From exercise characteristics to competence dimensions – exemplified by theoretical computer science in secondary education

Kirsten Schlüter, kirsten.schlueter@cs.fau.de
(University of Erlangen-Nuremberg, Didactics of Informatics
Martensstrasse 3, D-91058 Erlangen, Germany)

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

Abstract
Competence models as point of origin for educational standards are the foundation of assessment and evaluation in view of educational goals and thereby provide major information for revision and improvement of instruction. Computer science education is a young discipline, compared to, e.g., the core subjects analyzed in the PISA studies (OECD, 2006), mathematics, reading and science. The approaches to competence modelling, that are published up to now, are plausible but of pragmatic nature. This paper aims to introduce an empirical statistical way of competence modelling in secondary education, focusing the dimensional structure of competence in theoretical computer science. Theoretical computer science appears to be an adequate example because it is an integral and commonly approved part of computer science, on the one hand, and educational standards are notably important as a framework to help teachers, parents and students to orientate within the field of highly abstract contents, on the other hand. Starting from exercise characteristics, gained by regarding neighbourd sciences, especially mathematics, consulting experts, observing students and analyzing exercise difficulty, the dimensional structure of a competence model is to develop by means of empirical statistical techniques, such as factor analysis. Further statistical analyses, such as clustering competence profiles in terms of the characteristics, are to perform to elaborate typical competence profiles for the purpose of facilitating individual assessment in secondary schools. Therefore a competence model will be theoretically founded and empirically verified and a measuring instrument will be conceptually designed.

Keywords
Didactics of informatics, secondary computer science education, educational standards, competence model, theoretical computer science, characteristics of exercise difficulty, competence profiles

INTRODUCTION
Theoretical Computer Science in Secondary Education
Theoretical computer science, including formal languages and automata theory, the theory of computability and computational complexity, is an integral part of secondary computer science education. The subject is anchored for example in the ACM curriculum recommendations (ACM, 2003) that are of international impact on the implementation of secondary computer science education. Also in national curricula, for instance in the recommendations of the German Informatics society (GI) for educational standards (GI, 2008), main issues of theoretical computer science are of significance. Although there is general consent on the importance of theoretical computer science, as it provides fundamental concepts and imparts basic compe-
tences, presumably, in practice it is oftentimes disregarded, perhaps because it is difficult to learn and to teach, or perhaps because it is outranked by the apparent personal benefit of skill training.

Educational standards
On the one hand to fix binding competence goals and on the other hand to provide orientation in complex fields like theoretical computer science, educational standards are demanded (e.g. Schubert, 2005). For computer science, educational standards, especially competence modelling (e.g. Magenheim, 2005) are in their very beginning in Germany, but currently focused by expanded and intensive studies. The recently presented concept of German Principles and Standards for Computer Science in Secondary Education (GI, 2008) implicates a competence model that is deduced from the superior educational goal of preparing children for their responsible participation in the information society: children shall understand the principles and functionality of information systems so as to be enabled to reflect on interconnections between information technology, man, and society. They shall learn how information can adequately be represented by data and vice versa how to interpret data, how to utilize algorithms in problem solving and how to model and implement aspects of information processing by use of formal languages and automata. Beyond skilled application, critically reasoning and evaluating the employment of information technology enables them to structure and interrelate their computer science knowledge with other school subjects. Finally they shall learn to communicate using technical terms in order to cooperate and share their knowledge. The model captures these competences along two interleaving dimensions, the content and the process band, with the content and the process competences expressed by nouns and verbs, respectively (see above). As it is developed in a pragmatically way, the concept is very plausible but not yet empirically verified.

Measurability of competences
The abovementioned conception of educational standards of informatics is marking a consensus of hundreds of German didactists and teachers of informatics on what competences each learner of each German school type should have reached by the end of the 7th and 10th grade, respectively. However it does not say how to measure these competences. Exercises play an important role in this process. The exercise in figure 1 is related to theoretical computer science on secondary education level.

<table>
<thead>
<tr>
<th>User names in computer networks usually follow syntactical rules, for example: A valid user name consists of a sequence of one upper case letter followed by one or more lower case letters. The sequence may be repeated. “CatWeazle” or “EllyBeinhorn” are valid user names. Draw the state diagram of a finite automaton for user name verification.</th>
</tr>
</thead>
</table>

Figure 1: Secondary education level exercise

It is aimed at testing the ability to analyze the syntactical specification of, e.g., user names in computer networks and transfer it into a different representation, e.g., a state diagram. Whereas students may give a variety of answers -- they may specify the diagram all correct, they may confuse states or they may omit certain transitions -- conclusions apart from only “correct” or “incorrect” should be drawn. A competence scheme based on the exercise characteristics would be helpful to have when evaluating and interpreting students’ answers.
In the paper at hand the authors introduce a new methodology for competence modelling in the field of computer science based on exercise characteristics. Their conception is illustrated by theoretical computer science at secondary school level. Similar work has to be done for other important topics of secondary level computer science.

AN APPROACH TO COMPETENCE MODELING

Exercise Difficulty
In order to be empirically verified, competence has to be operationalized. According to Weinert competence is conceived as “a roughly specialized system of abilities, proficiencies, or individual dispositions to learn something successfully, to do something successfully, or to reach a specific goal.” (Weinert, 2001) Exercises afford access to competence operationalization and evaluation, as Klieme states in the German Expertise on the Development of National Educational Standards. “Translated into tests and test items, such competency models make it possible to describe the performance level of students in a criterion-referenced manner: not through comparison with the performance levels of others, but with reference to specifically defined criteria. This kind of criterion-referenced description of competence thus identifies concrete requirements that should be mastered by students at a specific level of competence, and also describes activities and subject matter that have not yet been mastered or attained.” (Klieme et al, 2003) Postulating that a student is the more competent the more difficult exercises he or she masters, the question arises: what are the exercise features, e.g., content, context, complexity, accounting for the exercise difficulty?

Some examples shall be presented to point out what makes the difference in exercise difficulty. The exercises are related to the content area of formal languages and automata theory, central curriculum issues in secondary computer science education. The examples deal with palindromes, because palindromes are both intuitively to understand and easily to define in terms of formal languages. Since the purpose is to illustrate how difficulty grows, sometimes the exercises may tend to rise above average secondary level.

Starting with the specification of palindromes, the left explanation is redundant, as some examples of palindrome words and sentences are added. If the redundancy is eliminated on the right, the explanation is supposedly more difficult to understand.

<table>
<thead>
<tr>
<th>You know what a palindrome is?</th>
<th>A palindrome is reading the same in either direction (like “otto” or “never odd or even”).</th>
</tr>
</thead>
<tbody>
<tr>
<td>You know what a palindrome is?</td>
<td>A palindrome is reading the same in either direction.</td>
</tr>
</tbody>
</table>

Figure 2: Exercise attributes: redundancy

Apart from being redundant, the specification above is a popular description, while below, the right one is formalized. The specification might be the more difficult to understand the higher the level of formalization is.

<table>
<thead>
<tr>
<th>You know what a palindrome is?</th>
<th>A palindrome is reading the same in either direction (like “otto” or “never odd or even”).</th>
</tr>
</thead>
<tbody>
<tr>
<td>You know what a palindrome is?</td>
<td>The set of palindromes is {w</td>
</tr>
</tbody>
</table>

Figure 3: Exercise attributes: level of formalization

Now regarding a simple task, the one on the left concerns palindromes, a well
known play on words, while the one on the right concerns prefix-free codes, far from the student’s *realm of experience* and therefore more difficult to understand.

<table>
<thead>
<tr>
<th>You know what a palindrome is?</th>
<th>A prefix-free code is one in which no code-word is a prefix in another (like “exam” is a prefix in “example”).</th>
</tr>
</thead>
<tbody>
<tr>
<td>A palindrome is reading the same in either direction (like “otto” or “never odd or even”).</td>
<td>Check up the palindrome(s).</td>
</tr>
<tr>
<td>Check up the palindrome(s).</td>
<td>Check up the prefix-free code(s).</td>
</tr>
<tr>
<td>□ 42311234</td>
<td>□ The set of all German words.</td>
</tr>
<tr>
<td>□ an eye for an eye</td>
<td>□ The set of all German postcodes.</td>
</tr>
<tr>
<td>□ madam I’m adam</td>
<td>□ The code with codewords 0, 10, 11.</td>
</tr>
</tbody>
</table>

![Figure 4: Exercise attributes: realm of experience](image)

In the above task, the educational objective according to Anderson’s taxonomy (Anderson et al, 2001) is *understanding*. It appears that the task difficulty grows with the *level of cognitive process*: the leftmost item below is level (1), *remember*, the one in the middle is level (3), *apply* and the rightmost is level (6), *create*.

<table>
<thead>
<tr>
<th>For simplification we mark the center of a palindrome (like “ot#to”, “never od#d or even”).</th>
<th>Give the definition of deterministic push down automata.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explain how the following push down automaton recognizes palindromes.</td>
<td>Specify a push down automaton to recognize palindromes.</td>
</tr>
</tbody>
</table>

![Figure 5: Exercise attributes: level of cognitive process](image)

Another characteristic accounting for the task difficulty is the *level of abstraction*. Construction of an automaton, see below on the left, is a *concrete* task, while reasoning, see below on the right, is an *abstract* task, and supposedly more difficult.

<table>
<thead>
<tr>
<th>For simplification we mark the center of a palindrome (like “ot#to”, “never od#d or even”).</th>
<th>Specify the adequate automaton to recognize palindromes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Give reasons that the language of palindromes is not regular.</td>
<td>Describe the set of words recognized by the automaton.</td>
</tr>
</tbody>
</table>

![Figure 6: Exercise attributes: level of abstraction](image)

Staying with automata theory, the following example shows how difficulty increases with *complexity*. The automaton with two states is less *complex* than the automaton with four states, which might be way more difficult to analyze.

<table>
<thead>
<tr>
<th>Describe the set of words recognized by the automaton.</th>
<th>Describe the set of words recognized by the automaton.</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
</tr>
</tbody>
</table>

![Figure 7: Exercise attributes: complexity](image)

**Theoretical Foundation: Characteristics and Categories**
The exercise attributes accounting for the difficulty shall be called “characteristics”. Altering one attribute, e. g., from less to more distinct, while keeping the other attributes unchanged, will make an exercise more or less difficult, respectively. To acquire a starting set of characteristics, some valuable sources are regarded:
The PISA 2006 framework of mathematical literacy, as mathematics and theoretical computer science share some calculi (e.g. logic), and mindsets (e.g. defining or proving by induction) and therefore require some coinciding competences. “The major components of the mathematics framework, consistent with the other PISA frameworks, include contexts for the use of mathematics, mathematical content and mathematical processes, each of which flows directly out of the literacy definition. The discussions of context and content emphasise features of the problems that confront students as citizens, while the discussions of processes emphasise the competencies that students bring to bear to solve those problems. These competencies have been grouped into three competency clusters [annotation of the authors: reproduction, connections, reflection] to facilitate a rational treatment of the way complex cognitive processes are addressed within a structured assessment program.” (OECD, 2006)

- Consulting teachers of computer science in secondary school or university education as experts.
- Observing students working on theoretical computer science problems in class, in written tests, or in contests.
- A classification of exercises in theoretical computer science extracted from schoolbooks and computer science contests addressing secondary schools (for a classification of exercises in object-oriented modelling see (Brinda, 2004)).

First reflections gave rise to the following selection of characteristics that appear to be relevant variables of the exercise difficulty in case of theoretical computer science in secondary education. Selection criteria can be for example if the attributes are directly observable, and as simple and atomic as possible, to facilitate further processing and to minimize overlap.

I. Content
- The content area spans formal languages, automata theory, theory of computability and computational complexity.
- The closeness to the student’s realm of experience is high, if a problem is directly concerning students’ realm of experience, medium, if concerning observable phenomena at least, or low, if a task is beyond students’ realm of experience.
- The level of abstraction ranges from concrete, e.g., recognizing syntactically correct e-mail addresses, to abstract, e.g., proving that regular languages are closed under intersection and union.
- The complexity depends for example on the number of states of an automaton to be constructed or on the Chomsky hierarchy level of a language to be analyzed.

II. Task Setting
- The level of formalization ranges from textual phrases like “all palindromes” to mathematically formalized phrases like “\{w \mid w = uu^R\}”.
- As the task setting can be for example textually or graphically represented, redundancy ranges from plain textual task setting to redundant task setting, by additionally providing information, for instance a diagram.

III. Learner Activity
- The level of requirement as described in the German Uniform Examination Requirements of Secondary Informatics Education (CMEC, 2004) can be reproduction, application, or problem solving.
- The process band as suggested in the German Principles and Standards for Computer Science in Secondary Education (Puhlmann et al, 2007) (see introduc-
tion) spans (A) model and implement, (B) reason and evaluate, (C) structure and interrelate, (D) communicate and cooperate, (E) represent and interpret.

- Anderson’s revision of Bloom’s taxonomy of educational objectives (Anderson et al., 2001) provides two variables, the type of knowledge, which can be factual, conceptual, procedural or metacognitive, and the cognitive process, which can be (1) remember, (2) understand, (3) apply, (4) analyze, (5) evaluate or (6) create.

This initial approach to competence modelling via exercise characteristics gives insight into the facets of difficulty, but contains too many variables yet and offers far too sparse information on the interplay of variables. A first categorization in order to structure the set of variables roughly results in three categories: content, task setting and learner activity. The characteristics content band, closeness to the student’s realm of experience, level of abstraction and complexity together describe the difficulty arising from the exercise content. The level of formalization and the redundancy denote the difficulty arising from the way of task setting, independent of the content. The characteristics level of requirement, process band, type of knowledge and cognitive process describe the difficulty arising from the kind of required learner activity, again independent of content and task setting. At that point the theoretical foundation of the competence model to develop, the identification and categorization of relevant characteristics gained from, e.g., exercise analysis and classification, results in hypotheses on the three categories to be candidates for orthogonal dimensions underlying the multivariate model. The very next step will be gathering empirical data to verify these hypotheses and to establish a competence model from the variables that is as near as possible to reality but – this may be conflicting – as clear and comprehensible as possible.

Example Exercise

The following example exercise (see figure 8) with a supplement for support (see figure 9) shows how empirical data along the characteristics could be gathered, for instance in a classroom test.

User names in computer networks usually follow syntactical rules, for example:

A valid user name consists of a sequence of one upper case letter followed by one or more lower case letters. The sequence may be repeated.

In terms of regular expressions, the set of valid user names is specified by:

\((A+\ldots+Z)(a+\ldots+z)(a+\ldots+z)^*((A+\ldots+Z)(a+\ldots+z)(a+\ldots+z)^*)^*\)

“CatWeazle” or “EllyBeinhorn” are valid user names.

(A) Draw the state diagram of a finite automaton for user name verification. (6)

Figure 8: Example exercise

The example exercise (see figure 8) deals with finite automata, concerns a problem at least tangent to student’s realm of experience, and is not too complex. The task setting is redundant, because the syntax is given both in textual phrases and formal specification, plus valid examples. The exercise focuses the process competence (A), model and implement, on level (6), create, according to Anderson’s taxonomy. Task accomplishment is used as indicator for competence achievement: a student who answers item (A) correctly is regarded as competent on level (6), create, in the capability modelling and implementing.

The exercise supplement (see figure 9) even allows refining the information gained from task processing. Two levels of support are provided for students who miss solving problem (A) without help. Support i. is classified level (4), analyze, and sup-
Port ii. is classified level (3), apply. The level of support, students take to accomplish the task, can be used as additional indicator of the taxonomic level. If a student who is not capable of solving item (A) takes support i. and succeeds he or she is regarded as competent on level (4), analyze. If he or she takes support on level ii. and answers correctly he or she is considered to be competent at least on level (3), apply, in the capability modelling and implementing. In this way, different levels of competence along one characteristic can be tested highly efficient within a single task.

(A) i. Check the automaton that accepts user names exactly according to the above specified syntax. (4)

(State diagrams of finite automata are set up, including the correct one below. For the incorrect automata, the transition for more than one lower case letter or the transition for sequence repetition may be omitted.)

(A) ii. The diagram specifies the automaton for user name verification.

Give the series of states passed through when processing the user name “BeNeDict”. (3)

Empirical Statistical Verification: Competence Dimensions
To approve the theoretically founded hypotheses on the dimensional structure of a competence model, statistical techniques to analyze the empirical data are adopted. The data obtained from the empirical study can be visualized as n-dimensional vector space spanned by n test persons (see figure 10).

Figure 9: Exercise supplement

Figure 10: Factor analysis of the characteristics
Each characteristic, for instance the complexity of the problem, is represented by a vector, where the nth vector component is given by the grade of competence the nth test person achieved concerning the complexity of the problem. Obviously, there are a great many of variables, yet too many and potentially correlated. Factor analysis is a statistical technique for analyzing the correlations between a large number of variables describing a complex structure, in order to reduce them to a smaller number of underlying dimensions, called factors, and to determine how much of the variance in each of the original variables is explained by each factor. The factor analysis of the empirical data may result in the extraction of a handful, maybe three factors that need to be interpreted. Since a factor shows large correlation with the characteristic level of formalization and medium correlation with the characteristic redundancy of the task setting, it may be interpreted as their common factor, describing the portion of task difficulty emerging from the task setting. In this way the complexity of the model is reduced to a manageable number of possibly three factors representing the model dimensions (see figure 10).

The factors are observable and measurable via the characteristics. In order to validate the hypotheses on the dimensional structure of the competence model the factor analysis is supposed to result in a number of factors consistent with the number of assumed dimensions. Each bundle of variables statistically determined is supposed to correspond to one of the theoretical categories of characteristics.

Towards a Measuring Instrument: Competence Profiles
To facilitate individual assessment in secondary schools it is proposed to classify the occurring competence vectors, again by means of statistical methods. Cluster analysis classifies objects in a population by seeking to identify a set of homogeneous groups of objects, clusters, which both minimize within-group variation and maximize between-group variation. Here, the clusters of individual competence vectors confined by cluster analysis are interpreted as typical competence profiles. So individual assessment, whether for the purpose of motivation, examination or evaluation, at first results in a competence vector with each component denoting the level of proficiency in one competence dimension. Now the competence vector can be associated with a cluster and therefore with a corresponding competence profile. Text modules that describe the typical competence profiles in terms of the characteristics provide a diagnosis that is both meaningful and comprehensible for teachers, students and parents, as demanded by Klieme. “Only through these competency models do educational standards acquire the power to provide a point of orientation for teaching – they demonstrate the developmental levels and stages of subject specific competencies in clear, immediately comprehensible terms.” (Klieme et al, 2003)

A vision of concrete competence profiles is illustrated in figure 11. The profiles refer to four dimensions and characteristics, respectively, process band, content, task setting and cognitive taxonomy. Profile A characterizes a minimum competence level that should be fulfilled by every student, whereas profile B characterizes a maximum competence level that is supposed to be attained only by high performing students.

OUTLOOK

The set of thoroughly selected characteristics together with a categorization according to adequate criteria gained from regarding neighbouring sciences, consulting experts, observing students, and the classification of exercises in schoolbooks and computer science contests, is an excellent theoretical foundation for the generation of hypotheses on dimensionality of a competence model.
As demonstrated in the example exercise, appropriate empirical data can be collected for example in classroom tests. There are powerful statistical techniques to verify the model by analyzing the correlations of the empirical variables and seeking to calculate the underlying factors. Factors are interpreted as dimensions that span the model and explain the variance of the variables. Further statistical techniques can be used to refine the model by identifying competence clusters, interpreted as competence profiles, and conceptually design a measuring instrument.

<table>
<thead>
<tr>
<th>Dimension/Characteristic</th>
<th>Competence Profile A</th>
<th>Competence Profile B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Band</td>
<td>The learners model</td>
<td>The learners model</td>
</tr>
<tr>
<td>Content</td>
<td>a course of action close to their realm of experience (e.g. traffic lights) by using automata.</td>
<td>a problem beyond their experience (e.g. syntax check of arithmetic expressions) by using automata.</td>
</tr>
<tr>
<td>Task Setting</td>
<td>If textually specified,</td>
<td>Even if formally specified,</td>
</tr>
<tr>
<td>Cognitive Taxonomy</td>
<td>algorithms are recalled and properly applied.</td>
<td>problems are analyzed and appropriate algorithms are created.</td>
</tr>
</tbody>
</table>

Figure 11: Competence profiles, see Schlueter & Brinda, 2007

Further work will pursue two objectives, elaborating the model and anchoring the model in secondary computer science education. Up to now only one of the two central aspects of competence modelling has been dealt with. The proceeding described above is adequate to determine the dimensions of competence but the grades of competences are chosen by plausibility. To elaborate the model, the grades have to be constructed from theoretical considerations and empirically verified by means of statistical analyses. To anchor the model in secondary computer science education in general, although it is constructed decidedly along theoretical computer science, and thereby get hints for further improvement, the model should be compared against existing models.

REFERENCES


Biographies

Kirsten Schlüeter is a scientist in the “Didactics of Informatics” group at the University of Erlangen-Nuremberg (Germany), with a research focus on competence models in secondary computer science education. She graduated in computer science 1992 at the University of Bonn, worked as scientist in statistics and geography, as teacher of mathematics and informatics and complementarily in a committee developing tasks for a national informatics contest.

Torsten Brinda studied Computer Science at the University of Dortmund (Germany) from 1992 to 1998. From 1998 to 2005 he worked as a scientist in the “Didactics of Informatics” groups at the Universities of Dortmund (until 2002) and Siegen (2002 to 2005), where he 2004 finished his dissertation. In 2005 he became a professor for “Didactics of Informatics” at the University of Erlangen-Nuremberg. Didactic systems and educational standards of informatics are his current research interests.

Copyright Statement

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs2.5 License. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/2.5/ or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.