Modelling and Measurement of Competencies in Computer Science Education

Johannes Magenheim[1]**,Sigrid Schubert**[2]**, Niclas Schaper**[3]
[1]Didactics of Informatics
University of Paderborn
jsm@upb.de
[2]Didactics of Informatics and E-Learning
University of Siegen
sigrid.schubert@uni-siegen.de
[3]Organizational Psychology
University of Paderborn
niclas.schaper@uni-paderborn.de

Abstract: As a result of the Bologna reform of educational systems in Europe the outcome orientation of learning processes, competence-oriented descriptions of the curricula and competence-oriented assessment procedures became standard also in Computer Science Education (CSE). The following keynote addresses important issues of shaping a CSE competence model especially in the area of informatics system comprehension and object-oriented modelling. Objectives and research methodology of the project MoKoM (Modelling and Measurement of Competences in CSE) are explained. Firstly, the CSE competence model was derived based on theoretical concepts and then secondly the model was empirically examined and refined using expert interviews. Furthermore, the paper depicts the development and examination of a competence measurement instrument, which was derived from the competence model. Therefore, the instrument was applied to a large sample of students at the gymnasium's upper class level. Subsequently, efforts to develop a competence level model, based on the retrieved empirical results and on expert ratings are presented. Finally, further demands on research on competence modelling in CSE will be outlined.

Keywords: Competence Modelling, Competence Measurement, Informatics System Application, Informatics System Comprehension, Informatics Modelling, Secondary Education

1 Motivation

The paradigm-shift to a learnercentred and an outcome-oriented view on learning processes has been influenced by discussions and ongoing research in different areas of education. Besides results of research according to constructivist and cognitive learning theories, the discussion on learning taxonomies and competencies were crucial for the design and evaluation of learning processes. The shaping of domain-specific competence models with regard to their internal structure and different competence levels basically served two main goals: They are used to define educational standards and thereby contribute to the development of curricula and they enable the measurement of competences and learning outcomes in diverse educational settings.

Especially as a result of the Bologna Process and the OECD Program for International Student Assessment (PISA) the development and assessment of educational standards became a high level objective in the educational system (Adams, 2002). In Europe standards for the major school subjects, like mathematics, natural sciences, and the first language were developed for different levels of education. In Computer Science Education (CSE) the development of educational standards is not as advanced as in those main school subjects. On an international level there are some standard-oriented curricula of CSE like the Model Curriculum for K-12 Computer Science published by the IEEE-CS and the ACM (Tucker et al., 2006) which has been revised later on by the Computer Science Teachers Association (CSTA, 2011) in 2011. In Germany the national CS-Society 'Gesellschaft für Informatik' (GI, 2008) published a proposal of informatics standards for lower secondary schools.

Nevertheless, these standards weren't based on an empirically proofed competence model for CSE. Therefore, in 2004 the research community of Didactics of Informatics in Germany started during the Dagstuhl-Seminar "Concepts of Empirical Research and Standardisation of Measurement in the Area of Didactics of Informatics" (Magenheim, Schubert, 2004) a discussion about educational standards of CSE on a higher secondary school level. A result of this seminar was the comparison of the different approaches to educational standards in Mathematics and CSE. The results of the seminar revealed that further theoretical and empirical research was necessary to examine the opportunities of the measurement of educational standards of CSE and that respective research should be founded on a sound CSE competence model.

In an effort to develop such a competence model researchers in the fields of CSE and psychology started their research on this subject area. The project

MoKoM (Modelling and Measurement of Competences in CSE) funded by the German Research Foundation (DFG) from 2008 to 2014 developed a competence model and measured related competences of senior class students. The research project focused on two specific domains: informatics system comprehension and object oriented modelling. In the present paper we describe the objectives and research methodology of the project MoKoM on object-oriented modelling and system comprehension (section 2) along with the actual research results: an empirically refined competence model (ECM) (section 3), a derived measurement instrument (section 4), results of an empirical survey which has been conducted in Germany by applying the MoKoM-instruments (section 5) and finally first steps towards a competence level model (section 6). In conclusion we give an outlook on the necessity of further research in this subject area (section 7).

2 Objectives and Research Methodology

In alignment with the discussion on CSE standards in secondary education and in order to develop a CSE competence model the project MoKoM investigated the following main research questions:

- Which competencies are necessary for informatics system application, informatics system comprehension and informatics system modelling in upper secondary education?
- How can these competencies be related to a theoretical derived competence model (TCM)?
- How could the TCM be used to gain an empirically refined competence model (ECM)?
- Which test items are adequate to measure these competencies of the learners in CSE with a competence-based test-instrument?
- Is the test-instrument able to measure the described informatics competences in a valid and reliable way when applied to a large sample of senior students?
- Can such a test-instrument be used to distinguish between different competences of a group of students?
- Does the test instrument validate the assumed competence model resp. competence structure?
- How can this model be used for the grading of competencies and how can it be used to evaluate the learning outcomes of a specific CSE-learning setting?

In a first phase of the project, competence definitions, expert papers and CSE curricula were analysed. Thus, all competence dimensions were theoretically derived from international syllabi and curricula, e.g., the "Computing Curriculum 2001" of ACM and IEEE (Cross, Denning, 2001), the "Model Curriculum for K-12 Computer Science" of the ACM (CSTA, 2011) and a variety of other ACM, IEEE, IFIP, GI and CSTA (e.g. CS2013) publications. Additionally, expert papers like the Rational Unified Process for software development (IBM, 1998) were used to identify important competence components for system modelling. Based on the analysis of these resources and applying Weinert's definition of competence (Weinert, 2001), a first competence framework, containing cognitive and non-cognitive competences was developed.

But a restriction on exclusively theoretically derived competencies would risk that the reference of competencies to complex requirements in real situations is neglected or disregarded. Therefore, an additional step was necessary in order to determine competencies more reliably, that is, ensuring an empirical access to determinate the relevant competencies. Conducting expert interviews by applying the Critical Incident Technique represents an appropriate empirical approach to detect the relevant competencies in the subject domains 'system comprehension' and 'object-oriented modelling'.

The interviews of the 30 experts (experts in the domain of didactics of informatics, computer scientists and expert informatics teachers) were based on a structured questionnaire manual and included twelve hypothetic scenarios (see figure 1) concerning application, testing, modifying and developing of informatics systems. The expert interviews were transcribed in full and analysed by means of qualitative content analysis according to Mayring (Mayring, 2003). The requirements of intercoder reliability were also considered during this empirical phase of analysis and were sufficiently achieved.

Scenario: "*You are asked by a colleague to test his software, which was developed to solve configuration problems, e.g. to set up a new car or a new computer.*"
Question 1: *"What is your strategy of testing to solve this problem? Which aspects do you have to bear in mind?"*
Question 2: *"Which cognitive skills are required for such a software exploration?"*
Question 2.1: *"Which informatics views are important for this task?"*
Question 2.2: *"Which complexity would you assign to this task?"*
Question 3: *"Are there any attitudes or social communicative and cooperative skills which are necessary to accomplish this?"*
Question 4: *"Which differences of competence levels would you expect between novices and experts?"*
Question 5: *"Could you imagine a potential pupil's procedure to solve this problem?"*
Question 6: *"Which obstacles would pupils have to cope with?"*

Figure 1: Interview scenario

The results of the qualitative content analysis have to be structured according to the dimensions of the competence model. Relations between the competence components and meaning units in the interview have to be found and described. An example shows the answer about social-communicative skills: "There is a serious contradiction between the competence of problem solving and the social-communicative competencies." This means it is necessary to supervise the development of social-communicative competencies, since they are not fostered as a side effect of informatics problem solving. Another example shows the answer about empathy, change of perspectives and roles: "When we test software of others, we have to learn to criticize in a fair and sensitive way." The task of systematic testing gives the opportunity to gain non-cognitive competencies on a higher level when the learner presents his results to other learners, e.g. the explanation of use cases, the presentation of test results including the visualization of large data collections.

3 Competence Model on Informatics System Comprehension and Object-Oriented Modeling

The described content analytic procedure led us to an empirically refined competence model (see figure 2). But the described empirical procedure to complement the theoretical model is nevertheless restricted. One methodological restriction implies, that the relevant competence requirements are closely linked

to the used scenarios. So it is important that the scenarios contain at least typical and representative tasks and problems to be solved. This was ensured by the representative ratings of the experts. Furthermore, the actions described by the informatics experts might not necessarily mirror their actual behaviour in those scenarios because they could describe idealized actions to solve the problems in the scenarios. On this issue, the different orientations of expertise of the interviewees serve as a corrective to some extent. The deployment of the qualitative content analysis took place adhering to comprehensible, methodical rules and principles. Nevertheless, qualitative analyses include inevitably interpretative processes, which might restrict the objectivity, reliability and validity of the described analyses.

As a result of these research efforts in the MoKoM-project a theoretically derived and empirically refined competence model was developed.

The empirical refined competence model contains four cognitive dimensions K1 'System application', K2 'System comprehension', K3 'System development' and K4 'Dealing with system complexity'. Additionally a non-cognitive dimension K5 covers 'Non-cognitive skills'.

A condensed version is depicted in figure 2. The extended version with all sub-categories was published in 2013 (Linck et. al., 2013).

While these categories of the competence model represent only the structure of the model in terms of components and hierarchy the derived items were contextualized and meet the requirements of competence definitions regarding a person´s ability to perform observable action.

K1 System application
K1.1 Structuring of application field K1.2 System exploration K1.3 System selection K1.4 Use of media to foster system application K1.5 Transfer to new application fields

K2 System comprehension
K2.1 System requirements K2.2 Systematic tests K2.3 System exploration K2.4 Evaluation of software quality K2.5 Architecture & organization K2.6 Algorithms & data structures K2.7 Informatics' Views

K3 System development
K3.1 Software development process models K3.2 Business Modeling K3.3 Requirements K3.4 Analysis K3.5 Design K3.6 Implementation K3.7 Test K3.8 Iterative development

K4 Dealing with system complexity
K4.1 Measures of complexity: Time & Space K4.2 Number of components K4.3 Level of networkedness K4.4 Stand-alone vs. distributed systems K4.5 Level of human-computer interaction K4.6 Combinatorial complexity

K5 Non-cognitive skills
K5.1 Attitudes K5.2 Social-communicative skills K5.3 Motivational and volitional skills

Figure 2: Competence Model

4 Development of a Competence Measurement Instrument

During further methodical steps, as described in the following sections, test items and an empirical test instrument was developed on the basis of the refined competence model. The empirical test instrument was applied on a representative sample of students in secondary schools in Germany, mainly from Bavaria and North Rhine Westphalia. The results of this survey not only provides an insight into the competencies and abilities of students in CSE at secondary schools but enables us to development a competence level model for the needs of grading competences.

4.1 Principles of Competence Measurement

Based on the empirical competence model the test instrument was developed following the principles of Situational Judgment Tests (SJT; Weekly, Ployhart, 2006). This means that we created knowledge application scenarios which specifically addressed the specific competence requirements of each model facet that had to be operationalized. We also took into consideration experiences of how to construct competence measurement items gained in large scale studies like TIMMS, PISA and DESI.

Based on detailed competence descriptions, tasks for every single competence item were created. After this, the answering format was created. In the majority, this included closed answering formats like multiple choice or classification items. But also tasks with open questions that required short sentences or the statement of keywords as answers were used. The answering format was chosen and created in accordance with the cognitive requirements and levels (according to the cognitive dimension of the Anderson, Krathwohl, 2001 taxonomy) that had to be addressed. We also used a complex item format which included multiple items resp. tasks that were integrated in one complex application scenario. So, we were able to address different competence facets in one task context and by this economize the measurement. To allow an objective and reliable appraisal of the answers (especially when evaluating open item format), a comprehensive grading manual was created alongside the test items. This contained different sample solutions as well as approaches to grade answers.

The test instrument was examined and optimized by conducting a preliminary test with students from local secondary schools. In addition, student computer science teachers from didactical courses at the universities of Paderborn and Siegen were asked to review the instrument. The main issues found during

this pre-test were ambiguous wording and oversight mistakes on the one hand and difficulties of applicability of the tasks on the other. Rewriting or extending the context of the tasks could easily fix the latter.

4.2 Design of Test Items of CSE

The empirically refined competence model allows the defining of competence profiles, which are the basis for the model for the instruments of competence measurement (see figure 3).

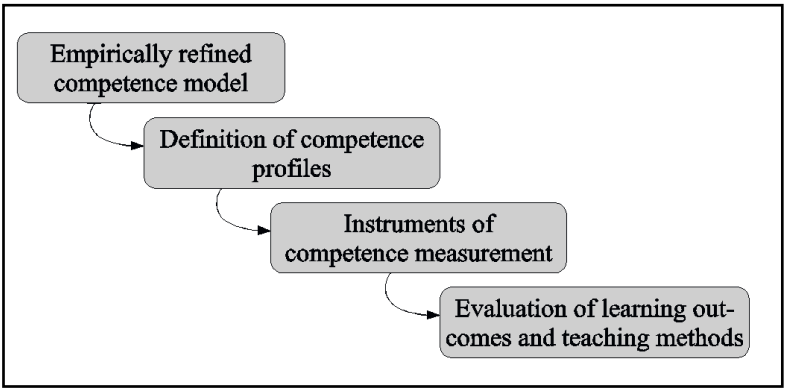

Figure 3: Impact of competence profiles on learning

To illustrate the procedure of defining competence profiles on informatics system comprehension, we will start with an example: (1) A competence component of the empirically refined competence model is chosen, e.g. “Errors as Learning Opportunities”. (2) All main expert statements, which are related with this competence component, are collected from the spreadsheets. (3) Step 3 is to select citations of the collection of expert statements, which have the most meaningful expressions. Such citations of expert statements will be called “anchored examples”. In this case two anchored examples are related to this component:

I. *“Most important is the ability not to give up after the first syntax error, but to learn from them, and to determine error messages. I want to deliver a completed product, which actually does, what it should do.”*

II. *"You have to intervene in this case and reflect once again, and in the very moment, when it happens, say: This error, you will never do it again."*

(4) A first competence profile definition of "Errors as Learning Opportunities" is based on the content of both statements. (I) implies that students should require the competence to identify errors and (II) implies that learners should detect and avoid errors:

> *"The learners are able to determine, to assess and to examine system-based errors. This acquired knowledge will be applied to error avoidance and improvement of tests."*

(5) Is to improve this definition. Therefore, keywords, referring to the cognitive processes in (Anderson, Krathwohl, 2001), are used. These keywords are called operators. An operator is a work instruction, which refers to the content and to the methods to solve a given task. In the competence profiles the operator ensures that misinterpretations of the requirements towards the learners are reduced. In the competence profile definition above is "apply" an operator. In contrast, "determine, assess and examine" have to be discussed. The challenge is to find synonyms or similar expressions and express the meaning of the first competence profile definition. A refined definition of the competence profile follows:

> *"The learners are able to identify, to differentiate, and to judge system-based errors. This acquired knowledge is applied to error avoidance and improvement of tests."*

Four operators "to identify, to differentiate, to judge, and to apply" are used in the definition of competence profiles. These operators support our aim, which is to assure the standardisation. After defining competence profiles for each component of the empirically refined competence model, test items can be developed. These test items measure the individual performance of a learner related to different components of the competence model in classroom practice. All cognitive and non-cognitive process dimensions, which are defined in a competence profile, have to be tested by such items. We developed and improved such test items with CSE teachers. This is an example of a successful test item: "You got the homework to write an algorithm, which sums up all numbers from 0 to n. Your friend already gave you his ideas noted in a pseudo

code (see figure 4). Decide which of the two algorithms is better regarding the running time."

Enter: n Set sum = 0 Set i = 0 Repeat from 0 to n Set sum = sum + i Set i = i + 1 Return sum	Enter: n Set sum = 0 If n odd-numbered, then Set sum = sum + n Set n = n - 1 Set sum = (n / 2) * (n + 1) Return sum

Figure 4: Algorithms in pseudo code

5 Applying the Measurement Instrument

5.1 The Population of the Test

Due to the large amount of items, the test instrument was not applicable in a classroom setting with timeslots of usually 90 minutes. Furthermore, students' attention to the test instrument shouldn't be required more than this time span. In order to adapt the instrument to a 90-minute timeslot, the items were divided into six blocks. Then six booklets were compiled from three item blocks each. Together with an additional questionnaire on attitudinal, motivational and volitional competences (representing facets of the dimension "non-cognitive skills" resp. K5), the whole test can be accomplished within 90 minutes. The application of such an arrangement, called "matrix design", is possible due to the application of the 'Item Response Theory' to analyse the test results. Though not all students answer every task due to not having them in their booklet and thus produce a lot of "missing values" in the final data, the IRT allows the calculated estimation of student abilities in combination with the overall item difficulty. This method provides coherent results even if the students worked on different subsets of items (Hartig, 2008), (Rost, 2004). The booklets were distributed to more than 800 computer science students in German upper secondary schools. The analysis of the returned data was done with ACER ConQuest, applying a 1PL partial credit model to estimate the item difficulties (Wu, Adams, Wilson, Haldane, 2007).

The booklets were originally distributed in 26 classes with 522 students in North Rhine Westphalia. Additionally 6 classes from Berlin, Hessen and Lower Saxony with a total of 82 students also participated. In Bavaria 244 students from 11 different classes (6 classes of grade 10 and 3 of grade 11) took part in

the test. According to the curriculum, the current learning content of most of the responding students was focused on object-orientation, the use of standards software like databases or spread-sheets and simplest concepts of programming. The test was conducted as a pencil-and-paper-test. The print-versions of the booklets were sent to teachers who volunteered to deliver them to their classes. To prevent the students from cheating, each teacher received two to three different booklets to distribute them among the class. From more than 800 tests we sent out we received back 583 completed and evaluable booklets. The investigated sample consists of 86 % male and 14 % female students with an average age of 17.5 years. 17 % of them had an immigrant background. Their self-assessed proficiency in computer science on a scale from 1 to 6 averaged at 2.65 points. They had participated in computer science classes for a mean of 3.5 years. Only 3.3 % had dropped the subject in the interim.

5.2 Analysing the Test Data – Test of Model Fit

The gathered data were analysed according to the Multidimensional Item Response Theory (MIRT). The main goal was to examine the dimensional validity resp. structure of the competence model and the reliability measurement instrument. To do so, several different IRT models were used to analyze the empirical data and the results were compared to assess the best fitting model.

IRT models assume that personality traits cannot be measured directly and test results can only be interpreted as an indicator for the existence and intensity of such a trait. Therefore, IRT models differentiate between latent variables, that can't be measured directly, but influence the response to a test item, and manifest responses that are assumed to be the observable manifestations of the latent traits. Thus, the ability of the test subject can be inferred from the responses. Furthermore, it is assumed that any subject has a certain probability to answer any item right or wrong. The difficulty of the item and the ability level of the subject determine this probability.

IRT has several advantages for the assessment of competences. For once, the estimation of the item difficulties and student abilities does not require for every participant to work on every task of the test instrument. This allows to use a matrix design with different booklets that only represent a part (about three-fourths) of the item resp. task pool of the competence test. Furthermore, the estimated parameters can be interpreted on the same scale and easily related to each other.

Since competence structures are complex constructs, they often result in multidimensional competence models. In our case this applies to the cognitive dimensions K1 to K4 with the additional non-cognitive dimension K5. The latter was excluded from the IRT analyses because the data for this dimension was raised by a questionnaire. To evaluate the dimensionality of the empirical data, multidimensional IRT models can be utilized, which assume that multiple latent variables (one per dimension) cause the responses to a test. Furthermore, MIRT allows for the comparison of different models, by analysing the conformity of the theorized model to the empirical data.

We also had to choose between a speed test and a power test variant to analyze the data (this has consequences concerning the handling of missing values). There are reasonable arguments for both variants. Therefore, we analyzed both. Since the results of both variants are very similar though, this article will concentrate on the results of the speed option. To calculate the MIRT analysis we used ACER ConQuest Version 2.

To evaluate the structure of the competence model, we analyzed four different IRT models with one to four assumed dimensions respectively. Since the test items were crafted with the intent to test for one specific competence, a between-item multidimensionality model was used in all cases. Because not all items could be coded as dichotomous responses, the partial credit model was applied to analyze dichotomous and polytomous data alike. Starting with the one-dimensional model, for which it was assumed that all items loaded on the same latent trait, every model added one additional dimension in accordance with the assumptions concerning the structure resp. dimensionality of the competence model described above. The analyses results concerning the IRT models with different competence dimensions can be seen in table 1.

Table 1: Final deviance, estimated parameters and reliability for evaluated models

Model	**Final Deviance**	**Estimated Parameters**	**Reliability for dimension 1 to 4 (if available)**
1-Dim	87379.09538	316	0.872 (K1,K2,K3,K4)
2-Dim	86695.99173	319	0.831 (K1) / 0.831 (K3)
3-Dim	86403.83657	323	0.749 (K1) / 0.806 (K2,K4) / 0.812 (K3)
4-Dim	85891.85717	328	0.779 (K1) / 0.763 (K2) / 0.861 (K3) / 0.759 (K4)

To compare the models, the final deviance – an indicator of how well the empirical data fits the IRT model – and the number of estimated parameters reported by Con-Quest can be used. Usually, both parameters should be as low as possible. If it is not possible to choose the better model by comparing the values alone (because one value is lower, while the other one is bigger than the parameters of the second model), a Chi-Square-Test can be calculated, using the difference in deviance and the difference in estimated parameters as the degrees-of-freedom. If the result is significant, the model with the smaller deviance parameter is selected. Otherwise the model with the lower amount of estimated parameters is deemed the better one. The parameters for each evaluated model can be seen in table 1.

Table 2: Chi-Square statistics for model comparisons with difference in deviance and difference in estimated parameters as degrees of freedom

	2-Dim	**3-Dim**	**4-Dim**
1-Dim	$\chi(3)2=683.1$, $p<.001$	$\chi(7)2=975.26$, $p<.001$	$\chi(12)2=1487.24$, $p<.001$
2-Dim		$\chi(4)2=292.15$, $p<.001$	$\chi(9)2=804.13$, $p<.001$
3-Dim			$\chi(5)2=511.98$, $p<.001$

Since with increasing dimensions the deviance decreases and the number of parameters increases, a Chi-square-test was calculated for every combination of models (see table 2). In every case the result was statistically significant and since the models with a higher number of dimensions have a lower deviance, it can be assumed that they better match the empirical data than the models with fewer dimensions. Thus, the four-dimensional model has the best model fit overall.

5.3 Analyzing the Test Data – Item Fit and Reliability

ConQuest calculates the EAP/PV reliability for each dimension, which can be compared to Cronbach's Alpha. Table 1 shows the reliability for all dimensions in each model. All values exceed 0.7 and can be considered acceptable.

To further evaluate the models, the item fit for individual items can be examined. The fit compares the predicted probabilities for each item within the model with the observed responses. To do this, ConQuest calculates the weighted mean squares (wMNSQ), which are expected to be 1 for perfectly fitting items. The wMNSQ for a good fitting item should fall between 0.8 and 1.2, and the corresponding t-values should not be greater than 1.96. Further-

more, the discrimination parameter shows how an item correlates to the overall test results. With the discrimination close to 0, an item may not be useful to differentiate between students with high levels of a trait and those with low levels. Values between 0.4 and 0.7 are considered good while values above 0.3 can be considered as acceptable.

The data for all models (see table 3) showed a good item fit overall, but the percentage of unfit items increased for models with more dimensions, from below 1 % (2 items out of 292) for the one-dimensional to 4.7 % (14 items) for the four-dimensional model. In addition, the number of items that might have a bad fit according to the t-values increased from 27 to 37 items. Unfortunately the discrimination parameters are not very good for a large part of the items. Just 22.6 % (66 items) are above the 0.4 threshold and even if we adjust the point at which an item is considered to have a too small discrimination to 0.3, roughly 43.8 % (128 items) remain under that line. Only one item had a negative discrimination, which was close to 0. The high ratio of low discrimination items necessitates a throughout examination of the items and how they fit to their corresponding dimension in further steps.

Table 3: Range of mean squares, t-values and discrimination values for all models

Model	**wMNSO**	***t***	**Discrimination**
1-Dim	0.86≤wMNSQ≤1.3	-2,9≤t≤4.4	-0.04≤Disc. ≤0.58
2-Dim	0.77≤wMNSQ≤1.42	-4.1≤t≤5.2	-0.04≤Disc. ≤0.58
3-Dim	0.76≤wMNSQ≤1.42	-4,2≤t≤5.2	-0.04≤Disc. ≤0.58
4-Dim	0.65≤wMNSQ≤1.42	-5.7≤t≤5.3	-0.04≤Disc. ≤0.58

5.4 Analyzing the Test Data – Difficulty Parameters and Latent Abilities

Main goal of IRT analysis is the estimation of two parameters: the item difficulty, that denotes the probability of answering an item correct given a certain level of the measured construct, and person parameters, that assess the level of the latent trait for individual students. One advantage of IRT analysis is that both estimates can be arranged on the same scale and easily compared. The item-person-map for each model visualizes the item difficulties on the right, by ordering them from more difficult (top) to less difficult (bottom), and the latent trait levels on the left (grouping persons with the same values together). Ideally, the item difficulties should be well dispersed around the mean, having the

most items in the medium difficulty range, but also providing items with high and low difficulties. Additionally, the latent traits are separated by dimension. Figure 5 shows the maps for the one- and four-dimensional models. As can be seen, the item difficulties are well distributed along the axis, though there are some aspects that have to be noticed and commented.

First, there are some outliers in the upper part of each map. This indicates, that some items are way to difficult for the targeted student groups, since no person was estimated to have a high enough proficiency to solve these items with an adequate probability.

Second, the latent traits in the different dimensional solutions are somewhat uneven dispersed. While the one-dimensional model indicates, that the overall difficulty of the test matches the ability of the population, the four-dimensional model reveals, that only the third dimension can be considered well matched. Dimension 1 and 4 lack items in the upper difficulty range, while dimension 2 necessitates less difficult items to adequately assess its competences.

Figure 5: Overview of the estimated item parameters for the one- and four-dimensional model

6 Modelling of Competence Levels

In further evaluation steps of our test instrument we want to grade the measured competences of modelling and comprehending informatics systems which can be interpreted as competence levels of the developed model. To create a competence level model you have to choose between different approaches (Hartig, 2004). We decided to use an inductive approach which is based on systematic post hoc analyses of task contents and requirements. To apply this approach, different steps have to be conducted to identify and generate the desired competence levels measured by a certain competence test:

First, you have to identify and define task features that determine the difficulty when coping with the task requirements and contents. Secondly, you have to determine and describe the different grades or levels of difficulty concerning each difficulty feature. In a third step you have to determine how the different difficulty features and grades are represented in each test item. Therefore you have to conduct an expert rating at which the experts examine and rate each item if specific difficulty features and levels are given or required when coping with the item. In a fourth step the expert ratings of the difficulty features of each test item are related to the empirically determined difficulty parameters (when the test is applied to a large sample of students). This is conducted by regression analyses to test if the assumed difficulty features and grades are really determining the empirically determined difficulty of the items. Only those difficulty features and grades that prove to be significant predictors of the empirical difficulty are kept in the further process of defining the competence levels. In a fifth step the items are ordered concerning their empirically determined difficulty and in an adjunct table for each item it is systematically determined and described if a difficulty feature is realized in the requirements and at which difficulty grade resp. level. This table is used in a sixth step to determine and define thresholds of competence levels. This is usually the case, when new difficulty features or grades appear at a certain type of items. After you have determined such thresholds and the number of different competence levels you have to describe each level in a seventh step. Therefore, you have to take reference to the requirements of the items that belong to a specific competence level. These requirements are especially derived from the difficulty features and grades, which characterize these group of items typically. In a last step you have to classify the persons of your sample according to the competence levels to determine how the sample is distributed over the competence levels.

In the following we describe the analyses we have conducted so far to generate a competence level on the basis of our test instrument and study sample.

6.1 Identification and Description of Difficulty Relevant Features of the Competency Test Items

To identify and describe difficulty relevant features of the competency test we first defined difficulty relevant features of the competence test items. We derived those features from the literature concerning difficulty relevant features of competence tests in general (e.g. Schaper et al., 2008). Furthermore we analysed the items concerning informatics specific difficulty facets and tried to define and grade them analogue to the more general features. On this basis altogether thirteen features were identified and defined: addressed knowledge taxonomy level (KTL), cognitive process dimensions (CP), cognitive combination- and differentiation capacities (CCD), cognitive strain (CS), scope of tasks (necessary materials, reading effort and understanding) (ST), inner- vs. outer computational task formulation, aspects of demands of computer science (IOC), number of components, level of connectedness (NC), stand-alone vs. distributed system (SDS), level of human-computer-interaction (HCI), (mathematical) combinatorial complexity (CC), level of the necessary understanding of systems of computer science (LUS), level of the necessary modelling competence of computer science (LMC). Because of extent restrictions only two of these features are described in more detail.

6.2 Cognitive Process Dimensions

Concerning this difficulty determining feature we analysed the structure of the cognitive process dimensions of the revised taxonomy for learning, teaching, and assessing by Anderson and Krathwohl (Anderson, Krathwohl, 2001). We assumed that these addressed the following process categories: 1. Remember, 2. Understand, 3. Apply, 4. Analyze, 5. Evaluate and 6. Create, which were also used to differentiate between different cognitive requirement levels of our test items. So we defined this difficulty-relevant feature with the following six feature levels:

- CP1: The successful solution of the task requires a memory performance. The students recall relevant knowledge contents from their memory.
- CP2: The successful solution of the task requires a comprehension performance. The students understand terms, concepts, and procedures of computer science and can explain, present and give examples for them.

- CP3: The successful solution of the task requires an application performance. The students are able to implement known contents, concepts and procedures within a familiar as well as an unfamiliar context.
- CP4: The successful solution of the task requires an analysis. The students are able to differentiate between relevant and irrelevant contents, concepts and procedures. They choose the suitable procedures from a pool of available procedures.

6.3 Cognitive Combination and Differentiation Capacities

We assumed that this feature addresses different forms of knowledge utilization like *Reproduction, Application, Networked application*, and that these requirements differentiate between different levels of difficulty concerning our test items. So we derived the third difficulty-relevant feature with the following three feature levels:

- CCD1: Reproduction of computer science knowledge and application of single, elementary terms, concepts and procedures of computer science in close contexts (no cognitive combination capacities required).
- CCD2: Application of single terms, concepts and procedures of computer science in bigger contexts, whereas an argumentative and/or intellectual consideration between competitive terms, concepts and procedures (approaches) for example has to be made.
- CCD3: Networked Application of terms, concepts and procedures of computer science in different, especially bigger scenarios, whereas an argumentative and/or intellectual consideration between competitive terms, concepts and procedures (approaches) for example has to be made (multiple challenging cognitive combination capacities required).

6.4 Expert Rating of the Difficulty Determining Task Resp. Item Features

In a second step we used the described features of task difficulty to rate the difficulties of the items of our competence test. Therefore experts in computer science education were asked to rate each item of the competence test with reference to the thirteen difficulty features. To conduct the expert rating a rating scheme and instructions were formulated. Furthermore, the measurement instrument was split into four parts of roughly equal size to keep the amount

of ratings at an acceptable extent. Each of the four instrument parts – including solutions for the items – was presented to two selected experts in the field of didactics of informatics, along with an explanation of each feature and its rating levels. The experts were asked to answer each item on their own, compare the solution with the given sample solution and then rate the item for each of the features. In addition, the experts had to give a subjective rating of the item difficulty on a scale from one to ten.

The resulting two ratings for each item were compared and treated in three ways: 1. exact matches between the two ratings of a certain feature per item were accepted and not further treated; 2. items with small rating differences (if the ratings only differ one point or grade from each other) were discussed within the project group to decide upon a final rating; 3. items with big rating differences (if the ratings deviate two points and more from each other); these cases were presented to two further experts that had to rate these features for a certain item again while considering the ratings of the two preliminary experts; again, resulting differences of these experts were discussed in the project group to decide upon a final rating. The expert group was composed of seven researchers with background in computer science, computer science education and psychology.

The rating process resulted in a classification of 74 items concerning each of the described difficulty determining features. The rating levels for each feature were coded as ordinal dimensions, e.g. coding KTL1 as 1 and KTL2 as 2. For every feature the "not relevant" rating was coded as 0. This way, we ended up with 13 nominal variables with n+1 categories for a feature with n levels. For almost all features it was reasonable to assume a ranking of the levels in the order they are described above. The assumption is that a higher level correlates with a higher item difficulty. As this assumption does not necessarily have to be true, the order was examined by the analysis of the rating data. This was done using descriptive and explorative methods to determine the relevant features that influence the item difficulty.

In the following only some of the results of the expert rating are described and summarized: The number of ratings of features related to cognitive demands like KTL, CP and CCD are mostly distributed at the medium rating levels. This makes sense and was intended when creating the tasks: The instrument should provide mainly items with a medium difficulty, since it can be expected for most subjects that they are able to solve items of medium difficulty. Therefore, the test instrument has to differentiate the best at this difficulty resp. competence level. In the upper difficulty range fewer items are required, since this would be enough to show the expertise of the more competent students.

The expert ratings though, show a tendency to lower rating levels. For example the cognitive process dimension “remember” was assigned more times (8) than the dimensions “evaluate” and “create” which were combined at one grading level (4 times). The same can be observed for the two features CS and CCD. For the features CS and ST the predominance of the lower rating levels is a result of the test design. To create an applicable instrument, the tasks need to adhere to certain constrains and thus the most items require only few processing steps and a minimal amount of additional materials. The overall difficulty of the test instrument was subjectively rated by the experts with a mean of 4.2 on a ten-point scale.

6.5 Regression Analysis and Further Analysis Steps

To determine which features have the most influence on the item difficulty, the expert ratings were related to the empirical difficulty estimates that were calculated by means of the Item Response Theory (IRT) (Schaper et al., 2008). The relations between the difficulty determining features rated by experts and the empirically determined item difficulty are examined by regression analyses. These analyses are not computed and evaluated at the moment though and therefore cannot be reported here at the moment. Also, to model the competence levels for our test instrument and model we still have to conduct the further analyses steps described at the beginning of this section. This will therefore be reported at another place later on.

7 Conclusion and Further Work

In this article we outlined essential research questions and the corresponding research methodology of the project MoKoM concerning upper secondary students’ competences. As a first main result we developed a theoretically grounded and empirically refined competence structure model in the subject area of informatics system comprehension and object-oriented modelling. Based on this model an empirical test instrument was developed and an empirical survey conducted. By applying IRT evaluation methodology to construct the test-instrument and to assess the data, gained from the survey with 583 upper secondary students in Germany. We finally took first steps to develop a competence level model that also considers the results of an expert rating on the difficulty levels of the test-items. Thus, we answered several of the research questions, which have been raised at the beginning of this article.

We also proved that our test instrument was able to identify competence profiles of learners and to indicate the difference of competences between members of a learning group (Neugebauer et al., 2014). We also conducted a survey in a joint project with the German University of Distance Learning (FernUniversität Hagen) on students in an introductory course of object-oriented software engineering. We were able to show, that the instrument could even be applied at undergraduate university level. The students underwent the test at the beginning and the end of the CS-course and we were able to analyse the students' increase in subject-related CS-competences during the course (Hering et al., 2014). Further research of the project will concentrate on the application of the MoKoM test-instruments to evaluate the learning outcomes of specific learning design settings in CSE. In general the MoKoM competence model and the related test instrument should be used to contribute to the theoretically founded and empirically based development of standards in CSE. Furthermore, the application of the test-instrument on an enhanced sample of students could provide an overview on students competences in CSE and reveal a possible gap between these competences students' really own and the expected learning outcomes according to the curricula of CSE.

References

Adams, R. J.: *Scaling PISA cognitive data. PISA Programme for International Student Assessment (PISA 2002)*. PISA 2000 Technical Report. pp. 99–108. OECD Publishing, Paris (2002).

Anderson, L. W., Krathwohl, D. R. (2001): *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. New York: Addison Wesley Longman.

Cross, J., Denning, P. (2001): *Computing Curriculum 2001, The Joint Curriculum Task Force IEEE-CS/ ACM Report*. 2001, www.computer.org/education/cc2001.

CS2013 Steering Committee, The Joint Task Force on Computing Curricula Association for Computing Machinery IEEE-Computer Society (2012): *Computer Science Curricula 2013*. http://ai.stanford.edu/users/sahami/CS2013/

German Informatics Society (GI) (2008): *Grundsätze und Standards für die Informatik in der Schule. Bildungsstandards Informatik für die Sekundarstufe I* (in German). Retrieved from: http://www.informatikstandards.de/

Hartig, J. (2004): Skalierung und Definition von Kompetenzniveaus. In: B. Beck, E. Klieme (Eds.): *Sprachliche Kompetenzen. Konzepte und Messung*. DESI-Studie. Weinheim: Beltz.

Hartig, J. et al. (2008): Methodische Grundlagen der Messung und Erklärung sprachlicher Kompetenzen. In: DESI-Konsortium (ed.) *Unterricht und Kompetenzerwerb in Deutsch und Englisch. Ergebnisse der DESI-Studie*. pp. 34–54. Beltz, Weinheim, Basel.

Hering, W., Huppertz, H., Krämer, B., Magenheim, J., Neugebauer J., Schreier, S. (2014): On Benefits of Interactive Online Learning in Higher Distance Education. Case Study in the Context of Programming Education. In: *Procceedings of the IARIA EL&ML 2014*, Barcelona.

IBM (ed.) (1998): *Rational Unified Process, Best Practices for Software Development Teams*. Retrieved from: http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

Lehner, L., Magenheim, J., Nelles, W., Rhode, T., Schaper, N., Schubert S., Stechert, P. (2010): Informatics Systems and Modelling – Case Studies of Expert Interviews. In: Reynolds, N.; Turcsányi-Szabó, M.: *Key Competencies in the Knowledge Society*. Springer, Boston, pp. 222–233.

Linck, B., Ohrndorf, L., Schubert, S., Magenheim, J., Nelles, W., Neugebauer, J., Schaper, N., Stechert, P. (2013): Competence model for informatics modelling and system comprehension. In: *Proceedings of the IEEE Global Engineering Education Conference (EDUCON)*. Berlin, pp. 85–93, DOI: 10.1109/EduCon.2013.6530090

Magenheim, J., Schubert, S. (eds.) (2004): *Concepts of Empirical Research and Standardisation of Measurement in the Area of Didactics of Informatics*. GI-Dag-

stuhl-Seminar, 19th–24th September 2004, Volume 1, Seminar-Edition “Lecture Notes in Informatics”, German Informatics society (GI).

Magenheim, J., Nelles, W., Rhode, T., Schaper, N., Schubert, S., Stechert, P. (2010): Competencies for Informatics Systems and Modeling. Results of Qualitative Content Analysis of Expert Interviews. In: *Proceedings of the Engineering Education Conference 2010, IEEE Computer Society*, pp. 513–521, DOI: 10.1109/EDUCON.2010.5492535

Mayring, P. (2003): *Qualitative Inhaltsanalyse*. Beltz, Weinheim, 2003.

Nelles, W., Schaper, N.: Developing a Competence Model for the domains of Informatics Modeling, Informatics System Comprehension and Application. In: *Proceedings of Earli Jure 2012 “A Learning Odyssey: Exploring new Horizons in Learning and Instruction”*, 23–27 July 2012, Regensburg, in press, URL: http://www.earli-jure2012.org/

Neugebauer, J., Hubwieser, P., Magenheim, J., Ohrndorf, L., Schaper, N., Schubert, S. (2014): *Measuring Student Competences in German Upper Secondary Computer Science Education*; accepted Full paper to be published in the Proceedings of the ISSEP 2014, Istanbul.

Rost, J. (2004): *Lehrbuch Testtheorie–Testkonstruktion*. Huber, Bern, Göttingen, Toronto, Seattle (2004).

Schaper, N., Ulbricht, T., Hochholdinger, S. (2008): Zusammenhang von Anforderungsmerkmalen und Schwierigkeitsparametern der MT21 Items. In: Bloemeke, S., Kaiser, G., Lehmann, R. (Eds.): *Professionelle Kompetenz angehender Lehrerinnen und Lehrer*. Muenster: Waxmann.

Tucker, A. (ed), Deek, F., Jones, J., McCowan, D., Stephenson, C., Verno, A. (2006): *A Model Curriculum for K-12 Computer Science. Final Report of the ACM K-12 Task Force Curriculum Committee*. (2nd Edition). Association for Computing Machinery (ACM).

The CSTA Standards Task Force. (2011): *CSTA K-12 ComputerScience Standards*. (D. Seehorn, S. Carey, B. Fuschetto, I. Lee, D. Moix, D. O’Grady-Cunniff, et al., Eds.) (pp. 1–73). http://csta.acm.org/Curriculum/sub/K12Standards.html

Weekly, J.A., Ployhart, R.E. (2006): *Situational Judgement Tests. Theory, Measurement and Application*. Mahwah, N.J.: Lawrence Erlbaum.

Weinert, F. E.: Concept of Competence: A Conceptual Clarification. In: Rychen, D., Salganik, L. (eds.): *Defining and Selecting Key Competences*. Seattle, (2001).

Wu, M. L., Adams, R. J., Wilson, M. R., Haldane, S. A. (2007): *ACER ConQuest Version 2.0: Generalised item response modelling software*. Melbourne, Australia: ACER Press.

Biographies

Johannes Magenheim is a Professor for Computer Science Education at the Institute of Computer Science at the University of Paderborn. His primary areas of research and teaching are CSE, E-Learning, CSCL. He is also member of different working groups of the German Informatics Society (GI) and member of IFIP WGs 3.1 and 3.3.

Niclas Schaper is a Professor for Work and Organisational Psychology at the University of Paderborn. His primary areas of research are job analysis, occupational training, approaches of competency modeling, methods of competency measurement, E-Learning, and teaching in higher education. He is a member of the German Society of Psychology and vice president of the German Society of Teaching in Higher Education.

Since 1979 **Sigrid Schubert** has taught informatics in secondary, vocational and higher education. She has been professor of "Didactics of Informatics and E-Learning" (Universities Siegen and Dortmund, Germany) since 1998. Her research interests are Informatics Teacher Education and E-Learning. She is Fellow of the German Society for Informatics (GI), member of the Technical Committee 3 "Education" of the International Federation for Information Processing (IFIP) and chair of the IFIP Working Group 3.1 "Informatics and digital technologies in School Education".