Measurement of pedagogical content knowledge: students' knowledge and conceptions

Laura Ohrndorf, Sigrid Schubert

Didactics of Informatics and E-Learning
University of Siegen

November 13th 2013, WiPSCE 2013

Introduction - KUI project

- Competency Model Research
- Funded by the German Federal Ministry of Education and Research from 2012 to 2015
- Interdisciplinary project:
  - Didactics of Informatics: University of Siegen, Paderborn and TU München
  - Organizational psychology: University of Paderborn

UNIVERSITÄT SIEGEN

Table of contents

UNIVERSITÄT SIEGEN

Process in the KUI project (overview)

Short overview of the single steps:

- Development of a theoretically derived competency model (analysis of curricula)
- Empirical approach to refine, supplement and rectify the theoretical competency model by expert interviews
- Creation of a test instrument to measure competencies and assessment

Here: research on students' errors and conceptions in computer science and development of test items as a part of the project.

Research Background

Our experiences in computer science teachers' education:
Students are often not familiar with learners' conceptions or their way of thinking:

- They have problems to design learning arrangements and tasks with respect to students' cognition
- They are overwhelmed by real class situation, where they have to react in unexpected situations
- They don't know about common errors and misconceptions

What are errors and misconceptions?

Error:

- Wrong answer to a question/task, which is based on wrong, incomplete or missing knowledge

Misconceptions:

- Knowledge created based on outer-school experiences
- Not necessarily wrong, but might be incomplete or from a very basic point of view

Both are "systematic": i.e. the student will make the same error again.

Why should they be considered?

- Teachers' reactions on errors are essential for the learning process: not just "yes" or "no".
- Creation of a positive error culture [Oser et. al.]: room for making mistakes and the possibility to learn from them
- One of the most important factors for a successful learning [e.g. Hattie and Timperley]

Conclusion: teachers need to develop student's cognition in order to understand the thinking process of their students and to help them.

UNIVERSITÄT SIEGEN

Research interest and questions 1/2

- Research interest in teachers' knowledge about students' errors and misconcpetions has grown in the last years
- Several empirical studies in science subjects (e.g. COACTIV, ProwiN), mainly as a part of PCK tests
- Mentioned in many German curricula for computer science teacher education

Nevertheless: only very few research results on errors and misconceptions in computer science

Research interest and questions 2/2

Research questions:

- How and when does the competency of students' cognition develop?
- Does it correlate with the experience of a teacher?
- Do (main subject) computer science students have this competency?
- ...

Our goal: create a test instrument to measure teachers' knowledge about students' knowledge and conceptions

Characteristics in computer science

Students' engagement in computer science topics in their free time can lead to wrong or incomplete concepts:

- everyday use of computers, software, internet ... ("Google is the internet")
- chatting, writing e-mails
- programming
- ...

Furthermore: of course there are also general misconceptions (e.g. class vs. object).

UNIVERSITÄT SIEGEN

Measurement categories

We use three categories to test the knowledge on students' knowledge and conceptions (based on the categorization in the COACTIV project)

- PCK task: potentials and problems of computer-science tasks
- PCK student: familiarity with the conceptions and misconceptions of learners
- PCK instruction: subject-specific instructional strategies

Goal of this project: 7 test items PCK student + 2 test items PCK task.

Computer science tasks

- Many different types of tasks: e.g. drawing class diagrams, describing the working of hardware, problem-solving, programming, developing algorithms ...
- Tasks often involve several steps to come to a solution: reading and understanding of a problem description, connecting knowledge, application of knowledge ...

What are good tasks for test-items?

UNIVERSITÄT SIEGEN

Development of test items

General requirements:

- clear wording
- widely taught topics and terms
- realistic settings

Use of tasks already used for students' assessment

- MoKoM test intrument used for the assessment of 650 German students
- access to students' answers and results

Evaluation of test items

Problems that might occur in the evaluation of the test items:

- Listing of all correct answers is not possible: careful decision of wrong/right is needed
- Number/scope of answers: requiring a concrete amount of answers ("give two examples") and design as open test with no time limit
- Evaluation by experts in computer science education needed

UNIVERSITÄT SIEGEN

Conduction of the test

- Implementation in online test-tool (Limesurvey)
  - easier application and interpretation
  - less effort for test-takers (no need to send documents back)
- Application in computer science education courses
- Large interest of teachers: broad feedback after first introduction

UNIVERSITÄT SIEGEN

Example test item

Disclaimer:

- Common belief among empirical researchers: "never ever show your test items"
- Items can not be used after publishing
- There is always something to criticize

But:

- The first question is always: "Can you show a test item?"
- Feedback is welcome

Example item: PCK student 1/2

Students' task: *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 pseudocode. Decide which of the two algorithms is better regarding the running time.*

Correct	Algorithm

	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

UNIVERSITÄT SIEGEN

Example item: PCK student 2/2

Correct	Algorithm

	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

Teachers' task: Give at least two reasons why a student could choose the wrong answer.

Answers

Wrong answers that were given by our students:

- Algorithm 2 uses division, which is more time-consuming for the processor
- Algorithm 2 only works for odd-numbered tasks because the if-statement only works with them
- Algorithm 2 uses three mathematical operations, algorithm 1 only two.

UNIVERSITÄT SIEGEN

Current state and further research

- Creation of testitems to measure teachers' competencies of students' cognition
- These will be part of the KUI instrument
- Pretest with computer science education students will follow in December 2013
- Final assessment in 2014

Thank you for your attention!

Contact: laura.ohrndorf@uni-siegen.de