on [23]. Depending on the type of school, typical learning objectives might range from “being able to use MS Office to write and print a letter” to “being able to implement a given class model in an object-oriented programming language”.

Concerning the debate, whether Informatics should be implemented in school curricula as a compulsory, dedicated subject or integrated into other subjects like Mathematics, Languages or Science, we always argued that it would be absolutely necessary to combine these two policies (firstly published in [15]). Some of our arguments in favour of a compulsory subject were the following [19]:

- Regarding the current organizational structures at German schools, it seems absolutely impossible to assure proper teacher education at university level without a regular subject. Otherwise this would be restricted to pure “in service training”.
- According to our experiences, the students do not take integrated topics seriously enough. In particular they prefer spending their learning time on regular subjects where they have to achieve grades.

After we had convinced the government, the state of Bavaria, which is the second largest state of Germany, decided in the year 2000 to introduce a new compulsory subject “Informatics” at its 405 Gymnasiums [16], starting in autumn of 2004. The Gymnasium is a specific type of secondary school, starting at grade 5 (with children at the age of 11) and, after 8 years of studies, leading to a degree that allows enrolling in universities. It offers four different directions of study (natural science/technology, foreign languages, economy and music/arts) and is currently attended by about 370,000 students. About 45,000 of them are attending grade 6 each year, about 34,000 reach grade 10. The first Informatics class entered grade 6 in autumn 2004 and completed the compulsory stage after grade 10 in summer 2009. In order to evaluate the results of this first pass, we conducted a study on the opinions of the teachers in december 2009. We designed an online questionnaire of 42 questions and asked all teachers concerned with the subject for their participation. The questionnaire was returned by 448 teachers.

In chapter 3 we present an overview over the Bavarian curriculum, which is constructed based on the “objects first” principle, introducing all relevant object-oriented concepts in the context of the usage of standard software, long before the students write their first object-oriented program. The acceptance of this principle by the teachers is one of the most important outcomes of our study. While the introduction
of a new subject causes a lot of problems — like a very urgent need for many qualified teachers in a very short time frame — it also provides us with the rare opportunity to investigate, how a modern curriculum is accepted by the teachers and the students. This also gives valuable insights into the preconceived notions teachers and students hold about computer science.

The outline is as follows: The following chapter 2 provides some related work, chapter 3 introduces the subject of Informatics in the Bavarian curriculum. Chapter 4 presents the relevant parts of our study, Chapter 5 the results of our analysis. Chapter 5 contains some topics of further research and a conclusion.

2. RELATED WORK

We will start the discussion with the presentation of related work concerning the teaching approach we have chosen, followed by publications that deal with research on teacher attitudes or behaviour of the students and finally the strategies the instructor uses. As shown by [21], "(1) only teachers with a formal CS background should teach CS in the high school; and (2) a general science-teaching certificate is not sufficient for teaching CS". This conclusion is supported by [6], where the problems with the diverse background of american CS teachers are described. Following the latter publication, a consistent level of certified knowledge is necessary for teachers in computer science to ensure proper teaching. Thus one of the most crucial predictors for the success of the students seems to be the quality of the teacher education.

In order to find out the teachers attitudes and opinions towards the topics of our curriculum, we had to additionally present some alternative topics that are not contained in it. We decided to arrange the topics in a structure similar to that presented in [22], where the attitudes of students towards CS are investigated. There, three different perceptions of computing are found: use, professional use and design. Additionally, [20] presents some initiatives to harmonise CS education.

3. THE SUBJECT OF INFORMATICS

The German school system is heavily federalised. Organisation, types of schools, subjects and curricula vary from state to state. However, every state has its own rather strict system of curricula. A typical curriculum will explicitly state the learning objectives (including suggested time frames), that students have to achieve in each grade.

The new subject of Informatics in Bavarian Gymnasiums is comprised of three stages:

- In grade 6 and 7 all students of the Gymnasium have to attend 1 compulsory lesson per week.
- In grade 9 and 10 there are 2 lessons per week, compulsory for all students that have chosen natural science/technology as their direction of study (typically about 50% of all students).
- In grade 11 and 12 the students that have attended Informatics in Grade 9/10 can choose the eligible Informatics course of 3 lessons per week.

The main ideas that drove the design of the curriculum in Informatics in Bavaria is covered by several publications (see e.g. [11], [12], [16]) already. The curriculum follows to a great extent our information-oriented teaching approach as presented in [1]. Basically the students should acquire three different basic competencies within this new subject:

- represent, structure, store, link and exchange information using suitable hardware and software combinations
- master, describe and communicate about complex systems using suitable — particularly object—oriented modelling techniques
- implement models using suitable software systems or programming platforms — particularly object—oriented programming languages.

Concerning the teaching methods, the subject of Informatics is encouraging, more than many other subjects, a very modern and student-oriented learning style: the intensive problem–oriented usage of computers forces the use of teamwork, group–teaching, project work and product–orientation in the lessons. During the design of the curriculum we had to cope with some serious problems. The hardest one to solve, was the didactical dilemma described in [13]: Modern constructivist teaching approaches postulate to pose authentic problems to the students that have relevance in the "real world" outside the school. Thus, it seems
advisable to start programming with interesting, sufficient complex tasks that convince the students, that the concepts they have to learn are really helpful for their later professional lives. Hence we have to start our programming course with quite complex programs that simulate processes that the students might encounter in their everyday life. On the other hand, if we start with quite complex object–oriented programs, we might ask too much of the students, because they will have to learn an enormous amount of new, partly very difficult concepts at one time. As pointed out in [14], we decided to solve this problem by following the “objects first” teaching strategy (see e.g. [10]) in a very radical way.

The basic idea was to start the course in grade 6 (where the students are 11 or 12 years old) with object modelling of standard software documents like vector and pixel graphics, texts or multimedia presentations or hypertext structures. This way, the students learn to use the object–oriented concepts object, attribute, class, method, association, aggregation and reference in order to manipulate documents, some years before they will have to apply them in the context of object–oriented programming. In grade 6 the students start working with objects of classes like Circle, Rectangle, Symbol, Paragraph etc. They find out, that some of the objects are connected by aggregations, which might even be recursive, as for example on the class Folder (within file systems). In grade 7 they learn to apply the concept of references (implemented as links) in order to construct hypertext structures and to exchange information using e–mail systems. At the end of grade 7, they learn to activate objects by programming their own methods, using simple robot systems (e.g. Karol or Lego Mindstorms). At the beginning of grade 9 the students apply the concept of functions by designing data flow diagrams, which are then implemented using spreadsheets. Following this, they construct object–oriented data models and implement them using relational databases. In grade 10, they finally start “real” object–oriented programming, and thereby have to apply all the concepts they have learned in the former grades. In the eligible course of grades 11 and 12, the students work with recursive data structures (lists, trees and graphs) and corresponding recursive algorithms. They apply the basic concepts of software engineering: software life cycle models and stages of software development. In grade 12, they design formal languages and learn how parallel processes can be synchronized. They simulate computer networks, analyze their topology and learn how to programme. In the second phase of working as a supervised teacher the students pass the second state examination. However, as Informatics was introduced as a new subject “from scratch”, there were several in service programs, that teachers could attend to get the teaching qualification for Informatics as an extra (third) subject. Mostly, teachers of mathematics or physics chose to attend those programs. Nevertheless, the need for qualified teachers in Informatics has not yet ended. So, particularly in the first two grades 6 and 7, many of the current teaching persons are not officially qualified.

4. DESIGN OF THE STUDY

The study, which was conducted in December of 2009, was designed to achieve three major goals. The first goal was to simply get first insights into how well the new subject is established. This included getting estimates about how many underqualified teachers are teaching the subject, finding out whether the computer equipment of the schools is appropriate or which software and proramming systems are used in the lectures. Secondly, we tried to find out how the subject of Informatics is perceived and accepted by teachers, students and their parents. The third goal, which this article will focus on, was to explore the attitudes and opinions, that teachers have about the subject and the items of the curriculum. Especially, we were trying to find out:

- How is the curriculum accepted by the teachers?
- In what respect (if at all) differ teachers who have a university degree in computer science from those who do not have such a degree?
- In what respect differ teachers that rate the curriculum positively from these that object against it?

The study was conducted as an online survey, using the open source tool “LimeSurvey”. We received 448 responses from teachers. We estimate, that this amounts to about a third of the relevant teachers. There are no official statistics about the number of teachers that are teaching Informatics without being qualified for the subject, so we can only estimate the total size of the population roughly. The survey was designed as a census (every teacher in Bavaria, who teaches Informatics, should have received an invitation by letter or by e–mail). So, given a response rate of well below 100% we will have to take into account a nonresponse bias in the results. Beforehand, we identified two major problems concerning non respondents:

1. Not every teacher (explicitly those, that are not officially qualified and part–time teachers) might have been informed about the survey.
2. Teachers that feel strongly (positive or negative) about the subject are more likely to take part than teachers that have no strong opinion.

Consequently, we have to treat the results to certain questions with caution, as we cannot be sure that the results are an unbiased random sample of the population. However, given the actual responses, we assume the bias to be negligible, even more so for the type of analysis that we’ve been doing here. The results of the exploratory part of the study is mainly based on the following two questions of the survey:
Table 1: Topics of the Bavarian curriculum in Informatics. The abbreviations are used later for presenting the results.

<table>
<thead>
<tr>
<th>Abbrev./Category</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>InfRep/OOM</td>
<td>Information and its representation using pictures, symbols, text or multimedia</td>
</tr>
<tr>
<td>Graph/OOM</td>
<td>Representing information in graphical documents — graphics software: object, attribute, method, class</td>
</tr>
<tr>
<td>Text/OOM</td>
<td>Representing information in text documents — word processors: classes Document, Paragraph, Symbol, aggregation</td>
</tr>
<tr>
<td>MultiMed/OOM</td>
<td>Representing information in multimedia documents — presentation software: class Slide</td>
</tr>
<tr>
<td>HierInf/OOM</td>
<td>Hierarchical information structures — file systems: classes File, Folder, recursive aggregations, tree structures</td>
</tr>
<tr>
<td>InfNet/OOM</td>
<td>Information networks — Hypertext: classes Hyperdocument, Link, Anchor, referencing by links, Hypertext structures</td>
</tr>
<tr>
<td>ExchInf/OOM</td>
<td>Exchanging information — e-mail: classes EMail, Attachment, sending and receiving e-mails using server structures</td>
</tr>
<tr>
<td>Algo/(OO)P</td>
<td>Describing processes using algorithms: control structures, programming methods of robot systems</td>
</tr>
<tr>
<td>FuncDat/OM</td>
<td>Functions and dataflow — spreadsheets: functions, dataflows, formula in spreadsheets</td>
</tr>
<tr>
<td>Datamod/OOM</td>
<td>Data-modelling, databases: object-oriented data modelling, relational database systems, selection, projection, join, key, redundancy</td>
</tr>
<tr>
<td>ObjProc/(OO)P</td>
<td>Objects and processes: object, class, attribute method, association, algorithms, state models</td>
</tr>
<tr>
<td>GenSpec/OOMP</td>
<td>Generalisation and specialization: inheritance, polymorphic methods, class trees</td>
</tr>
<tr>
<td>ProgProj/(OO)P</td>
<td>Programming project: cooperative software development</td>
</tr>
<tr>
<td>RecDat</td>
<td>Recursive data structures: list, tree, graph, recursive algorithms</td>
</tr>
<tr>
<td>SoftEng</td>
<td>Software engineering: life cycle modes, stages of the waterfall model, cost analysis, quality</td>
</tr>
<tr>
<td>FormLang</td>
<td>Formal languages: syntax, grammars, automata</td>
</tr>
<tr>
<td>ComSync</td>
<td>Communication and synchronization of processes: parallel processes, monitor concept, networks</td>
</tr>
<tr>
<td>HardArc</td>
<td>Hardware architecture: von Neumann architecture, register machine, assembler programming</td>
</tr>
<tr>
<td>LimComp</td>
<td>Limitations of computability: measuring performance of algorithms, principal limitations, halting problem, cryptography</td>
</tr>
</tbody>
</table>

Q1 Question How would you rate the following items of the curriculum regarding their importance for the future (professional) life of the students? Items see Table 1, restricted to grade 6 – 10

Response Items Ordinal scale with the 4 values: very important / important / rather unimportant / unimportant

Q2 Question How would you grade your own subject domain knowledge about the following items of the curriculum? (the items of Table 1 are used)

Response Items Ordinal scale according to the German grading system (1 through 6, 1 being the best). Since it’s a grading scale (that the teachers are used to), we will treat it as an interval scale where appropriate (for example by calculating average grades over responses).

We skipped the last two grades in Q1, because at the time of the survey, grade 11 had just started for the very first time after the introduction of the subject. So, at that point, no teacher had experience in teaching the subject in those grades and thus might not be familiar enough with the items of the curriculum to judge their importance (while the given items should be descriptive enough to be able to judge ones knowledge about the topic).

We analyzed the survey in a two step approach: Firstly, we use cluster analysis to identify typical “response patterns” to the above questions. Then, we used an automated process to test, for all other questions Qx, the hypothesis: “The responses of persons to Qx belonging to cluster i differ significantly from the responses of persons belonging to cluster j” for every pair (i, j) of clusters (treating the two clusters as two independent samples and using $\alpha = 0.09$ for all tests). This serves as a characterization of the persons belonging to a cluster. The questions that turned out to be the most interesting concerning this analysis are the following:

C1 Question According to your opinion, how do the students respond to the new subject of Informatics? (separately for the grades 6, 7, 9 and 10)

Response scale Ordinal – 5 values: they like it / they mostly like it / they don’t like it, but they don’t object / they mostly object / they object

C2 Question How content are you with the new subject Informatics, in general? (asked separately for grades 6, 7, 9 and 10)

Response scale Ordinal – 6 values: very content / mostly content / rather content / rather not content / mostly not content / not content

C3 Question If you had to design a curriculum for Informatics, would you include the following learning objectives? (see Table 2 for the items, asked separately for lower and higher grades (5 through 7 and 8 through 10).

Response scale Ordinal – 4 values: certainly / possibly / rather not / by no means

The list of learning objectives for C3 was comprised of the following group of elements (the order of the elements
was randomly chosen for every participant, and the origins of the learning objectives were kept secret):

- 6 elements of the Bavarian curriculum (BC), see above,
- 6 elements of a curriculum of Hamburg (HC), another federal state of Germany,
- 6 elements of the curriculum of the "European Computer Driving Licence (ECDL)")
- two additional items (Rest).

The first three categories were copied verbatim from the corresponding curricula, that we had chosen for the following reasons: The Bavarian curriculum stresses object oriented modelling and programming, while the Hamburg curriculum [18] puts the emphasis on practical aspects of using computers (however in an "academical" way, for example: "Being able to judge presentation software according to ergonomic aspects"). Finally, the ECDL ¹ has very clear objectives concerning the abilities to use computers in certain practical ways, like: "To understand what software is and to know the names of standard programmes". The two remaining topics were selected in order to represent pure technical or psychomotor skills. Following [22], the Bavarian curriculum represents the design perception, while the Hamburg topics mainly belong to the professional use. Finally the ECDL and the Rest topics focus on the pure use perception.

5. RESULTS AND DISCUSSION

For the cluster analysis, the responses to Q1 and Q2 were encoded as integer numbers, leading to a data matrix with 13 and 19 columns respectively and about 220 rows (respondents who haven’t given a complete answer to one of the questions have been removed for the clustering process). The clustering was done in GNU R using the Mclust package [9], which finds the best (according to the BIC measure) of several types of Gaussian Mixture Models and also identifies the optimal number of clusters (up to a maximal number that is specified manually). As clustering is a probabilistic process, the output consists of a map, giving the probability of row \( i \) belonging to cluster \( j \). For further analysis, we assign every response to the cluster that contains it with the highest probability. Additionally, for each cluster, the mean of each column of the matrix is computed which can be used to plot the "typical" answers of each clusters.

The results of the analysis of the answers to Q1 can be seen in Figure 1. Since the scale for Q1 is only an ordinal one, we cannot take the absolute positions of the means into account. We can, however, observe trends. Initially, we will discuss the results over all clusters. Then, most notably, there is a clear distinction between the first two grades 6 and 7 and the higher grades 9 and 10. All clusters tend to rate the topics of the higher grades 9, 10 less important than those of grades 6, 7, thus supporting the object first strategy of the curriculum. Students that don’t choose the natural science branch will only learn about modelling and skip the (object-oriented) programming in the higher grades. So, it is interesting to observe, that those learning objectives are considered important by most teachers beyond being mere prerequisite knowledge for later programming activities. So it seems like there is a benefit to be gained by learning about

<table>
<thead>
<tr>
<th>Source</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>BC1</td>
<td>Know basic elements of the object-oriented modelling of information systems</td>
</tr>
<tr>
<td>BC2</td>
<td>Use hierarchical structures to organize information</td>
</tr>
<tr>
<td>BC3</td>
<td>Know the functional view as a general approach to the way spreadsheets work</td>
</tr>
<tr>
<td>BC4</td>
<td>Being able to model a moderately sized amount of data with models and associations</td>
</tr>
<tr>
<td>BC5</td>
<td>Being able to structure a temporal sequence using states and transitions</td>
</tr>
<tr>
<td>BC6</td>
<td>Being able to translate moderately sized algorithms into programs</td>
</tr>
<tr>
<td>HC1</td>
<td>Know that the data privacy law is a personal right</td>
</tr>
<tr>
<td>HC2</td>
<td>Know how to use search machines and catalogs</td>
</tr>
<tr>
<td>HC3</td>
<td>Know and use important structuring- and styling-rules when writing texts</td>
</tr>
<tr>
<td>HC5</td>
<td>Digitalize an image using a digital camera or a scanner</td>
</tr>
<tr>
<td>HC6</td>
<td>Being able to judge presentation software according to ergonomic aspects</td>
</tr>
<tr>
<td>ECDL1</td>
<td>Being able to use simple word-processing and printing capabilities of the operating system</td>
</tr>
<tr>
<td>ECDL2</td>
<td>Know how viruses may enter a computer</td>
</tr>
<tr>
<td>ECDL3</td>
<td>Know how operating systems organize drives, directories and files in a hierarchical structure</td>
</tr>
<tr>
<td>ECDL4</td>
<td>To understand what software is and to know the names of standard programmes</td>
</tr>
<tr>
<td>ECDL5</td>
<td>Being able to create and merge paragraphs in a word-processing software</td>
</tr>
<tr>
<td>ECDL6</td>
<td>Understand, that associations in databases are mainly for avoiding duplicates</td>
</tr>
<tr>
<td>Rest1</td>
<td>Being able to build a PC from its components</td>
</tr>
<tr>
<td>Rest2</td>
<td>Being able to use touch typing</td>
</tr>
</tbody>
</table>

Table 2: List of learning objectives used in C6.

¹see www.ecdl.com
object-oriented modelling without programming — this aspect might be considered when designing curricula in computer science for non computer scientists (be it in secondary schools or in subjects in university). As a matter of fact, the only two items of the curriculum which have a median of 1 (very important) — when taking all responses into account, are Graph and Text, which can be clearly attributed to object-oriented modelling. Concerning the subject domain knowledge of the teachers (Q2), we found that all tend to rate their knowledge in the higher grades (starting from grade 10) not as profound as for the lower grades 6 to 9. However, the fact, that the teachers hadn’t been yet actively teaching grades 11 and 12 might have influenced the answers to this question. Concerning preconceived notions about typical subjects of computer science education in secondary schools, our results show that teachers accept the information-oriented approach to computer science [1]. To support this hypotheses, we can use the results of C3: The learning objectives that have a median of certainly (considering all responses) for the first grades 6, 7 are:

1. (BC2) Use hierarchical structures to organize information
2. (HC2) Know how to use search machines and catalogs
3. (ECDL5) Being able to create and merge paragraphs in a text processing software

For the higher grades 9 and 10, the list of objectives with a median of certainly consists of:

1. (BC1) Know basic elements of the object-oriented modelling of information systems
2. (BC3) Know the functional view as a general approach to the way spreadsheets work
3. (BC4) Being able to model a moderately sized amount of data with models and associations
4. (BC6) Being able to translate moderately sized algorithms into programs
5. (ECDL6) Understand, that associations in databases are mainly for avoiding duplicates

This list portrays a curriculum, that approaches computer science mainly from a modelling and information based approach, thus also supporting the hypothesis, that object-oriented modelling is generally quite well accepted by the teachers. However we should also note, that some of the chosen learning objectives clearly deal with typical user skills — at least for the first grades.

Next, we analyze how the clusters differ from the general results. We refer to the clusters as 1 through 4 following Figure 1. The clusters have roughly equal sizes, from 1 to 4 containing 25%, 44%, 13% and 18% of the responses. Our analysis leads to the following characterization of the 4 clusters of Q1, when considering the questions presented above and, particularly, Q2:

Cluster 1 (“office users”) Emphasizes the object-oriented approach to standard software, and attributes least importance to algorithms and programming. The teachers in this cluster don’t feel as confident in their expert knowledge of the topics as the other teachers. This holds particularly for the higher grades. The average response (taken from the answers of C2) for the topics of grades 10, 11 and 12 is a mere 3.1 (the other clusters range from 2.2 to 2.6 for those topics). Taking all grades into account, the mean is 2.2 (others range from 1.6 to 1.7).

Cluster 2 (“fans of the curriculum”) Rates the curriculum very positively. Most importance is given to object-oriented modelling, least important are AlgO, ObjProc and GenSpec — mostly dealing with OOP. In the teachers’ opinion, the students think mostly positive about

---

Figure 1: Results of clustering the answers of Q1. The mean of every cluster is plotted by the numbers identifying a cluster. Additionally, the category of each item of the curriculum is given, where appropriate. The vertical lines separate the grades from each other.
Figure 2: Results of clustering the answers of Q2. The mean of the clusters is plotted. Additionally, the dotted line shows the mean taken over all responses. The vertical lines separate the grades from each other.

the subject (the medians for C1 are they like it for grades 6 and 7, they mostly like it for grade 9 and they don’t like it, but they don’t object for grade 10. Additionally, they are very content with the new subject. The median for C2 are for grades 6, 7, 9 and 10: mostly content, very content, mostly content, mostly content. They separate themselves from other clusters especially by their opinion regarding the grades 6 and 7.

Cluster 3 (“anti-programmers”) Clearly emphasizes the importance of OOM over (OO)P. Concerning the other questions, the teachers are somewhere between those in clusters 1 and 2. They differ from cluster 1 in a significantly better rating of their subject domain knowledge of the higher grades (the mean is 2.6 for grades 10, 11 and 12) and they differ from cluster 2 mostly in being not as content with the new subject in general (for grades 6, 7 and 10). Interestingly enough, their responses to Q1 also lie somewhere between those of clusters 1 and 2.

Cluster 4 (“traditional computer scientists”) Relative to other clusters, cluster 4 puts most emphasis on the more mathematically–oriented functional modelling, algorithmic structures and object–oriented programming (Algo, FuncDat, DataMod, ObjProc). Although, the typical response pattern of persons belonging to cluster 4 is rather special, there seems to be no variable that separates them from all other clusters. However, concerning C1 and C2, there are differences between cluster 4 and every other cluster albeit for different grades. However, in all cases, the persons belonging to cluster 4 are less content and think less positively about the students’ opinion towards the subject than the other clusters.

So, generally speaking, there are two main groups of teachers concerning the relative appraisal of OOM vs. OOP. Cluster 4 is the only cluster that doesn’t like our approach to the curriculum, putting more emphasis on a traditional view on computer science including programming, algorithms and functions. Interestingly enough, those teachers not only tend to rate the students’ opinion worse, they also are less content with the students’ engagement in the lessons. Cluster 2 on the other hand puts more emphasis on modelling than programming and is generally content with the subject and the students. Still, both cluster 2 and cluster 4 are generally rating the items of the curriculum important. Cluster 2 is the only cluster, that rates the importance of every item of the curriculum at or above important, while cluster 1 rates not a single item this important. For all items in the 10th grade (the 3 rightmost in Figure 1), the difference between the means of cluster 1 and cluster 2 is even strictly greater than 1.

Additionally, the following trends can be observed:

- The importance of algorithmic structures (Algo) is, up to that item, considered the least important by all clusters except cluster 4, which rates it among the two most important items up to that point. However, as almost all teachers rate their knowledge of this topic rather well, this cannot be a consequence of the teachers feeling insecure.

- The concepts of generalisation and specialisation are considered rather unimportant by all clusters — even less than the “real world” programming project. This is interesting, as it is one of the central concepts of both OOM and OOP. This could indicate, that, as it is introduced very late in the curriculum, it is not seen as a modelling technique by the teachers and thus rated only in the programming context.

- Most teachers rate their knowledge in software engineering (SoftEng) lower than any other item of the curriculum. This is to be expected, however, as teachers normally never gain any work experience as a com-
puter scientist (and a large fraction of the teachers currently teaching Informatics have studied mathematics or physics as another subject). Still, it indicates, that teachers might need more experience in some of the topics that they have to present to the students.

Figure 2 shows the results, when clustering the answers to question Q2. Further analysis is needed, before a thorough analysis to this question can be presented. However, at the moment, we can see some interesting developments nevertheless. Clusters B and A are relatively large, so, unsurprisingly, the mean over all responses follows very closely the pattern of those clusters. All clusters tend to rate their subject domain knowledge worse, for the higher classes, which is to be expected to some extent, of course. Cluster C will most probably contain underqualified teachers, that have to teach Informatics in the lower classes 6 and 7 only. In the long run, when the subject has been firmly established in the curriculum, we would hope for most teachers showing a pattern similar to cluster D. Secondary school teachers with a university degree in computer science (the usual way in Germany) should rate their subject domain knowledge concerning the curriculum with a (very) good grade. This, in turn, means, that for the time being, additional in-service training is required, to help the current population of teachers with their problems concerning the subject. How this can be done efficiently for a large number of teachers is another interesting topic of research in the field of didactics.

6. CONCLUSION AND FUTURE WORK

Taking the results of our analysis into account, we can draw the following conclusion: Although the idea of starting an introductory informatics course with pure object-oriented modelling is accepted by the teachers, the importance of using this knowledge in the context of object-oriented programming is not as well accepted, and teachers don’t feel as secure with the complex task of OOP — programming remains difficult. As we had already anticipated this result, we offer a well-balanced set of programming courses in our in-service training program for Informatics teachers: the introductory courses Basics 1 and Basics 2 that deal with simple aspects of OOP are followed by two more advanced courses that cover programming of graphical user interfaces and the connection of Java programs to relational data base systems like MySQL respectively. These courses will be attended by about 120 teachers in the years of 2009 and 2010. Concerning our research program, we will proceed from asking the teachers towards evaluating the motivation and the connection of Java programs to relational data base courses that cover programming of graphical user interfaces and the connection of Java programs to relational data base courses that deal with simple aspects of OOP are followed by two more advanced courses that cover programming of graphical user interfaces and the connection of Java programs to relational data base systems like MySQL respectively. These courses will be attended by about 120 teachers in the years of 2009 and 2010. Concerning our research program, we will proceed from asking the teachers towards evaluating the motivation and the performance of the students. To that end we work on the definition of competencies that we might test in the future. Conclusively, in an overall view, we are quite content with the performance of the new subject Informatics in Bavaria, so far.

7. REFERENCES


