Expert Rating of Competence Levels in Upper Secondary Computer Science Education

Johannes Magenheim[1], **Wolfgang Nelles**[1], **Jonas Neugebauer**[1],
Laura Ohrndorf[2], **Niclas Schaper**[1], **Sigrid Schubert**[2]
[1]University of Paderborn, Germany
{jsm, jonas.neugebauer}@uni-paderborn.de,
{wnelles, nschaper}@mail.uni-paderborn.de
[2]University of Siegen, Germany
{laura.ohrndorf, sigrid.schubert}@uni-siegen.de

Abstract: In the project MoKoM, which is funded by the German Research Foundation (DFG) from 2008 to 2012, a test instrument measuring students' competences in computer science was developed. This paper presents the results of an expert rating of the levels of students' competences done for the items of the instrument.
At first we will describe the difficulty-relevant features that were used for the evaluation. These were deduced from computer science, psychological and didactical findings and resources. Potentials and desiderata of this research method are discussed further on. Finally we will present our conclusions on the results and give an outlook on further steps.

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

1 Introduction

As a result of the on-going discussion about educational standards, competence models were developed for many subjects. They structure the particular learning field into different dimensions and sub-dimensions. In the project MoKoM a competence model with main focus on system comprehension and object-oriented modeling was developed.

This was done in several sub steps, which are shortly described in the following:

1. A theoretically derived competence model was created through the analysis of curricula and syllabi.
2. This model was refined with an empirical approach in terms of expert interviews, which were transcribed and analyzed.
3. On base of the empirically derived competence model a test instrument was created which was applied in a study with more than 500 students.
4. The evaluation results will be used to develop a competence level model that includes differentiated proficiency levels.

As a current research step, an expert rating for each item of our test instrument was done. The main research objectives for this are as follows:

1. To identify, describe, and examine empirically difficulty relevant features of the test items of a competence test of informatics competences
2. To develop a basis for the derivation of a competence level model

2 Difficulty-Relevant Features and Feature Levels

To identify and describe difficulty relevant features of the competency test we first defined difficulty relevant features of the competency test items. We derived those features from the literature concerning difficulty relevant features of competency 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. In this section we describe for each of the thirteen difficulty-relevant features their feature levels in computer science education (CSE).

2.1 Addressed knowledge categories

In a first step we analysed the structure of the knowledge dimension of the revised taxonomy for learning, teaching, and assessing from Anderson and Krathwohl (Anderson, Krathwohl, 2001) as a possible difficulty relevant feature of the test items. We assumed that the knowledge categories, A. Factual

Knowledge, B. Conceptual Knowledge, C. Procedural Knowledge, D. Metacognitive Knowledge and its usage for solving the test tasks, would differentiate between different levels of difficulty concerning our test items. So we derived the first difficulty-relevant feature with the following four feature levels:

- WI1: The successful solution of the task requires bare factual knowledge, and conceptual knowledge about the basic elements of computer science.
- WI2: The successful solution of the task requires basic and elaborated conceptual knowledge. The students recognize functional connections among basic elements of computer science within a bigger structure and task formulation.
- WI3: The successful solution of the task requires procedural knowledge. The students understand methods, rules and concepts to actively apply skills of computer science.
- WI4: The successful solution of the task requires meta-cognitive knowledge. The students know which cognitive requirements are needed for the available computer science task and how they can obtain and use the required contents to solve the task.

2.2 Cognitive process dimensions

In a second step 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 could assume that these addressed process categories, 1. Remember, 2. Understand, 3. Apply, 4. Analyze, 5. Evaluate and 6. Create, would also differentiate the levels of our test items. So we got the second difficulty-relevant feature with the following six feature levels:

- KP1: The successful solution of the task requires a memory performance. The students recall relevant knowledge contents from their memory.
- KP2: The successful solution of the task requires a comprehension perfor-mance. The students understand terms, concepts, and procedures of computer science and can explain, present and give examples for them.

- KP3: 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.
- KP4: 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.
- KP5: The successful solution of the task requires a rating (evaluation). The students are able to evaluate the suitability of concepts and procedures of computer science for the solution of the task.
- KP6: The successful solution of the task requires a creation. The students are able to develop a new computer science product by using concepts and procedures of computer science.

2.3 Cognitive combination and differentiation capacities

In a third step we applied findings of developmental psychology, e.g. of Piaget (Piaget, 1983). We could assume that these addressed combinations, *Reproduction, Application, Networked application*, would differentiate between different levels of difficulty concerning our test items. So we derived the third difficulty-relevant feature with the following three feature levels:

- KV1: Reproduction of computer science knowledge and application of single, elemental terms, concepts and procedures of computer science in close contexts (no cognitive combination capacities required).
- KV2: 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.
- KV3: 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).

2.4 Cognitive stress

In a fourth step we applied findings of cognitive psychology, e.g. of Jerome Bruner (Bruner, 1960). We assumed that these abstraction levels would differentiate the levels of difficulty concerning our test items. So we derived the fourth difficulty-relevant feature with the following three feature levels:

- KB1: For the successful solution of the task little, consecutive processing steps and no transfer performances are required: The degree of abstraction is very low.
- KB2: For the successful solution of the task many, consecutive processing steps and average transfer performances are required: The degree of abstraction is medium.
- KB3: For the successful solution of the task very many, consecutive processing steps and huge transfer performances are required: The degree of abstraction is very low.

2.5 Scope of tasks (necessary materials, reading effort and understanding)

In a fifth step we applied findings of educational psychology, e.g. of Benjamin Bloom (Bloom, Engelhart, Furst, Hill, Krathwohl, 1956). In this case we assumed that the addressed scope levels would differentiate between the levels of our test items.

So we derived the fifth difficulty-relevant feature with the following three feature levels:

- UM1: The task is formulated very short. No additional materials are required.
- UM2: The task is formulated extensive, only less material is required and the reading effort is kept within limits.
- UM3: The task is formulated very extensive. A high reading effort (quantitative and/or qualitative) and extensive materials (e.g. in the form of descriptions, APIs, overviews) are required for the solution of the task.

2.6 Inner vs. outer computational task formulation

In a sixth step we applied findings of didactics of informatics, e.g. of Deborah Seehorn (The CSTA Standards Task Force, 2011). We assumed that these addressed relation between inner and outer computational task formulation would differentiate the levels of competence concerning our test items. So we derived the sixth difficulty-relevant feature with the following two feature levels:

- IA1: For the successful processing of the task, *no* translation in an inner-computational format has to take place. The task is already present in a determined computational format.
- IA2: For the successful processing of the task, a translation in an inner-computational format has to take place. The task is already present in a determined computational format.

Aspects of demands of computer science
For aspects concerning special demands in computer science tasks we utilized dimension K4 of our competency model as a feature. This dimension covers the complexity of systems (Linck et al., 2013).

2.7 Number of components

The amount of components is a feature for the complexity of systems. This does apply to the understanding as well as the development of these. It is important to understand the effects and modes of operations in existing systems. The more components interact together, the more interactions have to be considered. When transferring this to the development of systems it is extended by the decision which components are needed and which tasks they fulfil.

2.8 Level of connectedness

As it might appear at first, the level of connectedness is not restricted to a concrete connection between systems (i.e. a network connection). It also refers to the connection of information used like the handling of data organized in a database. The more connectedness is required to fulfil the task, the more complex it is.

2.9 Stand-alone vs. distributed system

When dealing with distributed systems knowledge of the interaction of components and the connectedness of these is needed. This introduces a further level of abstraction since this involves different systems, which more or less multiplies the elements or parts that have to be considered.

2.10 Level of Human-Computer-Interaction (HCI)

The level of HCI needed is not necessarily given in the definition of the task, but can also be a part of the solution or the path to the solution. In this case learners should be able to decide which level is appropriate to fulfil the requirements. This also includes a decision based on the actual target group, e.g. it is depending on the user of software if it should be implemented as a simple command-line tool or a GUI.

2.11 Combinatorial complexity (mathematical)

The combinatorial complexity addresses the area of software tests with the creation of test cases as the main purpose. This is relevant not only for the actual testing process but also for the development of algorithms, where requirements have to be defined first and then verified. This can only be done by the development of suitable software tests.

2.12 Level of the necessary understanding of systems of computer science

This aspect describes the level of in-depth knowledge of computer science. It starts with a basic knowledge level, which can be build up through an everyday experience with computer science systems. It does not require a lengthy learning process. This aspect transitions through an interim level up to the need of fundamental ideas and concepts in computer science education. Furthermore for these tasks an independent evaluation of the system is needed.

2.13 Level of the necessary modelling competence of computer science

Computer science tasks often require modelling skills, which are covered by this difficulty feature. The feature varies from the basic illustration of tasks with a pseudo code to a complex transition with different UML-diagrams.

3 Research Methodology of Expert Rating

The experts were asked to rate each item of the competence test with reference to the thirteen difficulty features. Therefore a rating scheme and instruction was designed.

Furthermore, to conduct the expert ratings the measurement instrument was split into four parts of roughly equal size. To test the rating process one of these parts was used in a preliminary rating with hessian computer science teachers in the course of a teachers' workshop. The discussions during this test resulted in the addition of a "not relevant" rating level for all features, since the teachers thought some features inapplicable for some of the items. Each of the four instrument parts – including solutions for all 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 ratings were considered final, 2. Items with big differences for one or more features were transferred to a new rating booklet and 3. Every other rating was discussed within the project group to decide upon a final rating. A "big" difference was considered to be a substantial disagreement in the experts opinion, e.g. one expert rated the feature SG "not relevant" (SG0) for an item while the other saw the need to use high levels of modelling skills (SG3) to solve it. An example for a small difference would be the differentiation between "high levels of modelling activities" and "medium levels of modelling intensity".

After that the new rating booklet was given to two new experts together with the ratings of both previous experts. Then they were asked to go through the same process as the other experts to rate the items. Though they had the two previous ratings for each feature available for orientation, they could rate each item independently from them. The results from this second rating were compared in the same way as explained above. This time all differences in the new ratings were discussed by the members of the project team in order to decide upon a final rating for each item and feature.

The group was composed of seven researchers with background in computer science, computer science education and psychology. Since the group had to thoroughly discuss every feature, the process was done in two sittings. Afterwards each item had been assigned to a distinct rating level for each feature.

4 Results of the Expert Rating

The rating process resulted in a classification of 74 items concerning each of the de-scribed features. The rating levels for each feature were coded as increasing numbers, e.g. coding WI1 as 1 and WI2 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. Thus, the variables are considered to have an ordinal scale. Though this presumption does not necessarily have to be true, the order will be reviewed through the analysis of the rating data. This was done using descriptive and explorative methods to determine the relevant features that influence the item difficulty.

Comparing the number of times a feature was rated as "not relevant" for an item implies that the experts hold some features to be less useful in determining the difficulty of an item. Especially the features derived from the fourth competence dimension *K4 Dealing with system Complexity* were mostly considered to be inapplicable by the experts. The number of times each feature was deemed not relevant can be seen in figure 1. Interestingly the two features not derived from K4 with the highest number of "not relevant" ratings were *inner vs. outer computational task formulation* (IA) and the degree of necessary modelling competence (SG).

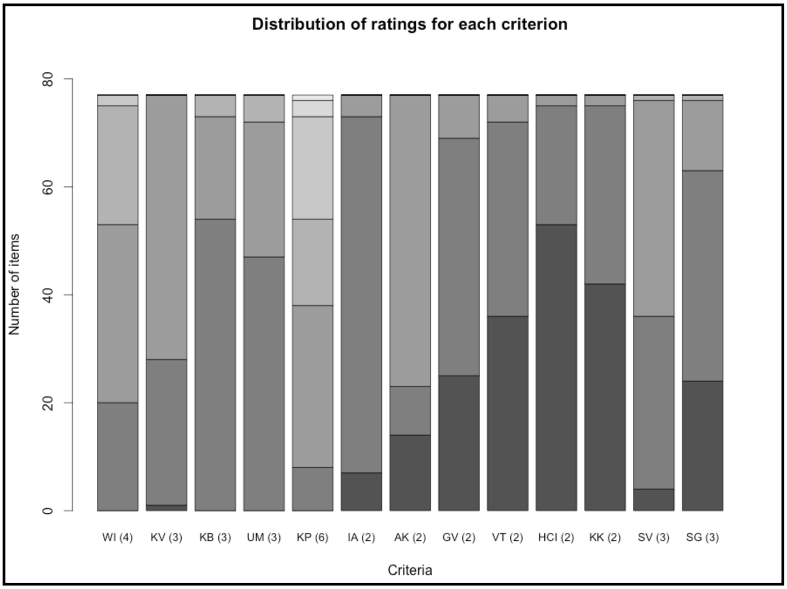

Figure 1: Each feature was rated for 77 items in two to six categories (in brackets) plus the "not relevant" (dark grey) rating.

The ratings for the feature IA suggest that the differentiation between inner and outer computational formats is not that easy in computer science tasks. The experts rated the majority of items as already being in an inner computational format. Judging by this, the feature might not be well suited to differentiate item difficulties. The SG feature was derived from the competence dimension *K3 System Development*, which represents an important part of the competence model. The experts saw no relevance of this feature for 24 items, over 30 % of the instrument. This actually could be expected, since each item was designed with exactly one competence profile in mind. Looking at the items with a relevant SG rating, they include almost all of the items designed for K3. Therefore the rating suggests that the test items were well constructed with regards to their competence profiles. On the other hand though, this raises the question why the feature SV, derived from *K2 System Comprehension*, was considered relevant for all but 4 items, since the items intended to measure these competences were constructed the same way as those for K3. An explanation for this is the need to comprehend system functions on an external and internal level, before being able to design and construct such systems. This is why items that refer to K3 often times also require some form of system comprehension. Figure 1 shows the number of ratings for each category of each feature.

The overall difficulty of the test instrument was subjectively rated by the experts with a mean of 4.2 on a ten-point scale. The distribution of difficulty estimates shows a tendency to the lower ratings, suggesting that the overall item pool might be marginally too easy (see figure 2). Ideally the difficulty ratings would be distributed normally, showing the most items in the medium difficulty range and an equal amount of easy and hard items. Though these ratings are subjective estimates by the experts they substantially correlate with the estimates from the IRT analysis ($r(72) = .553$, $p < .001$) and thus can be seen as an indicator for the validity of the expert ratings.

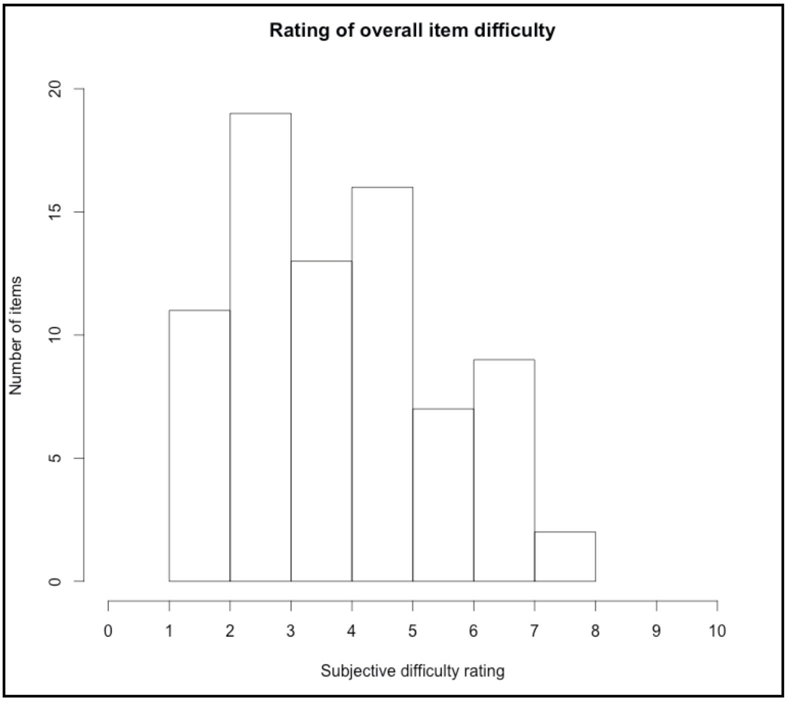

Figure 2: Histogram of subjective difficulty ratings on a ten-point scale.

4.1 Regression analysis

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) (Moosbrugger, 2008; Rost, 2004). The utilization of the IRT allowed for the application of a matrix design, in which not all test subjects have to work on every item (Hartig, Jude, Wagner, 2008). The test instrument was partitioned into six booklets with only parts of the tasks. All together the booklets were distributed to 538 computer science students in upper German high school education. 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 estimated parameters had a mean of -3.405 and standard deviation of 1.25.

To be able to use regression methods each ordinal variable of *n* levels had to be dummy coded into *n-1* dichotomous variables. Each dummy variable *i* would be 1 if the ordinal variable had the value *I*. The “not relevant” rating was coded as all dummy variables being 0. The dummy coding for the feature IA can be seen in table 1.

Table 1: Dummy coding for IA

IA	IA1	IA2
0	0	0
1	1	0
2	0	1

Since for some features one of the rating levels never was assigned to any item (e.g. "not relevant" for WI) the number of levels for those effectively was reduced by one. Thus, for these features another rating level had to be omitted from the dummy coding. The resulting 32 variables were used as the explanatory variables in a linear regression analysis with the difficulty estimate from the IRT analysis as the dependent variable (Hartig, 2007; Moosbrugger, 2008; Schaper, Ulbricht, Hochholdinger, 2008; Watermann, Klieme, 2002).

To evaluate the regression model the coefficient of determination can be examined. A value of 1.0 indicates that the item difficulty is completely explained by the analysed features, a value of 0.0 means that there is no link between the features and the empirical item parameters (Bortz, Schuster, 2010; Hartig, 2007). The analysed features significantly predict about 71 % of the differences in the item difficulties ($R^2 = 0.717$, $F(32,42) = 3.241$, $p < .001$). Though this is a good result, due to the high number of explanatory variables, this value might be overestimated. The adjusted R^2 takes the number of variables into account and takes a value of $R^2adj = 0.496$. Table 2 shows the regression coefficients, t-values and significance for the regression model.

The significance of the results is low for most of the features. The rating levels with the most significant influence on the item difficulty are AK2, HCI2 and SV3, with the last having the most substantial impact on the difficulty, increasing it by $b = 6.37$ points if the third level of the feature SV was assigned to an item (Hartig, 2007). The number of assignments for all three rating levels was very low (9, 2 and 1 times) respectively, but the features might still be valuable to differentiate item difficulties. The features AK and HCI stem from the competence dimension K4. As mentioned above, those features were considered "not relevant" for large parts of the items. Despite this, they still seem to be relevant for the estimation of item difficulties.

Table 2: Results of the regression analysis

	First regression model			**Second regression model**		
	b	*t*	*p*	*b*	*t*	*p*
Constant	-6,470	-4,746	< .001	-6,574	-5,265	< .001
WI2	-0,358	-0,926	0,3598	-0,274	-0,765	0,4482
WI3	-0,232	-0,416	0,6797	-0,143	-0,283	0,7786
WI4	*-3,732*	*-2,626*	*0,0121*	*-3,792*	*-2,827*	*0,0069*
KV1	1,647	1,406	0,1673	1,892	1,796	0,0791
KV2	1,218	1,074	0,2890	1,390	1,368	0,1780
KB2	0,403	0,857	0,3966	0,314	0,753	0,4554
KB3	-0,196	-0,152	0,8796	-	-	-
UM2	-0,511	-1,385	0,1735	-0,466	-1,450	0,1538
UM3	0,682	0,887	0,3801	0,576	0,944	0,3503
KP2	*1,134*	*2,514*	*0,0160*	1,138	2,658	0,0108
KP3	*1,310*	*2,316*	*0,0256*	*1,433*	*2,733*	*0,0089*
KP4	0,599	1,079	0,2868	0,601	1,209	0,2327
KP5	-0,044	-0,048	0,9621	0,050	0,058	0,9540
KP6	*2,410*	*2,052*	*0,0466*	2,360	2,127	0,0388
IA1	0,077	0,16	0,8737	-	-	-
IA2	0,633	0,745	0,4607	-	-	-
AK1	0,130	0,301	0,7651	-	-	-
AK2	***2,899***	***3,59***	***0,0009***	***2,927***	***4,088***	***0,0002***
GV1	-0,010	-0,023	0,9816	-	-	-
GV2	*-1,747*	*-2,162*	*0,0365*	-1,853	-2,648	0,0111
VT1	0,740	1,684	0,0997	0,705	1,917	0,0615
VT2	*-1,494*	*-2,615*	*0,0124*	-1,567	-3,024	0,0041
HCI1	***-1,916***	***-4,229***	***0,0001***	***-1,746***	***-4,360***	***0,0001***
HCI2	***-4,797***	***-3,969***	***0,0003***	***-4,411***	***-4,076***	***0,0002***
KK1	0,499	1,48	0,1466	0,461	1,717	0,0927
KK2	0,877	1,153	0,2554	0,889	1,247	0,2187
SV1	0,220	0,377	0,7083	0,187	0,338	0,7369
SV2	0,839	1,392	0,1714	0,863	1,584	0,1200
SV3	***6,376***	***3,75***	***0,0005***	***6,170***	***3,904***	***0,0003***
SG1	0,542	1,67	0,1026	0,515	1,715	0,0931
SG2	0,820	1,345	0,1859	0,888	1,811	0,0766
SG3	2,020	1,03	0,3089	2,278	1,330	0,1901

As a consequence from the results, the regression model was modified to get a minimal model that adequately could predict item difficulties. Features with low significance and low impact on the item difficulty can be dropped from the model. Furthermore, significant differences in the rating levels can be assessed by a one-way analysis of variance.

For this reason, the feature IA was removed. Both rating levels have low significance and the regressions coefficient of IA1 is fairly (IA1: *b* = .077, *r*(42) = .16, *p* = 0.87; IA2: *b* = .633, *r*(42) = .75, *p* = .46). Additionally, no significant differences between the three rating levels could be assessed. The rating levels of the feature KB showed a significant difference between KB1 and KB2, but not between KB2 and KB3. For this reason, the upper two rating levels were combined. Though the feature AK seems to differentiate well on the level AK2, the difference between AK0 and AK1 is not significant, which lead to the combination of those two rating levels. The distinction between "no", "few" and "many" system components can be reduced to "few or none" and "many" components. The same is true for the feature GV, now only differentiating between "low or none degree of connectedness" and "large degree of connectedness". Figure 3 shows the boxplots for the mentioned features.

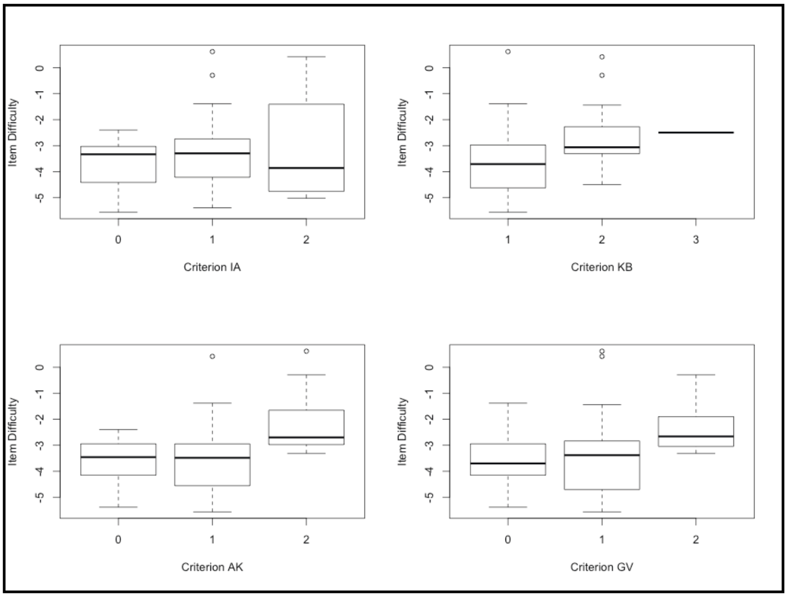

Figure 3: Boxplots for the features IA, KB, AK and GV

The newly evaluated regression model is still significant with a slightly lower coefficient of determination (R^2 = 0.707, *F*(27,46) = 4.114, *p* < .001) and an increased R^2adj = .535. To get to a minimal model with only significant rating levels, the insignificant variables can be stepwise removed and the model revaluated. By doing so, the features AK, VT, HCI, SV and SG remained in a significant model (R^2 = 0.48, F(8, 65) = 7.504, p < .001). These features can explain about 48 % of the item difficulties.

5 Conclusion

The analysis of the results of the expert ratings shows that most of the variance in item difficulties can be explained by the selected features. By reducing the features to the most significant ones, item difficulties can still be predicted an amount of 48 % variance determination. To allow for a feature-oriented interpretation of the IRT results, the next step will be to use the significant features to calculate the expected difficulty of items, rated with certain combinations of the features. These combinations will define appropriate thresholds between the proficiency levels (Beaton, Allen, 1992; Hartig, 2007; Schaper et al., 2008). The selection of suitable combinations of the features has to be based on theoretical and empirical sound decisions. For example the proficiency levels should be appropriately spaced and include items that define them, by satisfying the selected features. Moreover, the features should be useful to give meaningful explanations of the expected abilities in each proficiency level (Hartig, 2007; Watermann, Klieme, 2002).

If the a-priori rating of items yields no appropriate features to reasonably explain the difficulty of the items, it might be necessary to utilize post-hoc analysis methods used in other large-scale assessments like TIMSS III and PISA 2000 (Helmke, Hosenfeld, 2004; Schaper et al., 2008). With this method, distinctive items, characterized by certain thresholds in the item difficulties, are analysed for features that can be used to describe the proficiency levels. This approach has the disadvantage though, that the description for each level is dependent on the items used in the competence assessment.

References

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.

Beaton, A. E., Allen, N. L. (1992). Interpreting Scales Through Scale Anchoring. *Journal of Educational and Behavioral Statistics, 17*(2), pp. 191–204. doi:10.3102/10769986017002191

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., Krathwohl, D. R. (1956). *Taxonomy of educational objectives: Handbook I: Cognitive domain*. (Bloom, B. S., Ed.). New York: David McKay.

Bortz, J., Schuster, C. (2010). *Statistik für Human-und Sozialwissenschaftler. Lehrbuch mit Online-Materialien* (7 ed.). Berlin, Heidelberg, New York: Springer.

Bruner, J. S. (1960). *The Process of Education*. Cambridge, MA: Harvard University Press.

Hartig, J. (2007). Skalierung und Definition von Kompetenzniveaus. In Klieme, E., Beck, B. (Eds.), *Sprachliche Kompetenzen. Konzepte und Messung. DESI-Studie (Deutsch Englisch Schülerleistungen International)* (pp. 83–99). Weinheim: Beltz.

Hartig, J., Jude, N., Wagner, W. (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). Weinheim, Basel: Beltz.

Helmke, A., Hosenfeld, I. (2004). Vergleichsarbeiten – Standards – Kompetenzstufen: Begriffliche Klärung und Perspektiven. In Wosnitza, M., Frey, A., Jäger, R. S. (Eds.), *Lernprozess, Lernumgebung und Lerndiagnostik Wissenschaftliche Beiträge zum Lernen im 21. Jahrhundert* (pp. 56–75). Landau: Verlag Empirische Pädagogik.

Linck, B. et al. Competence model for informatics modelling and system comprehension. In *Proceedings of the 4th global engineering education conference, IEEE EDUCON 2013*. pp. 85–93, Berlin (2013).

Moosbrugger, H. (2008). Item-Response-Theorie (IRT). In Moosbrugger, H., Kelava, A. (Eds.), *Testtheorie und Fragebogenkonstruktion* (pp. 215–259). Berlin, Heidelberg: Springer. doi:10.1007/978-3-540-71635-8_10

Piaget, J. (1983). Piaget's Theory. In P. Mussen (Ed.), *Handbook of Child Psychology* (Vol. 1). New York: Wiley.

Rost, J. (2004). *Lehrbuch Testtheorie–Testkonstruktion* (2nd ed.). Bern, Göttingen, Toronto, Seattle: Huber.

Schaper, N., Ulbricht, T., Hochholdinger, S. (2008). Zusammenhang von Anforderungsmerkmalen und Schwierigkeitsparametern der MT21-Items. In Blömeke, S., Kaiser, G., Lehmann, R. (Eds.), *Professionelle Kompetenz angehender Lehrerinnen* *und Lehrer: Wissen, Überzeugungen und Lerngelegenheiten deutscher Mathematikstudierender und -referendare – Erste Ergebnisse zur Wirksamkeit der Lehrerausbildung* (pp. 453–480). Waxmann Verlag.

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

Watermann, R., Klieme, E. (2002). Reporting Results of Large-Scale Assessment in Psychologically and Educationally Meaningful Terms. *European Journal of Psychological Assessment, 18*(3), pp. 190–203. doi:10.1027//1015-5759.18.3.190

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.

Wolfgang Nelles is psychologist (Diplom-Psychologe) and his research interests concern Work- and Organizational Psychology. Since 2008 he is employed as a research assistant at the University of Paderborn.

Jonas Neugebauer received his first state exam for secondary education in computer science and mathematics from the University of Paderborn, Germany (2011). Since April 2011 he is employed as a research assistant at the Computer Science Education Group at the University of Paderborn.

Laura Ohrndorf received her diploma in computer science (2011) from the University Duisburg-Essen, Germany. Since July 2011 she is a research assistant at the University of Siegen. Her main research interests concern misconceptions in computer science.

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".

Copyright