Better Use Pedagogical Content Knowledge for Threshold Concept Research in Computer Science? We Don't Think So.

Bert Zwaneveld (Open University NL)

Jacob Perrenet (Eindhoven University of Technology)

Roel Bloo (Eindhoven University of Technology)

Overview

- Who we are (Jacob)
- Our reasons for threshold research (Jacob)
- Our methods (Jacob)
- Our results (Bert)
- Conclusion & discussion (Bert & Jacob)

Why this research?

- Interest in the threshold concept; some experience with it in educational context.
- No experience with it in research context, but much experience in research in mathematics and computer science education
- Triggered by Shinner-Kennedy and Fincher's criticism on the threshold methodology using retrospective methods and their advice to use the PCK (Pedagogical Content Knowledge) method, but individually in stead of groupwise
- PCK method is asking a group of teachers for key concepts, students difficulties and ways to overcome these difficulties → Concept Representations (CoRe's)
- We met Fincher before and we met PCK before: we are not convinced.

Shinner-Kennedy & Fincher's criticism on threshold methodology + our reaction

- Retrospective methods are unreliable because of:
  - Hindsight bias
  - Emotional load
- Alternative: individual PCK

- Arguments for hindsight bias and emotional load have little relevancy
- PCK from teachers is not better: time consuming and indirect
- PCK is less valid for tertiary education
- So, go on with retrospection!
- We believe in the fruitfulness of the threshold concept and developed our own method based on retrospection: asking students on paper to describe concepts, asking their teachers about their students' concepts, and compare.

Context and Method

- BSc's programme Computer Science (CS) at Eindhoven
- At the end of the programme, a series of reflection assignments (digital)

- Students (about 60) were asked to write down one or more threshold concepts from their experience and to indicate the applicability of the five characteristics
- First the *threshold concept* was defined and explained (Ray & Land) and the five characteristics were given:

  Transformative / Irreversible / Integrative /

  Alien or anti-intuitive / Bounded
- Also four examples from various non-CS beta domains were given

Method (continued)

- We asked their teachers (about 20):

Which three threshold concepts do you think students mentioned most?

- Teachers got the same definition, explanation, characteristics and examples
- Paper-and-pencil task at an “Education Day”
- We analysed the data from both groups
- We compared the data with relevant results from literature

Research questions

- *Which threshold concepts do students mention?*
- *Are these concepts similar to what has been reported before?*
- *What is the applicability of the five characteristics?*
- *Which concepts do teachers think that students mention most?*
- *What are the similarities and differences between actual students' concepts and students' concepts according to teachers?*

Results: an example a student wrote

One of the threshold concepts in CS is ‘object’. For the starting programmer hard to understand. One of the first assignments was ‘Scanner’, and I use the syntax of the example given during a lecture. But I don’t really understand I was doing. It was *strange* that something simple as a scanner has such a complicated syntax. (**alien**) Only at the end of my first year I fully understand the concept, during an assignment Pentomino (a puzzle). Then I defined objects/classes myself. The syntax was not any longer strange. During that year *my view* on this subject *was dramatically changed*. (**transformative**) I started to *connect* concepts. (**integrative**) Through this assignment a lot of things became fully clear: f.i. equals() vs = (equal sign), pointers. (**integrative**) ‘Object’ is a concept that is *very typical for CS* (and not for the related domain of mathematics). (**bounded**)

Students' results

- Almost every student mentioned 1 - 3 concepts
- One student mentioned none
- Nine concepts without explanation asked for are omitted

number of students	59
total number of concepts	101
total number of *different* concepts	51
average number of threshold concepts per student	1.7

Results: students threshold concepts linked to the five characteristics

Characteristics	*applicability*
transformative	88%
irreversible	72%
integrative	64%
alien, anti-intuitive	78%
bounded	55%

Results: the variation in the students' list

appearance of specific threshold concepts in the students' list of the 51 different concepts	*percentage*
10 times or more mentioned	4
2 – 9 times mentioned	27
1 time mentioned	69

Results: the variation in the students' list
(59 students)

threshold concepts	*fr*
object-oriented	17
logics	12
undecidability	5
algorithm	3
induction	3
abstraction/ structure layers	3
complexity	3
programming	3

threshold concept	*fr*
programming methods	3
recursion	2
halting problem	2
pointers	2
datastructures	2
race-conditions	2
functional programming	2
asymptotic running time	2
other, unique concepts	35

Results: the variation in the students' list: the unique concepts

matrices	sequential execution
objects	rational/real; countable/uncountable
implication	operating systems
hardware	software specification
software-engineering	programming paradigm
static	contract by method/class
architecture	tree
join operation	your design is never perfect
code- and commenting standards	research results are never true
graphics operations	projects
graph problems	efficiency
classifying objects in sets	test driven development
linear thinking	asynchronous communication
proof	fake security
analytic perspective	lambda-calculus
Open Graphics Library	capturing all cases
code-specification	currying
	dividing in modules/layers

Threshold concepts mentioned by Shinners-Kennedy&Fincher (students) & comparison

- object-oriented
- pointers
- agreement on the first concept: *object orientation* is on top in our list as well
- no agreement on the second concept, *pointers* is mentioned by our students, but it plays a minor role
- our second candidate would be *logic*, instead of *pointers*; o*bject-orientation* is often in experts' lists, *logic* not; neither 'our' is *undecidability*
- there is some overlap (*object-orientation*), but there are more differences

Teachers' results

- Almost all teachers mentioned 2 to 3 concepts
- One teacher mentioned none

number of teachers	18
total numbers of concepts	49
total number of *different* concepts	21
average number of threshold concepts per teacher	2.7

Results: the variation in the teachers' list

appearance of specific threshold concepts in the teachers' list of the 21 different concepts	*percentage*
6 or more times mentioned	10
2 - 5 times mentioned	28
1 time mentioned	62

Results: the variation in the teachers' list

threshold concepts	*fr*
recursion	9
complexity	6
object-oriented	5
undecidability	5
induction	4
logic	3
computability	2
variable	2
other, unique concepts	13

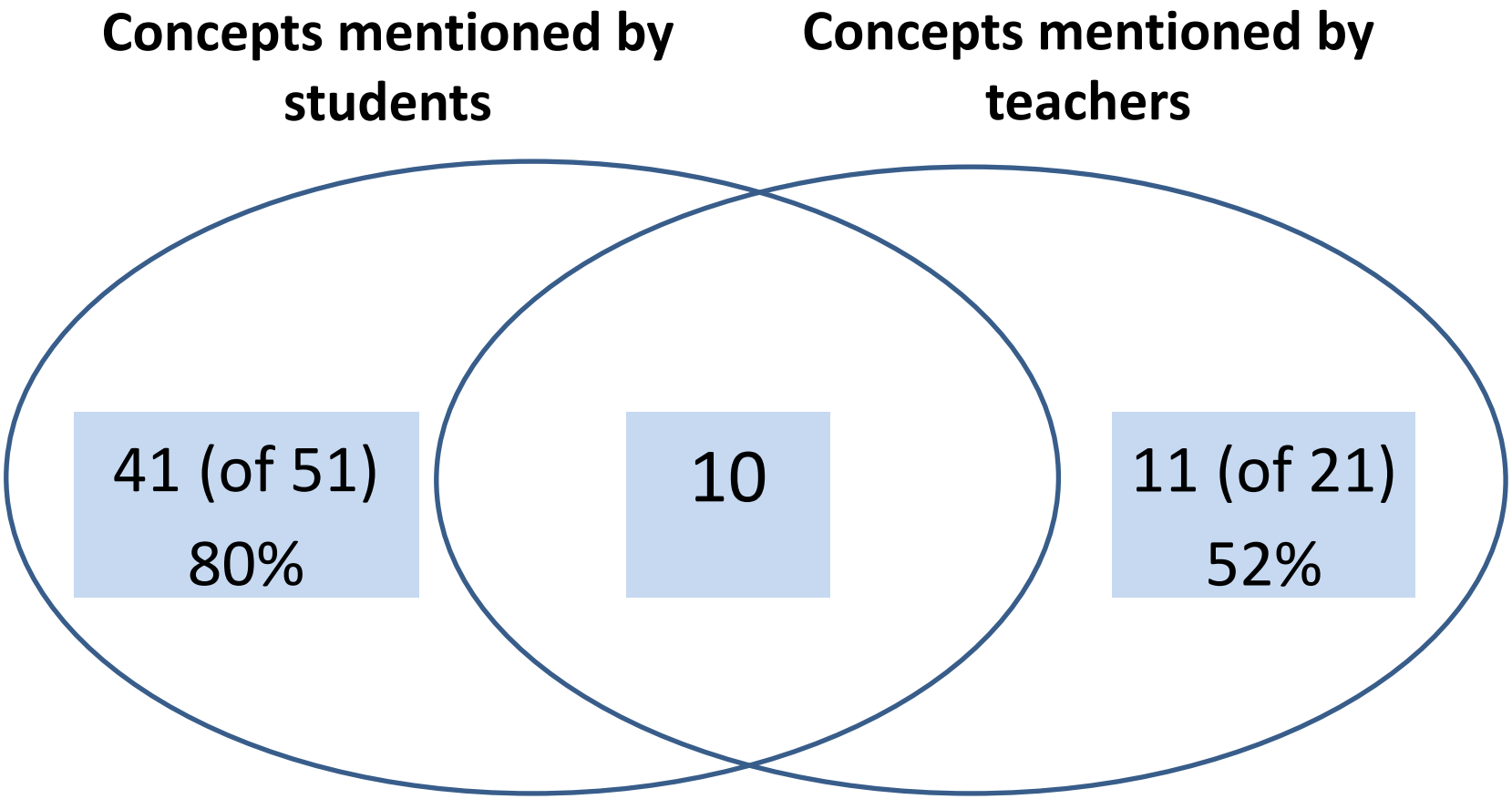
Results: the variation in the teachers' list: the unique ones

unique threshold concepts
physical representation of computers
concept of programming
no assignment in functional programming
pointers
Turing machine/mechanical concept of computers
(complete) axiomatization of constraints
invariant
interaction with stakeholders by Software Engineering Project
Prolog-like languages
(trap door) one-way functions asymmetric crypto
proof
testing covers minor part of all instances
programming correctness independent from programming language

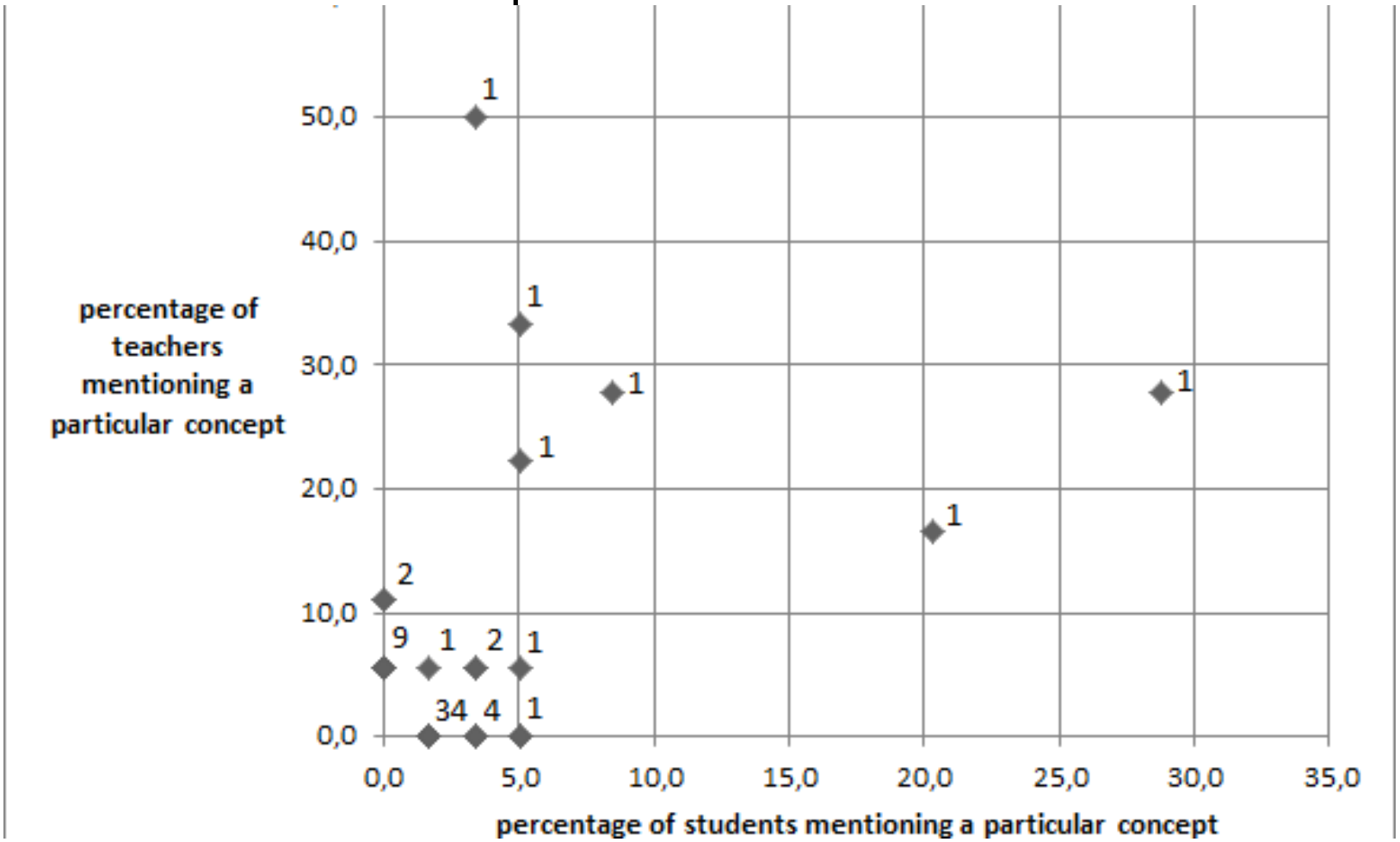
Comparison: students vs teachers

	students	*teachers*
number of students/teachers	59	18
total number of *different* concepts	51	21
overlapping number of concepts	10	

Results: students vs teachers numbers of (non-)overlapping different concepts



Percentage of students vs percentage of teachers mentioning a particular threshold concept



Point (20.3; 16.7) (1) represents that 1 concept (*logic*) is mentioned by 20.3% of the students and 16.7% of the teachers; point(3.4; 5.6) (2) represents that 2 concepts (*functional programming, pointers*) is mentioned by 3.4% of the students and 5.6% of the teachers

Conclusions

- Threshold concepts as described with five characteristics do exist in CS, but not all characteristics are equally relevant according to students
- No signs of hindsight bias or emotional load
- So, go on with retrospection! Individual PCK is no alternative
- Asking students for their threshold concepts using a *paper-and-pencil task* in combination with asking teachers for their students' threshold concepts produces useful results

Conclusions (continued)

- Few concepts are thresholds for many students
- Many concepts are thresholds for few students
- Students' concepts are only partly the same as threshold concepts reported before
- Teachers mention various students' threshold concepts
- Teachers' concepts are only partly the same as concepts by students

Discussion

- Categorizing threshold output is a hard task, because of variation in specificity/abstraction of response
- Curriculum innovations explain differences of our results with results reported before
- Personal study history explains between students' variation
- If there are only few top threshold concepts, what is the use for education?
  - Teachers should be aware of variety in individual difficulties
  - Asking for threshold concepts stimulates students' reflection on their learning

Ideas for further research

- Analysing CS curricula and their influence on threshold concepts
- Analysing specificity/abstraction levels of CS threshold concepts mentioned by students and teachers
- Analysing thresholds using this method in other disciplines (f.i. mathematics)
- Threshold concepts as a reflection task