

BACHELOR THESIS
COMPUTING SCIENCE



RADBOUD UNIVERSITY

**Perceptions of Dutch (Former) CS
Teachers regarding the Assessment
of Fundamentally New Learning
Objectives**

Author:

E.M.M. (Evelien) van Workum
s1002650
E.vanWorkum@student.ru.nl

First supervisor/assessor:

dr. J.E.W. (Sjaak) Smetsers
S.Smetsers@cs.ru.nl

Second supervisor:

R.E. (Renske) Weeda MSc
R.SmetsersWeeda@ru.nl

Second assessor:

prof. dr. E. (Erik) Barendsen
E.Barendsen@cs.ru.nl

June 26, 2020

Abstract

Contribution: From prior studies, it is known that paying attention to assessment of a new curriculum is very important to implement that curriculum successfully. Related studies have looked into the Pedagogical Content Knowledge of teachers in general, which includes their knowledge and beliefs about assessment, in the context of new science curricula. This paper presents the perceptions of (former) teachers on the assessment of the new (sub)domains in the core program of the recently reformed Dutch Computer Science curriculum specifically.

Background: Many teachers have difficulties with the assessment of student work and learning. To make the implementation of the reformed Dutch Computer Science curriculum more successful, investigating the perceptions of teachers on the assessment of the fundamentally new learning objectives in the curriculum is therefore worthwhile.

Research Question: How can Dutch (former) high school CS teachers be typified based on their perceptions about the assessment of the fundamentally new learning objectives in the reformed high school CS curriculum of the Netherlands?

Methodology: Six participants participated in this qualitative, exploratory study. The Repertory Grid Technique with additional member checking interviews was used to collect the data. This data was analyzed by means of performing a content analysis on the repertory grids and the member checking interviews, and a cluster analysis on the repertory grids.

Findings: Many opinions and expectations exist which can be used to typify (former) CS teachers. In this study however no clear typification could be found. One important finding was that many teachers find adapting questions difficult due to having to keep the same level of difficulty and think of a new, suitable context.

Index Terms: *assessment, computer science curriculum, informatics education, Professional Learning Community, repertory grid, secondary education*

Foreword

Acknowledgements

This study was carried out as a Bachelor Thesis of the program Computing Science at Radboud University Nijmegen. This research was supervised by Sjaak Smetsers and Renske Weeda. I would like to thank them for giving me the opportunity to perform a research which is not only of indirect importance, but also directly useful for the participants of the Professional Learning Community meetings related to this study and for the study Renske Weeda is performing (see section 1.1 'Context and overarching research'). Additionally, I would like to thank them for their advice and support throughout the entire process.

Furthermore, I would like to thank all participants of those related Professional Learning Community meetings who were all willing to spend (in some cases even several hours of) their time to help me with this research. I also would like to thank Erik Barendsen for his advice regarding for example the specific definitions of concepts and his feedback. Finally, I would like to thank my parents and my partner for their listening ear and their support.

Contents

1	Introduction	6
1.1	Context and overarching research	7
1.2	Development of research	8
1.3	Outline of paper	8
2	Theoretical framework	9
2.1	Professional Learning Community	9
2.2	The new CS curriculum	10
2.3	Assessment of the new curriculum	10
2.4	Repertory Grid Technique	11
2.4.1	The repertory grid	11
2.4.2	Linking of elements and constructs	12
2.4.3	Member checking the repertory grid	13
2.4.4	Analyzing repertory grid data	13
2.5	Pedagogical Content Knowledge	14
3	Aim of study	16
4	Methodology	18
4.1	Participants	18
4.2	General procedure	19
4.3	Collecting data	21
4.4	Analyzing data	22
5	Results	25
5.1	Collected data	26
5.2	Analysis: Characteristic perceptions	27
5.2.1	Most common characteristic perceptions	27
5.2.2	Reasons for opinions and expectations	29
5.3	Analysis: (Dis)similar perceptions of learning objectives	29
5.3.1	Dissimilarity of learning objective X	30
5.3.2	Similarity between learning objectives III and IX	30
5.3.3	Similarity between learning objectives III and VI	31
5.3.4	Relation between clusters and (sub)domains	31

5.3.5 Distinction between implementation and theory	31
6 Conclusions	33
7 Discussion	36
7.1 Reflection on conclusions	36
7.2 Reflection on findings in general	37
7.3 Reflection on decisions during data collection	38
7.4 Limitations and future directions	39
7.5 The original problem	41
A Repertory grid abbreviations	45
A.1 Elements	46
A.2 Constructs	47
B Development of research	48
B.1 Initial research plan	48
B.2 Interim research plan	49
B.3 Final research plan	49
B.4 Second main research question	50
B.4.1 Aim of study	50
B.4.2 Methodology	51
C Shared found perceptions	53
D In-depth explanation of the results per participant	55
D.1 Characteristic perceptions	55
D.1.1 Charles	55
D.1.2 David	57
D.1.3 James	58
D.1.4 John	60
D.1.5 Michael	61
D.1.6 William	64
D.2 (Dis)similar perceptions of learning objectives	66
D.2.1 Charles	66
D.2.2 David	67
D.2.3 James	67
D.2.4 John	68
D.2.5 Michael	69
D.2.6 William	70
D.3 Description of Focus cluster	70
D.3.1 Charles	70
D.3.2 David	72
D.3.3 James	74
D.3.4 John	76

D.3.5 Michael	78
D.3.6 William	80
E Repertory grids of participants	83
E.1 Raw repertory grids	83
E.2 Focus cluster	86
F Online questionnaire	90
G Semi-structured interviews	91
G.1 Interview questions used to elicit constructs	91
G.2 Interview about perceptions and developing assessment task .	91
G.3 Neat transcription of MemberChecking	94
G.3.1 Charles	94
G.3.2 David	99
G.3.3 James	106
G.3.4 John	117
G.3.5 Michael	122
G.3.6 William	128

Chapter 1

Introduction

On the First of August, 2019, the new, reformed high school curriculum for Computer Science (CS) has been implemented in the Netherlands (Ministerie van Onderwijs, 2016). At the time of this research, the Dutch high school CS teachers are thus still little experienced with teaching and assessing the fundamentally new learning objectives of this curriculum. On top of that, Computer Science teachers often have difficulties with the assessment of CS topics in general (Yadav et al., 2016). No study however could be found which investigates in-depth what the teachers have difficulties with; are the difficulties related to for example creating assessment tasks or judging them? Do the difficulties only exist for specific topics within Computer Science? Answers to these questions could not be found.

In this study, we will make a start with answering these questions by investigating the opinions and expectations of the teachers regarding the assessment of the fundamentally new learning objectives and how high school CS teachers can be typified based on these perceptions. Its answer will not only fill a part of the mentioned knowledge gap in scientific literature, but will also be useful for high school CS teachers and teacher educators in the Netherlands. By relating their own perceptions, experiences and difficulties (and, for teacher educators, those of their students) to the ones found in this study, they can become aware of, for example, knowledge gaps regarding this assessment of which they were not aware before. Additionally, by paying more attention to these knowledge gaps and difficulties shared by many teachers, the teachers are likely to have less difficulties with the assessment and will hopefully therefore also be able to deliver a(n) (even) higher quality regarding their assessment.

An important term in the mentioned research problem is the rather general concept 'perception'. Therefore, we will focus on two more specific aspects of perceptions. First of all, we will focus on the characteristic opinions and expectations - after all, the (former) teachers are still little experienced with the assessment of the fundamentally new learning

objectives - of the teachers regarding the assessment of the learning objectives. What is seen as a characteristic or outstanding perception is explained in section 4.4 'Analyzing data'. On top of that, we will focus on how, and based on which opinions and expectations the (former) teachers divide these learning objectives into categories. We will thus, in this research, identify the perceptions of the teachers by investigating their outstanding opinions and expectations as well as which learning objectives they feel to be (dis)similar regarding their own opinions and expectations of the assessment of these learning objectives.

To discover the opinions and expectations of the teachers, we will make use of the Repertory Grid Technique, because that technique is very suitable to elicit and understand individual perceptions (Curtis et al., 2008). This technique is, for example, also used by Henze, Van Driel, and Verloop (2007) to investigate the content of science teachers' personal knowledge regarding the teaching of a new science syllabus. How the Repertory Grid Technique works, is explained in section 2.4 'Repertory Grid Technique'. By investigating the perceptions of the teachers found with this technique, we will in the end be able to answer how Dutch high school CS teachers can be typified based on their perceptions regarding the assessment of the fundamentally new learning objectives in the core program of the recently reformed curriculum.

1.1 Context and overarching research

The current study takes place in the context of a Professional Learning Community (PLC). This PLC consists of four meetings about the assessment of, and developing assessment tasks for the fundamentally new learning objectives of the recently reformed Computer Science (CS) curriculum in the Netherlands. Several (former) CS teachers (including teacher educators) are participating in these meetings. The PLC is initiated by Renske Weeda and Martin Bruggink, and is supervised by Erik Barendsen. One of the objectives of the initiators of this PLC is to research how the perceptions of the teachers regarding the assessment of these learning objectives develop over time when participating in the PLC, and which factors contribute to this shift. Renske Weeda conducts this research and is supervised by Erik Barendsen.

The current study provides a baseline for the research of Renske Weeda; by taking a look at the perceptions of the participants at the start of the first PLC meeting, the baseline data and analysis for the larger study is gathered. The data collection and findings of the current study will be used in the larger study to see how the perceptions of the teachers participating in the PLC have developed. Because of this relation between our studies, and for her to get a clear idea of this conducted research, Renske

Weeda also contributed to this study. She contributed, for example, by creating the repertory grid together - she created and held the interview for identifying the constructs to give an example -, performing the pilot study for the questionnaire and holding a semi-structured interview, as explained in section 4 Methodology.

1.2 Development of research

In connection with the larger research, the research plan of this study had to be adjusted several times. Initially, the plan for this study was to look at the so-called Pedagogical Content Knowledge of the teachers (see section 2.5). Then, that plan changed into investigating how the software Rep Plus can be used to both find out the PCK of the teachers and to investigate a possible shift in their PCK. Finally, this focus changed into (RQ1) investigating the opinions and expectations of the (former) teachers (which is only one aspect of their PCK), as well as (RQ2) finding possible sources for a shift in these opinions and expectations by taking a look at how the (former) teachers develop assessment questions. This second research question could however not be executed entirely due to time constraints, and is therefore not discussed in the main content of this paper. A more elaborate explanation on the development of the research plan, and a short summary of what we have and would have done for that second research question can be found in appendix B.

1.3 Outline of paper

The rest of this paper is organized as follows. In the next section (2 'Theoretical Framework'), we will explain all concepts important for this study, including several short summaries of findings of prior studies. Consequently, in section 3 'Aim of study' we state the research question and subquestions of this study, as well as their relation with the purpose and important concepts of this study. Next, we will describe the methods used to perform this study in section 4 'Methodology'. The methodology is followed by section 5 'Results' containing the findings of this research. Then section 6 'Conclusions' follows in which the main findings and content of this paper are summarized. Lastly, an interpretation of the results in the lights of prior research, limitations of this study, and future research are described in section 7 'Discussion'.

Chapter 2

Theoretical framework

As described previously in the foreword and the introduction, the context of this study consists mainly of the assessment of the new CS curriculum for high school students in the Netherlands, and the meetings of a Professional Learning Community (PLC) about the development of assessment tasks for the domain Foundations of this new curriculum. Therefore, it is important to know what a PLC is, what the fundamentally new (sub)domains of the new CS curriculum comprise, and why paying attention to assessment is important when a new curriculum is implemented. Therefore, we will provide this background information in this chapter.

Furthermore, we will describe the Repertory Grid Technique and how this technique is used in this study, because we will use this technique to extract the participants' perceptions, or more specifically their opinions and expectations, about the learning objectives of the fundamentally new subdomains of the reformed high school CS curriculum.

This study would originally cover two research questions (see section B.4 'Second main research question'). To understand the theoretical and scientific relation between these research questions, we will also describe the concept Pedagogical Content Knowledge (PCK), its background and why the answers to the research questions together would give a description of important aspects of the participants' PCK regarding assessment.

2.1 Professional Learning Community

A Professional Learning Community (PLC) is defined as "a group of experts in a specialized field who work towards a common goal" (Turner, 2015). In the field of education, however, many different definitions of PLCs exist (e.g., Stoll and Louis, 2007; Turner, 2015). The general definition by Stoll and Louis (2007) fits with the idea of the PLC this study is about, namely

a group of teachers who share and critically question their practice in a reflective, collaborative, learning-oriented, growth-promoting way.

Other often mentioned characteristics of PLCs are the shared vision or goal, and action orientation (learning by doing) (Turner, 2015). Achieving this goal is not the only positive effect of a PLC; participating teachers also enhance among others their understanding of course content, higher satisfaction, and of course better education (Hord and (U.S.), 1997). PLCs have thus a positive impact on both students and teachers.

2.2 The new CS curriculum

As of the 1st of August 2019, the regulation corresponding to the reformed Computer Science curriculum for high schools in the Netherlands has come into effect (Ministerie van Onderwijs, 2016). The main differences between this reformed curriculum (College voor Toetsen en Examens, 2020) and the previous curriculum (College voor Toetsen en Examens, 2018) are mentioned in the "Handreiking SE Informatica" (Guide School exam Computer Science) of the national center of expertise for curriculum development SLO (SLO Nationaal expertisecentrum voor leerplanontwikkeling, 2020). The main difference is the focus of the curriculum; whereas the focus in the previous curriculum was on business automation and specific applications ('foundations') and also includes modern applications like programming and security (College voor Toetsen en Examens, 2020). Fundamentally new and part of the core program (thus compulsory to include) are thus the (sub)domains:

- domain B Foundations,
- domain D Programming, and
- subdomain A8 Designing and implementing (digital artefacts).

2.3 Assessment of the new curriculum

As mentioned in section 2.2 'The new CS curriculum', several (sub)domains and thus learning objectives in this reformed curriculum are fundamentally new. To successfully implement a new curriculum, paying attention to the assessment of that curriculum is very important (Grover and Pea, 2013). McDonald and McDonald (1999) also already mentioned that assessment is an essential part of the process for curriculum improvement. A lot of teachers however have difficulties assessing student work and learning (e.g., Yadav et al., 2016). These difficulties are not expected to be less in the Netherlands, as many CS teachers are not fully or not at all educated academically to teach CS (Tolboom, Krüger, and Grgurina, 2014). Being

able to adequately assess the new (sub)domains, and especially the new domain 'Foundations', is specifically important, because the students can get knowledge gaps without the teacher noticing if a teacher is not able to teach or assess the fundamental skills from this domain adequately.

2.4 Repertory Grid Technique

The Repertory Grid Technique (RGT) is a method for exploring (bipolar) personal construct systems. The technique is based on the Personal Construct Theory from Kelly (1955). A personal construct can be described as a personal theory or assumption of an individual constructed by their experiences (Edwards, McDonald, and Young, 2009). The RGT can thus be used to understand how an individual or group perceives different artifacts in their environment (Curtis et al., 2008). As such, the RGT can be used in education research to determine and describe personal criteria and beliefs of teachers (Herman, 1996). Not only can the technique be used to investigate people's perceptions at a specific moment, but also changes in perceptions over time can be elicited (Henze, Van Driel, and Verloop, 2007).

2.4.1 The repertory grid

The Repertory Grid Technique provides a structured format of how people perceive different artifacts, namely the repertory grid (Edwards, McDonald, and Young, 2009). As can be seen in Figure 2.1, the repertory grid is essentially a matrix in which each column represents an *artifact* and each row represents a bipolar *construct*. One label of the bipolar construct is shown on the left hand side of the matrix, the other on the right. The artifacts in the RGT are often called *elements*. In the repertory grids shown in this thesis, the elements are shown underneath the repertory grid. The constructs give an indication of how the different elements are perceived.

		Display Nonexisting person		
easy to assess	5 5 5	difficult to assess		
best to assess using a written test	1 2 2	best to assess through practical work		
best to assess with group work	1 2 4	best to assess individually		
		designing elements of a user interface		
		evaluating user interfaces of digital products via user research		
		describing and explaining the operation of user interfaces based on cognitive and biological models		

Figure 2.1: Example of a repertory grid with random ratings

The items within the repertory grid are numerical values from a rating scale. In our study, a (linear) rating scale from 1 to 5 is used. If an item has a rating of 1, the label on the left hand side of the matrix describes the corresponding element most accurately according to the participant(s);

a rating of 5 implies that the label on the right hand side is seen as the most accurate description of the corresponding element.

Three types of repertory grids exist, namely full repertory grids, partial repertory grids and fixed grids (Edwards, McDonald, and Young, 2009). The full and partial repertory grids are repertory grids where the participant identifies at least the constructs, implying that the constructs are likely to differ per participant. The difference between these types is that, in full repertory grids, the participant also identifies the elements whereas in partial repertory grids, the elements are identified by the researcher(s) (see Edwards, McDonald, and Young, 2009). Our study will make use of a fixed grid: a grid where the participant is supplied with both the elements and the constructs (Edwards, McDonald, and Young, 2009). As a consequence, the grids of the different participants are all the same and can thus easily be compared with each other.

The elements and constructs in a repertory grid can thus be identified by either a researcher(s) or each participant (Curtis et al., 2008). Both ways of identifying elements and constructs have their own (dis)advantages. Individual elicitation reduces less researcher bias, but takes much more time than identification by a researcher. Because of this duration, the elements and constructs in our study are elicited by a researcher. To reduce the researcher bias, the constructs used in this study were identified based on interviews and the fundamentally new (sub)domains from the reformed curriculum - as mentioned previously in Section 2.2 - were used as elements.

Several requirements exist which the constructs and elements must meet to be useful in the repertory grid. A clear overview of these requirements with an explanation can be found in Chapter 2 'Constructs and elements' of the book 'A manual for repertory grid technique' by Fransella, Bannister, and Bell (2004). These requirements include for example:

- all constructs should be explicitly bipolar,
- all elements should be representative of what is investigated, and
- all constructs should be relevant to all elements used.

2.4.2 Linking of elements and constructs

After the elements and constructs are identified, several ways exist to elicit the links - thus how the participant can indicate how he feels about an element regarding a particular construct - within the repertory grid (Curtis et al., 2008). One of the techniques which can be used is let the participants rate each of the elements on all constructs. In contrast to (for example) ranking the elements on a continuous scale, rating the elements implies that a semantic (and not a continuous, numerical) scale is used. This

method is seen by Curtis et al. (2008) as the best option to link elements with constructs. The rating of the elements (and thus the filling in of the repertory grid) can be done either during an interview or by using a questionnaire. The advantage of an interview is that the researcher receives more information. It does however take much more time than filling in a questionnaire. Using a questionnaire can thus be more practical than the interview for both the researcher and the participants due to these time constraints, but a questionnaire has the disadvantage that it gives (almost) no insight in the reason behind the ratings.

2.4.3 Member checking the repertory grid

After having elicited the elements, constructs and ratings of the repertory grids of all participants, the researcher has the possibility to perform a member check during the so-called post-interview stage (Curtis et al., 2008). Member checking is a technique to improve for example the validity of a study. In the case of the repertory grid technique, the researcher reviews the created repertory grid with the participant to confirm that the repertory grid correctly represents the perceptions of the participants. During the member check, the participant is allowed to change any ratings he or she believes to be incorrect; these ratings are treated as if they were the original ratings. Note that although member checking improves the validity of the study, the repertory grid still can contain ratings which are a misrepresentation of the perceptions of the participant as during the member checking only the main findings are reviewed.

2.4.4 Analyzing repertory grid data

Several ways exist to analyze repertory grids (Curtis et al., 2008). In our study, we will focus on the numerical evaluations of the elements on the constructs. An often used technique used for the analysis of the grid ratings is cluster analysis (Curtis et al., 2008). This technique can be used to analyze relationships between elements. By comparing the ratings of different constructs in the groupings, the reasoning behind the groupings can be investigated (Henze, Van Driel, and Verloop, 2007). Furthermore, the groupings of different repertory grids can be compared in order to investigate the similarities and differences between those different grids (Curtis et al., 2008; Henze, Van Driel, and Verloop, 2007).

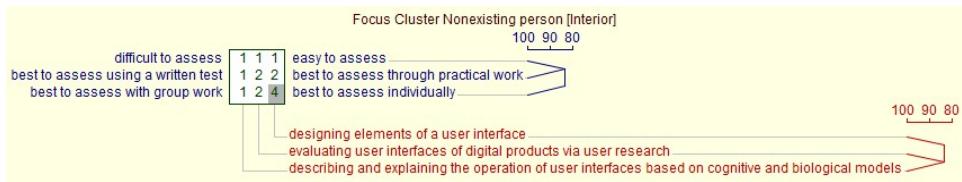


Figure 2.2: Example of the focus cluster of a repertory grid with random ratings

Figure 2.2 shows an example of a cluster analysis of a repertory grid. The dendograms on the right show how similarly the elements respectively constructs could be rated by a participant. In this specific case - with randomly assigned ratings - the elements about 'evaluating user interfaces of digital products' and 'describing and explaining the operation of user interfaces' are shown in a cluster, which implies that they are rated very similarly (> 80% similar). In such case, the conclusion can be drawn that this non-existing person perceives these elements to be more similar than the element 'designing elements of a user interface'.

Furthermore, one can understand the ground based on which these elements are grouped this way by taking a closer look at the ratings for these elements. In this case, one can see that the non-existing person perceived 'designing elements of a user interface' to be different from the other elements, because he feels that this element is better to assess individually whereas the other elements are seen by him as better to assess with group work. In a similar way, conclusions could be drawn about the constructs.

2.5 Pedagogical Content Knowledge

Pedagogical Content Knowledge was first presented by Shulman (1986). He defined PCK as 'the content knowledge that embodies the dimension of subject matter knowledge for teaching'. In this definition content knowledge is defined as 'the amount and organization of knowledge per se in the mind of the teacher'. In his work, he mentions three components of PCK: knowledge of the most regularly taught topics in one's subject area, knowledge of the understanding of learners of these topics, and the ways of representing and formulating a subject that make it comprehensible to others.

Even though the components of most PCK models' are based on the PCK model of Shulman (e.g., Grossman, 1990; Magnusson, Krajcik, and Borko, 1999; Tamir, 1988; Shulman, 1986), no consensus exist on the components of which PCK exists (Jing-Jing, 2014). In this study, we will use the definitions and components of PCK from the PCK model of Magnusson, Krajcik, and Borko (1999). In this PCK model, PCK is defined as "a teacher's understanding of how to help students understand specific subject matter".

The model is indirectly based on the PCK model of Shulman, and consists of the same, slightly modified components with two additional components added. The components in the PCK model of Magnusson are:

- (M1) orientations towards science teaching;
- (M2) knowledge and beliefs about science curriculum;
- (M3) knowledge and beliefs about students' understanding of specific science topics;
- (M4) knowledge and beliefs about assessment in science;
- (M5) knowledge and beliefs about instructional strategies for teaching science.

As can be seen, the two additional components in this PCK model - compared to the model of Shulman - are 'knowledge and beliefs about science curriculum' (M2) and 'knowledge and beliefs about assessment in science' (M4). By including the knowledge and beliefs about the curriculum, Magnusson, Krajcik, and Borko followed the PCK model of Grossman (1990). The way they defined their components is also very similar to the PCK' conceptualization of Grossman (Magnusson, Krajcik, and Borko, 1999; Grossman, 1990).

Several ways exist to elicit the PCK of a teacher which all have their advantages as well as disadvantages (Magnusson, Krajcik, and Borko, 1999). The most important aspect of choosing a technique to elicit a teacher's PCK is that the technique should focus on all four aspects of PCK, namely what a teacher knows, what a teacher does, what a teacher believes, and what the reasons for the teacher's actions are (Magnusson, Krajcik, and Borko, 1999).

Chapter 3

Aim of study

To summarise, recently a new high school CS curriculum has become effective in the Netherlands. For implementing a new curriculum it is important that teachers are able to assess it well. However, many CS teachers find it difficult to assess students' work. Therefore, a Professional Learning Community is organized about the development of assessment questions for the fundamentally new (sub)domains in the CS curriculum. In this qualitative and exploratory study, we will focus on the opinions and expectations from Dutch (former) high school CS teachers about the assessment of the learning objectives from the fundamentally new (sub)domains at a time when they have still little experience with teaching and assessing the topics corresponding to those learning objectives.

We will investigate the opinions and expectations of the participants by using the Repertory Grid Technique; we will both look at the outstanding aspects of each repertory grid, and at the similarities and differences between the different learning objectives. These outstanding aspects include for example an extreme or almost no spread of the ratings, or ratings which seem to be contradicting with other ratings. Together these aspects give a good idea of the perceptions of the participants about the assessment of the fundamentally new (sub)domains according to the definition of "Management - The Perceptual Process" (2013); the general overview and outstanding aspects give an indication of the interpretation of the assessment of the different learning objectives by the participant, the differences and similarities (groupings) indicate the organization of it, and the identification of it especially comes to view during the member checking interviews (e.g. does the participant see several ways to assess a specific learning objective).

By looking at both the perceptions of the participants about assessment, we consider thus one of the components of the Pedagogical Content Knowledge of the participants specifically, namely what the participants believe about the assessment in (Computer) Science. We aimed for doing an in-depth investigation by looking into each particular participant separately

and comparing the results. We did thus not aim to make statistically significant conclusions about Dutch CS teachers' perceptions, nor did we aim to investigate what the best way to assess each fundamentally new learning objective is.

With this focus and aim in mind, we formulated the following research questions and subquestions:

RQ: How can Dutch (former) high school CS teachers be typified based on their perceptions about the assessment of the fundamentally new learning objectives in the reformed high school CS curriculum of the Netherlands?

SQ1: What general characteristics can be identified in the outstanding opinions and expectations of the teachers regarding the assessment of the different learning objectives?

SQ2: What are the differences and similarities between the opinions and expectations of the teachers regarding the assessment of the different learning objectives?

Chapter 4

Methodology

In this chapter, we describe the methods we used to answer our subquestions and our research question, and thus how we obtained our results. This section is divided into four sections: in section 4.1 Participants is described who participated in this study, including their demographics, how they were invited and 'what is in it for them' personally. Consequently, we describe in section 4.2 'General procedure' the context in which the research happened, namely the first up to the second meeting of the PLC. This brief description of the first PLC meeting and what happened after that meeting, includes how this research is intertwined with the PLC.

Finally, in sections 4.3 'Collecting data' and 4.4 'Analyzing data', we describe which materials and how these materials were used to collect and analyze the data and how we are able to answer our sub- and research question by executing this procedure. Note that these sections are not divided in subsections for each separate material or each subquestion, because both subquestions make use of the same materials, except that the first subquestion uses some additional materials, as explained in these sections themselves.

4.1 Participants

In this research, a total of six (current or former) teachers participated. All participants were male and Dutch. The teachers varied regarding their previous education, years of teaching experience, type of CS classes they currently teach, and other relevant experience(s) as can be seen in Table 4.1. In this table all numbers are rounded up, meaning that for example someone who has ten and a half years of teaching experience is included in the category '11 - 20'.

An invitation for participating in the PLC and this research was distributed in two ways. First of all, at a teacher conference after a

workshop on assessment, teachers were informed about this PLC and asked if anyone wanted to participate. Secondly, a mail invitation was sent to high school teachers, authors of lesson materials, and teacher educators with the question to participate in this PLC. In the invitation both the PLC and this research were mentioned, although the details of the research were not known at that time and thus not described. Nine persons agreed to join as response to the invitation. At the end, only six of them participated as the other respondents had to cancel due to being too busy to participate ($N=2$), or never showed up ($N=1$).

The participants did not receive a monetary compensation, but they received compensation in the form of more knowledge and experience regarding assessment of the fundamentally new (sub)domains of the reformed Computing Science curriculum in the Netherlands. Participating in this research was entirely voluntary. Informed consent was obtained from all the participating teachers.

4.2 General procedure

The four participants were all present during two PLC meetings with eight weeks time in between. Each PLC meeting took four hours and was located in a closed room in Utrecht.

First meeting

At the start of the first meeting, the content of the PLC meetings was explained once again and the participants filled in the consent form. Then the participants were asked what their expectations were of this PLC. Next, the participants were given the (shortened) URL of an online questionnaire which they filled in on the spot. This questionnaire was developed as explained in section 4.3 'Collecting data'. The questionnaire was used as data to typify the participants based on their perceptions of the different learning objectives, as explained in section 4.4 'Analyzing data'. On average, filling in the questionnaire took 25 minutes (SD: 4 minutes). After all participants had finished it, the actual content of the PLC meeting started. A framework for making assessment questions (including how to make variations on them) was explained by means of showing a worked example. The content of this example and its relation to the corresponding subdomain was explained in detail. Meanwhile, all participants could ask questions about or give comments on this example whenever they wanted.

After a dinner break, the participants were divided into pairs. They were asked to choose a subdomain and to use the framework to make an assessment question about that subdomain themselves. After about half an hour, the created questions were discussed just like the example question

Table 4.1: An overview of the participants' demographics

Pseudonym	Age range (years)	Years of CS teaching experience	Relevant previous education	Type of CS classes currently teaching (Dutch system)	Other relevant experience
Charles	61 - 70	> 20	* Consortium retraining for CS teachers	HAVO 4, and VWO 5, 6	* Student CS supervisor * Teacher in primary education
David	31 - 40	11 - 20	* Computer Science: second grade	VWO 4, 5, 6	* Co-author teaching method
James	51 - 60	11 - 20	* Scientific education master: Computer Science * Regular teacher education Computer Science: first grade * Teacher variance 'Learning and communicating' (math)	HAVO 4, 5, and VWO 4, 5, 6	* Teacher education centre * Inspection of the Education * School director (primary education)
John	61 - 70	11 - 20	* Scientific education master: Computer Science	None	* Account manager * Author * Educational development * Educational software
Michael	51 - 60	0 - 5	* Scientific education master: Computer Science	None	* * Taught CS and math for 5 years in higher education
William	41 - 50	0 - 5	* Scientific education master: Computer Science * Regular teacher education CS: first grade	None	* Teaching material (a.o. physical computing, security, programming)

before. Next, the participants were asked to make at least one assessment question for a specific subdomain in the time between this first PLC meeting and the next. Lastly, a retrospective questionnaire was handed out to the participants which they also filled in on the spot. This questionnaire took on average two minutes to fill in.

After the first meeting

Between the first and second meeting, the participants all made an assessment question as asked during the first meeting. Each participant was interviewed to find out how and based on what he made the assessment question. This interview data was needed for the original second main research question, as explained in section B.4 'Second main research question'. Additionally, a part of this interview consisted of the member checking of the results from analyzing the filled in questionnaires. The entire interview took on average 60 minutes (SD: 21 minutes) and took place after the participant had finished the largest part of their assessment question and before the second meeting.

4.3 Collecting data

First, we created a repertory grid by identifying elements and constructs. The fundamentally new (sub)domains - as mentioned in section 2.2 - were used as elements, as identified by means of the domain analysis layer in Evidence-Centered Design. The constructs were identified by interviewing three Dutch Computing Science teachers not taking part in the study itself. The interview was a semi-structured interview consisting of ten open questions. These questions can be found in appendix G.1. An overview of the identified and used elements and constructs can be found in appendices A.1 and A.2 respectively.

An online questionnaire was made for filling in the repertory grid, using the online software tool 'Enquêtes Maken' ("Maak Gratis Online Enquêtes & Vragenlijsten met Enquêtes maken", n.d.). The questionnaire consisted of several demographic questions and 96 items in total which the participants had to answer with a 5-point Likert scale. These items were divided over 8 pages with 12 items each. Each page started with an introduction explaining that the items on that page should be answered with a particular learning objective (an element of the created repertory grid) in mind. Underneath that introduction 12 items - using a 5-point Likert scale as mentioned before - were listed, which were the same items and in the same order on each page. Along the sides of the Likert scale the bipolar constructs of the created repertory grid were shown. The labels of the Likert scale were (translated from Dutch) as follows: "Agree entirely with left side", "Agree more with

left side”, “neutral”, “Agree more with right side”, and “Agree entirely with right side”. An example page of the (final) questionnaire can be found in appendix F.

We verified that the elements and constructs in the questionnaire and thus in the repertory grid are clear and familiar to (former) CS teachers with the help of a pilot study. The participant in this pilot study was a Dutch Computing Science teacher who had not participated in the identification of the constructs. During the pilot study, the participant was asked to think out loud while filling in the questionnaire. Questions were asked when the participant was in doubt, taking time thinking or answering an item with ‘neutral’. While filling in the questionnaire, the participant gave feedback on the wording of the items. After having finished filling in the questionnaire, the participant was asked for additional comments on the questionnaire. The only comment given was that the questionnaire was very long. The questionnaire was changed improved on the feedback from this pilot study participant, although the questionnaire was not made shorter.

Finally, for the member-checking interview, the software tool Skype (n.d.) was used to call the participants and record the interview part of those calls; the interviews were not done face-to-face because of the COVID-19 virus. The member-checking recordings were transcribed and subsequently used to check the validity and get a better understanding of the interpretation (thus the analyzed results) of the repertory grid of each participants. This member-checking interview is only used for the first subquestion, and thus not for the second subquestion.

4.4 Analyzing data

The online analysis tool WebGrid Plus is an updated and online version of the computer program Rep IV (“Rep Plus — Conceptual modelling”, n.d.). This older version Rep IV was used in among others the study by Henze, Van Driel, and Verloop (2007). In summary, WebGrid Plus consists of tools to analyze, compare, and visualize repertory grids. This software was used to analyze the data of the online questionnaire. The following tools were used to analyze the data:

- **Display:**

The ‘Display’ tool presents the repertory grid as a matrix of ratings of elements (the learning objectives) on constructs (the perceptions), as explained in the documentation: “RepGrid Manual - Eliciting, Entering, Editing and Analyzing a Conceptual Grid” (n.d.).

The ‘Display’ tool is used in this study to see whether the data from the online questionnaire is accurately reproduced in the WebGrid Plus

software. In addition, during the semi-structured member checking interviews (see section G.3 for the transcriptions of these interviews) the 'Display' of the participant's raw grid was used to quickly find the indicated perception(s) of the participant on a particular learning objective.

- **Focus:**

The 'Focus' tool performs an algorithm to create a so-called 'Focus Cluster'. In this Focus Cluster, the ratings are colored based on their value: ratings with value 1 or 2 are colored white, ratings with value 3 are colored light grey, and ratings with value 4 or 5 are colored dark grey. In addition, the rows and columns of the repertory grid display are sorted in such a way that similarly rated elements and similarly rated constructs are closer together. Furthermore, the Focus Cluster shows the hierarchical structure of similarities that results from sorting the repertory grid in this described way ("RepGrid Manual - Eliciting, Entering, Editing and Analyzing a Conceptual Grid", n.d.).

The 'Focus' tool is thereby useful and thus used to give a general overview of, and find both outstanding and thus characteristic aspects (see also 3 Aim of study) in a repertory grid and similarities and differences between the participant's perceptions about the different learning objectives. Outstanding aspects in a repertory grid (which are seen as characteristic perceptions) can be:

- Ratings which seem to be contradicting with other ratings;
- Elements or constructs with an extreme spread in their ratings;
- Elements or constructs with almost no spread in their ratings;
- Constructs for which one or two of the elements have a very different rating than all other elements;
- Perceptions which are explicitly mentioned by a participant during his member checking interview.

After this analysis, we will thus for each participant have a general overview and outstanding aspects of their perceptions about the assessment of the different learning objectives. With this information we can answer subquestion 1 "What general characteristics can be identified in the outstanding opinions and expectations of Dutch (former) high school teachers regarding the assessment of the fundamentally new learning objectives in the Dutch reformed high school CS curriculum?". In addition, we will have an investigation of the similarities and differences between each participant's perceptions on the assessment of the different learning objectives. Therefore, we will be able to answer on subquestion 2 "What

are the differences and similarities between the opinions and expectations of Dutch (former) high school teachers regarding the assessment of the different fundamentally new learning objectives?”.

Together, these subquestions describe for each participant their perception of the assessment of the learning objectives in the fundamentally new (sub)domains in the reformed CS curriculum, as explained in the aim of the study (Chapter 3). Finally, we can combine all information gathered in the subquestions to make an classification of (former) CS teachers based on this data, and thus based on the participants ’perception of the assessment of the different learning objectives. As a result, we will then have an answer to our main research question ”How can Dutch (former) high school CS teachers be typified based on their perceptions about the assessment of the fundamentally new learning objectives in the reformed high school CS curriculum of the Netherlands?”.

Chapter 5

Results

A description of the results gathered can be found in this chapter. First, we describe the results (thus the data collection) in general. Consequently, for each subquestion a section follows in which the results of analyzing the corresponding collected data is described. These sections begin with a general overview of those results, followed by illustrative examples of the results mentioned in that overview.

Decisions made during data collection

During the filling in of the questionnaire, some questions arose with regards to the terminology in the constructs. These questions had thus to be answered immediately; the decisions made in the answers given, are reflected upon in section 7 Discussion. David asked whether oral tests are seen as written tests or practical work. The decision was made to include oral tests in practical work. Next, he asked whether a test on the computer is seen as a written test or practical work. The next decision was made about this topic: If computer tests are made individually and without a compiler, then they are seen as written tests. If these computer tests are, however, either not made individually or with the help of a compiler, they are seen as practical work.

Lastly, James said he quite often uses ‘individual skill assignments’ (in Dutch: ‘individuele vaardigheidsopdrachten’) for assessing particular subjects. These assignments are individual, time-bound tests (mostly several hours) which consist of several parts. For example, his test about program design and programming consists of the following three parts: reading a schematic solution (like a flow chart), designing a schematic solution based on a problem description, and translating a schematic solution to a programming language. The decision was made to see these individual skill assignments as written tests in this study.

5.1 Collected data

As six (former) teachers participated in this study (see section 4.1 Participants), the results consist of six filled in repertory grids, recorded audio data from the six interviews, and the same number of transcriptions of the member checking part of those interviews. These transcriptions can be found in appendix G.3 and the repertory grids can be found in appendix E.1. In the results section and the appendices, we use the pseudonyms of the participants as shown in section 4.1 Participants, namely Charles, David, James, John, Michael and William.

By looking at the average and standard deviation of the ratings of all participants for all constructs (see table 5.1 Statistics below), we obtain some general information of how the participants rated the learning objectives on the constructs. Since the combinations of learning objectives and constructs did get ratings between 1 and 5, the midpoint ('neutral') is 3. The majority of the averages of the constructs' ratings are relatively close (between 2.5 and 3.5) to this midpoint indicating that no pole is chosen significantly more than the other for these constructs. The standard deviations do however show that for almost all constructs, the ratings of the elements were all quite distributed.

Table 5.1: Constructs' statistics of all participants

Construct	Average of ratings	Standard deviation of ratings
A	3.07	1.29
B	2.63	1.19
C	4.00	0.97
D	3.58	0.94
E	2.78	1.27
F	2.35	0.84
G	2.60	0.59
H	3.35	1.03
I	2.80	1.49
J	2.62	1.08
K	3.77	0.83
L	2.53	0.71

One construct has a substantially lower standard deviation than the other constructs, namely the construct whether a learning objective is more suitable for students with a NT profile or for students with a CM profile (G). This observation combined with the average rating of this construct (2.60)

indicates that the participants feel that learning objectives are generally either more suitable for students with a NT profile (G:left) or as equally suitable for students with a NT profile as for students with a CM profile (G:neutral).

For only two constructs, the average of all ratings was not relatively close to the midpoint, namely the constructs about group or individual assessment tests (C) and about the duration of judging students' work (F). The ratings' average of the first construct (4.00) is relatively close to the right side of the construct (thus quite close to a rating of 5) indicating that in total more learning objectives were seen by the participants as better to assess individually than with group work (C:right). The ratings' average of the latter construct (2.35) is relatively close to the left side of the construct (thus quite close to a rating of 1) indicating that in total more learning objectives were seen by the participants as quickly to judge students' work for than time-consuming (F:left).

5.2 Analysis: Characteristic perceptions

By taking a look at the characteristic perceptions (see 4.4 Methodology - Analyzing data) within the repertory grids of the participants and their member checking interviews, we can find part of the participants' perceptions on the fundamentally new (sub)domains of the reformed CS curriculum. In this section we will only show the characteristic perceptions shared between at least two participants. An in-depth explanation of all results per participant regarding their characteristic perceptions can be found in appendix D.1 'In-depth explanation of the results per participant - Characteristic perceptions'.

5.2.1 Most common characteristic perceptions

Although the indicated characteristic perceptions differed quite a lot per participant, several participants had the following perceptions about almost all mentioned learning objectives:

1. They are best to assess individually (C:right), according to Charles, David, James, and Michael;
2. Charles, David, John, and Michael indicated that adapting (and creating) assessment questions for the learning objectives is (quite) difficult (L:right) for them;
3. They were seen as both suitable for students with both CM and NT profiles (G:neutral) by David, James and Michael, and John and Michael indicated to have at least a strong wish that they are suitable for everyone (also G:neutral);

4. Charles and David indicated that it is difficult to assess in-depth understanding (H:right) for them due to the difference in the study/learning backgrounds of the students;
5. Charles and David also indicated to know (quite) little about the misconceptions and common mistakes (J:left) for them.

David, John, and William indicated that they find adapting/creating assessment questions hard (L:right) [2], because they find it difficult to find the right level of difficulty for an assessment question and to keep that level when adapting that question. Michael also indicated that finding the correct level of difficulty is hard, although he did not indicate that most fundamentally new learning objectives are hard to assess. Charles and William gave as reason for the difficulty of creating or adapting questions (L:right) that it is difficult to think of a new context (especially combined with keeping the same level of difficulty). David also indicated that he thinks it is hard to come up with a new context, although he did not mention it in relation with this perception. Charles, David and William indicated that thinking of a new context is difficult, because one needs to be creative, and the context has to be attractive for the students for example. Lastly, Charles and James indicated that adapting/creating assessment questions is especially hard for the learning objective about grammars (VI) which is in their eyes at least partly caused by missing a theoretical background for that topic.

Although most participants did indicate that almost all learning objectives are suitable for both students with a NT profile and students with a CM profile (G:neutral) [3], they also indicated that students with a NT profile are generally better in learning objectives about exact sciences (e.g. algorithms (e.g. I) and programming (VII)) than students with a CM profile (G:left), and that learning objectives about design (e.g. X) are in general more suitable for students with a CM profile than for students with a NT profile (G:right). The relation between students with a NT profile and exact sciences is indicated by Charles, David, James, and John; the relation between students with a CM profile and design is indicated by James and William.

As mentioned before, only Charles and David indicated to know (quite) little about the misconceptions and common mistakes (J:left) of the fundamentally new learning objectives [5]. Other participants did also mention some perceptions on their knowledge of misconceptions: Not only David, but also John indicated that misconceptions grow over time by experience. Additionally, John and William both indicated that documents containing misconceptions help to know more about the misconceptions and common mistakes of a specific topic.

5.2.2 Reasons for opinions and expectations

Furthermore, the repertory grids of and member checking interviews with the participants show that they see several relations between different opinions and expectations regarding the learning objectives of the fundamentally new (sub)domains. These relations can also be seen as characteristic perceptions of the participants who mentioned them:

1. James and Michael indicated that the learning objectives are more difficult to assess (e.g. A:right) if one has no affinity with the corresponding topic,
2. The learning objectives are more difficult to assess (e.g. A:right) if one has (almost) no theoretical background for the corresponding topic, according to Charles and James.
3. Design (e.g. X) can best be assessed through group work (C:left), because that is how it happens in practice (within companies) according to Charles and William, and
4. David, Michael, and William indicated that written test (B:left) can almost always best be carried out individually (C:right) for the mentioned learning objectives.

5.3 Analysis: (Dis)similar perceptions of learning objectives

To answer the second sub question, we used the Focus clusters of the participants' repertory grids (see appendix E.2) to describe the opinions and expectations of the participants (all (former) CS teachers) on the differences and similarities between the assessment of the fundamentally new learning objectives. Overall, we can say that the participants have different ideas about how (dis)similar the different learning objectives are regarding assessment, but also that specific ideas of (dis)similar learning objectives are shared. The following ideas are shared by multiple people:

1. The learning objective about digital products (X, domain A) is rated to be quite different from the learning objectives from domain B and domain D by James, John, Michael and William.
2. The learning objectives about reasoning about correctness and efficiency of a given program (III) and explaining the structure and behavior of a given program (IX) are rated to be very similar by Charles, James, and Michael.
3. The learning objectives about correctness and efficiency (III) and grammars (VI) are rated quite similarly by David, John, and Michael.

4. A relation between groups of similarly rated learning objectives and the (sub)domains to which these learning objectives belong to, seems to exist in the cluster analyses of David, John, and William.
5. A distinction between the ratings of learning objectives about the implementation of programs and more theoretical aspects of Computer Science seems to be made in the cluster analyses of Charles, David, and John.

5.3.1 Dissimilarity of learning objective X

The difference between the learning objective of digital products (X) and the other learning objectives is most clearly illustrated in the Focus cluster of William. He rated this learning objective, in contrast to the other learning objectives, as best suitable for CM students (G:right), best to assess through group work (C:left) and practical work (B:right), and difficult to judge objectively (E:right). His perceptions that this learning objective is best suitable for CM students (G:right), best to assess through group work (C:left) and difficult to judge objectively (E:right) are shared with at least two out of the three other participants in this category (namely James, John, and Michael). Furthermore, in Williams Focus cluster, it is shown that the percentage of similarity between the ratings for all other learning objectives in his repertory grid is about 80% or higher whereas the similarity between his ratings for the learning objective about digital products and all other learning objectives is lower than 70%. Such relatively huge difference in the level of similarity between the learning objective of digital products (X) and (almost) all other learning objectives is also shown in the Focus cluster of Michael.

5.3.2 Similarity between learning objectives III and IX

The similarity between the learning objectives about reasoning about the correctness and efficiency (III) and explaining the structure and behavior (IX) of a given program is best represented by the Focus cluster of Charles. Some main differences between his ratings of these elements and the other elements are that these ratings are seen as best to assess using a written test (B:left), quite difficult to judge objectively (E:right), and best to assess individually (C:right). Three of his ratings are also shared with the other teachers in this category (namely James and Michael); they all agree that both the learning objective about correctness and efficiency (III) and the learning objective about structure and behavior (IX) are easy to assess (A:left), and best to assess individually (C:right) and using a written test (B:left).

5.3.3 Similarity between learning objectives III and VI

The cluster of the learning objective about correctness and efficiency (III) and grammar (VI) in the Focus cluster of David characterizes the fairly great similarity between these learning objectives as perceived by not only David, but also John and Michael. Despite this similarity, there is only one perception about these learning objectives all three participants agree upon, and that is that in general only one correct answer is possible (I:right).

5.3.4 Relation between clusters and (sub)domains

Johns Focus cluster is a clear representation of a clustering with a relation between the learning objectives and the (sub)domains to which they belong; the only learning objective from domain A (digital products, X) has a cluster of its own, all learning objectives from domain D can be found in one cluster together with one learning objective from B1 (translating a solution into an algorithm, I), and all other learning objectives from domain B are also clustered together. Furthermore, his cluster with learning objectives from domain B contained a subgroup including the learning objectives from subdomains B2-B4 and excluding the ones from subdomain B1. Although the Focus clusters of David and William also show this relation, the relation is less clear. The relation in Davids Focus cluster consists of the difference he sees between the learning objectives from domain B, and the learning objectives from domains A and D. In the Focus cluster of William, especially a clear difference can be found between the learning objective from domain A (digital products, X) and the learning objectives from domain B and D. Additionally, all learning objectives from domain B are grouped together in two clusters which are again grouped together. The perceptions, based on which the learning objectives are grouped together with respect to the (sub)domains by each of these participants, have nothing in common, so nothing can be concluded from that aspect.

5.3.5 Distinction between implementation and theory

The Focus cluster of Charles is the most illustrative for the clusters showing a distinction between learning objectives about implementation/programming and more theoretical learning objectives; his Focus cluster shows that he rated the learning objectives about standard algorithms (II), data structures (IV), automatas (V), imperative programming (VII), and digital products (X) to be more similar to each other than to the learning objectives about translating a solution into an algorithm (I), correctness and efficiency (III), grammars (VI), structuring a program (VIII), and explaining the structure and behavior of a program (IX). Both Charles and the other participants in this category (David and John) have all put the learning objectives about correctness and efficiency (III) and grammars (VI) in

the group with more theoretical learning objectives. Additionally, two of them also put the learning objective about data structures in this group. Only one learning objective was put in the group with learning objectives about implementation by all three participants, namely the one about imperative programming (VII). In addition, six learning objectives were clustered in that group by two participants, namely the learning objectives about translating a solution into an algorithm (I), standard algorithms (II), automatas (V), structuring a program (VIII), explaining the structure and behavior of a program (IX), and digital products (X). Note that this last learning objective was put in a totally separate cluster, thus not similar to learning objectives about implementation and also not similar to more theoretical learning objectives.

Chapter 6

Conclusions

SQ1: *What general characteristics can be identified in the outstanding opinions and expectations of Dutch (former) high school teachers regarding the assessment of the fundamentally new learning objectives in the reformed Dutch high school CS curriculum?*

The perceptions of the (former) teachers differ quite a lot, but still several characteristic perceptions emerged among several teachers. Many (former) teachers feel that almost all fundamentally new learning objectives are best to assess individually and that adapting and creating assessment questions for them is quite difficult. Additionally, many (former) teachers indicated to have difficulties with creating new contexts in such a way that the level of difficulty stays the same for that question.

Furthermore, a lot of (former) teachers feel that almost all mentioned learning objectives are suitable for both students with a CM and students with a NT profile, or that they wish all learning objectives are equally suitable for everyone. The majority of the (former) teachers did however also indicate that students with a NT profile are, in general, better in learning objectives about exact sciences, and that learning objectives about design are generally more suitable for students with a CM profile.

In addition, some (former) teachers indicated to know quite little about the misconceptions and common mistakes of the fundamentally new learning objectives as they have to grow over time by experience, that assessing in-depth understanding is difficult caused by the difference in study backgrounds of the students, and that written tests are almost always best to carry out individually.

Finally, several perceptions exist about the reason for particular perceptions about learning objectives. Several (former) teachers feel that learning objectives are more difficult to assess if one has no affinity with the corresponding topic or has almost no theoretical background for that topic.

Lastly, some feel that how tasks corresponding to a topic are performed in practice, is important for how those topics can best be assessed.

SQ2: *What are the differences and similarities between the opinions and expectations of Dutch (former) high school teachers regarding the assessment of the fundamentally new learning objectives in the Dutch reformed high school CS curriculum?*

Many (former) teachers feel that the learning objective 'The student can design, implement and evaluate a digital product for a given situation' differs quite a lot from the other fundamentally new learning objectives and is in contrast to those other learning objectives, for example, best suitable for students with a CM profile, best to assess through group work and practical work, and difficult to judge objectively.

Several (former) teachers perceive the assessment of the learning objectives 'The student can reason about the correctness and efficiency of a given algorithm' and 'The student can explain the structure and behavior of a given program' to be similar; these learning objectives are, in their eyes, easy to assess and best to assess individually and using a written test. Additionally, several teachers feel that learning objectives from the same (sub)domain are more similar to each other regarding their assessment than to learning objectives from other (sub)domains.

Lastly, several (former) teachers feel that both learning objectives about the implementation of programs and learning objectives about the more theoretical aspects of Computer Science are more similar to the other learning objectives in their group regarding their assessment than to the learning objectives in the other group.

RQ: *How can Dutch (former) high school CS teachers be typified based on their perceptions about the assessment of the fundamentally new learning objectives in the reformed high school CS curriculum of the Netherlands?*

In this study, we found that many opinions and expectations exist which can be used to typify the CS teachers; in total, we found 22 perceptions which were indicated by at least two (former) teachers in this study. We did however also find that only two perceptions which were shared by the exact same group of (former) teachers, namely:

- finding it difficult to assess in-depth understanding for almost all fundamentally new learning objectives *and* knowing little about common mistakes for almost all learning objectives, and
- feeling that adapting questions is especially hard for grammars *and* that learning objectives are more difficult to assess if one has no theoretical background in them.

Additionally, we found that the majority of the perceptions indicated by four out of the six (former) teachers, were also indicated by a group of three (former) teachers who shared at least one other perception together, which suggests that there might be a relation between these perceptions:

- All (former) teachers who indicated to feel that the learning objectives about implementation are more similar to each other than to more theoretical learning objectives and the other way around, also indicated to feel that almost all learning objectives are difficult to adapt assessment questions for and that students with a NT profile are generally better in exact sciences than students with a CM profile;
- Everyone who indicated to feel that almost all learning objectives are both suitable for students with a CM and NT profile, also indicated to feel that almost all learning objectives are best to assess individually and that the learning objectives about correctness and efficiency and grammars are quite similar regarding their assessment;
- The (former) teachers who indicated to feel that the learning objectives about reasoning about the correctness and efficiency and explaining the structure and behavior of a given program, also indicated to feel that almost all learning objectives are best to assess individually;
- All (former) teachers who indicated to see a relation between groups of learning objectives and (sub)domains, feel that written tests can best be assessed individually or that the learning objective about digital products is quite different from other learning objectives, also indicated to find it hard to find and keep the same level of difficulty.

In the next section (7 Discussion), we will discuss possible explanations and interpretations for all above explicitly mentioned perceptions which can be used to typify the (former) teachers.

Chapter 7

Discussion

7.1 Reflection on conclusions

As mentioned in section 6 Conclusions, two (former) teachers both indicated that they find assessing in-depth understanding difficult and that they know little about misconceptions and common mistakes for almost all learning objectives. No (former) teacher was asked explicitly whether he feels that a relation exists between assessing in-depth understanding and knowledge about misconceptions and common mistakes. However, during the member checking interviews, for both perceptions possible causes were given; The main reason given for having little knowledge about misconceptions and common mistakes for a topic was the little experience with the fundamentally new learning objectives. Difficulty with assessing in-depth understanding is caused by not knowing whether a question is indeed an in-depth question or a application question, because that also differs per student, according to the (former) teachers in this study. As these causes seem to have nothing to do with each other, it is not very likely that a relation exists between these perceptions.

Of the two (former) teachers indicating that adapting questions is especially hard for grammars, one indicated that he feels most uncomfortable with assessing grammars because he has almost no theoretical background for that subject and also because he is not convinced that students would need that knowledge. The other (former) teacher indicated that this difficulty was at least partly caused by the abstractness of grammars and because he has done nothing with grammars for almost thirty years; he does however have a theoretical background in it. This observation leads to not only the belief that a type of (former) CS teachers may exist who have no theoretical background in grammars and therefore have some difficulties with assessing the corresponding learning objective, but also that having difficulties with assessing grammars is not necessarily caused by the lack of a theoretical background. Whether the lack of a theoretical background in

grammar always leads to difficulty with the assessment of that topic, can not be concluded from this study.

One of the common perceptions found, is that all (former) teachers who indicated to see a relation between groups of learning objectives similar in their assessment and their corresponding (sub)domains, also indicated to feel that the learning objective about digital products is quite different from the other learning objectives. When looking at the (sub)domains of the fundamentally new learning objectives, we see that the learning objective about digital products is the only learning objective from domain A whereas all other learning objectives are from domain B or D. This observation suggests that this found relationship is very likely to exist. Nothing from this study indicates or refutes a relation between the other common perceptions which are shared by approximately the same groups of (former) teachers, other than the similarity of the groups of (former) teachers who indicated these perceptions.

7.2 Reflection on findings in general

To find a possible explanation for the differences found in the indicated opinions and expectations of the teachers, we explored the relation with the teachers' demographics. We found that all former teachers, who are not currently teaching a high school CS class, indicated to feel that the learning objective about designing, implementing and evaluating a digital product is different from the other learning objectives regarding its assessment whereas the teachers who are currently teaching high school CS classes did not. This observation indicates that the assessment of the learning objective about digital products might seem different from the other learning objectives, but is not that different in practice.

Additionally, we found that only the two (former) teachers without a master in Computer Science indicated to know little about the misconceptions and common mistakes for almost all fundamentally new learning objectives, both in this study and in their retrospectives at the end of the first PLC meeting (see section 4.2 'General procedure'). Our findings suggest that this lack of knowledge is at least partly caused by the lack of experience in teaching and assessing the learning objectives (as indicated by two teachers), but also might be caused by their lack of theoretical background; two of the teachers with a CS master indicated to feel that documents with misconceptions corresponding to a particular topic are useful for one's knowledge of misconceptions and common mistakes for that topic, and none of the teachers without a CS master did.

Two common perceptions were noted which were also found by Yadav et al. (2016). Yadav et al. mentioned in their study that "one teacher stated

that his lack of computer science background made it difficult for him to come up with assessment items” which is in accordance with the perception indicated by two (former) teachers in this study that ‘learning objectives are more difficult to assess if one has almost no theoretical background for the corresponding topic’ (see appendix C ‘Shared found perceptions’). Additionally, Yadav et al. (2016) reported that one teacher indicated that with group work it is difficult to know whether each individual student understands the corresponding topic, which was also mentioned by two (former) teachers in the current study.

Two of the common perceptions as indicated by the participants were that ‘Almost all learning objectives are difficult to create or adapt assessment questions for’ and ‘Thinking of a new context is difficult’ (see appendix C ‘Shared found perceptions’). These perceptions were also already indicated indirectly in the expectations of the PLC as mentioned by the participants (see section 4.2 ‘General procedure’); For example, Charles said that he expected to learn how to make better assessment questions and James indicated that he expected to learn how to think of suitable contexts (which also appeal to the imagination) for assessment questions. Additionally, James said he expected to learn more about guidelines regarding the level expected of students regarding their CS knowledge and skills. This expectation seems to relate to the found common perception ‘Keeping the same level of difficulty and finding the right level of difficulty is difficult’. These observations increase the credibility of these three found common perceptions.

Three perceptions regarding misconceptions and common mistakes were found to be shared by at least two of the participating (former) teachers, namely ‘For almost all learning objectives they know little about misconceptions and common mistakes’, ‘Knowledge of misconceptions and common mistakes grow over time by experience’, and ‘Documents with misconceptions corresponding to a particular topic are useful for one’s knowledge of misconceptions and common mistakes for that topic’. These findings suggest that the knowledge of misconceptions and common mistakes of the (former) teachers will be increased (J:more right) at the end of the PLC as the teachers then have more experience. Therefore, we may expect that the ratings’ average of the construct about misconceptions and common mistakes (J) will be higher than the current average 2.62 if the participants of this study fill in the repertory grid again at the end of the PLC meetings.

7.3 Reflection on decisions during data collection

During the data gathering, the decision was made to see oral tests as practical work, in contrary to as a written test (see section 5 ‘Decisions made during data collection’). In practical work, a student has to perform

a task or activity whereas in written tests, the student responds to specific questions by writing or typing. In oral tests, the teacher ask questions to the student verbally, who then responds to those questions verbally. Therefore, one can argue that oral tests fall under written tests as only a different way of responding is used. Based on that reasoning, the made decision might be revised.

As the practical assignment skills do consist of questions to which the students respond by writing, one can argue that the practical assignment skills indeed belong to the category 'written test'. However, as one of the parts of these practical assignment skills is the translation of a schematic solution to a programming language (see section 5 'Decisions made during data collection'), one can also argue that the student has to perform a task and that these practical assignment skills belong to the 'practical work' category. Therefore, in future studies, we suggest to look at these practical assignment skills as if they are three different tests as each part addresses a distinct topic/learning objective. By doing so, one can simply look at the content of each of these tests to see whether they should be seen as a written test or as practical work.

The same reasoning about the difference between written tests and practical work can be applied to the decision about the computer tests. If computer tests are made without a compiler, the test is likely to ask mainly theoretical questions in contrary to practical (programming) tasks. If however computer tests are made with a compiler, we might assume that the computer tests includes practical (programming) tasks. Therefore, this decision does not need to be revised. However, whether the computer test has to be made individually or in groups does not - without assumptions about the relation between written tests and individual work as well as practical work and group work - give a clear indication whether the computer test contains especially (theoretical) questions or practical tasks. One might thus argue that the decision made should not involve this aspect of the computer test.

7.4 Limitations and future directions

In the present study only a small number of (former) teachers participated. While this small number of (former) teachers precludes making statistically significant conclusions of (former) Dutch high school CS teachers' perceptions about the assessment of the fundamentally new learning objectives in general, this property allowed the current study to investigate the opinions and expectations of the participants in-depth. In future studies, the findings of this study can be used as hypotheses for a larger number of (former) Dutch high school CS teachers to increase the external validity of the findings.

Furthermore, the participating (former) teachers were all eager to learn more about and to improve their knowledge of and skills regarding the assessment of the fundamentally new learning objectives, as indicated by their entirely voluntary participation in this PLC. Therefore, the group of participants are not expected to be representative for all (former) high school CS teachers. In future research, we suggest to not only look at this kind of teachers, but also at teachers who are less or not willing to develop professionally. Except for this characteristic of the participating (former) teachers, the participants are quite diverse (see section 4.1 'Participants') and are thus likely to represent a large part of the Dutch (former) high school CS teachers.

In the questionnaire used by the participants to fill in their repertory grid, the learning objectives (thus the elements) and constructs were always asked in the same order (see the appendix A 'Repertory grid abbreviations' for the used order). Because in the order of the elements those in the same (sub)domains are placed next to each other, this order might have had an effect on the number of participants indicating to feel that 'Learning objectives from the same (sub)domain are often more similar to each other than to learning objectives from other (sub)domains regarding their assessment'.

In this study, we looked into the outstanding opinions and expectations of the (former) teachers regarding the assessment of the different learning objectives as well as the differences and similarities between their opinions and expectations of the assessment of these learning objectives in order to find the perceptions of the (former) teachers regarding the assessment of the learning objectives. In future studies, one can also compare the different repertory grids of the participants to investigate the teachers' perceptions even more in-depth (for example with the tool Compare of RepPlus, see "RepGrid Manual - Eliciting, Entering, Editing and Analyzing a Conceptual Grid" (n.d.)).

Finally, when taking a look at the results and conclusions of this study, it is important to realise that the participants who did not indicate to have a particular opinion or expectation, may or may not have that opinion or expectation. This observation implies that more common perceptions might exist and that more or different perceptions might be shared by the exact same group of participating (former) teachers. In future work, we suggest to overcome this limitation by asking the participants whether they do or do not share each found opinion or expectation. With this check, the credibility of the study also increases.

7.5 The original problem

As explained in Chapter 1 Introduction, CS teachers often have difficulties with assessing CS topics but no study could be found which investigates in-depth what the CS teachers have difficulties with. In our study, we found that many teachers have difficulties with creating and/or adapting assessment questions (for a resit for example). Additionally, many (former) teachers indicated to find the right level of difficulty and keep that same level of difficulty when adapting an assessment question as well as to think of a new, suitable context for an assessment question. The findings also suggest that the difficulties with creating and/or adapting assessment questions are related to the last mentioned difficulties. Therefore, we expect that teachers can improve their assessment skills if (more) attention is paid to these skills/knowledge in teacher education, in training opportunities and in relevant Professional Learning Communities.

Finally, some teachers indicated to feel that adapting questions is especially hard for the learning objective about grammars (for different reasons). No other learning objectives were found for which several of the (former) teachers indicated to have more difficulties with than for the other learning objectives. This observation suggests that either Dutch (former) CS teachers do not have difficulties with the assessment of specific learning objectives (except for the one about grammars) from the fundamentally new (sub)domains in the reformed curriculum, or that the specific learning objectives that a teacher has difficulties with assessing it, differ a lot per teacher. In the latter case, the findings in this study suggest that which learning objectives are found to be more difficult by the (former) teacher depends on the theoretical/educational background of the teacher and their affinity with the corresponding topic.

Bibliography

- [1] Cultuur en Wetenschap Ministerie van Onderwijs. *Regeling van de Staatssecretaris van Onderwijs, Cultuur en Wetenschap van 27 september 2016, nr. VO/1048771, houdende wijziging van de Regeling examenprogramma's voortgezet onderwijs in verband met het vernieuwen van het examenprogramma informatica havo en vwo.* <https://zoek.officielebekendmakingen.nl/stcrt-2016-52085.html>. Retrieved December 17, 2020. 2016.
- [2] A. Yadav et al. “Expanding computer science education in schools: understanding teacher experiences and challenges”. In: *Computer Science Education* 26 (2016), pp. 235–254. DOI: 10 . 1080 / 08993408 . 2016 . 1257418.
- [3] A. M. Curtis et al. “An Overview and Tutorial of the Repertory Grid Technique in Information Systems Research”. In: *Communications of the Association for Information Systems* 23 (2008), pp. 37–62. DOI: 10 . 17705 / 1cais . 02303.
- [4] I. Henze, J. Van Driel, and N. Verloop. “The Change of Science Teachers’ Personal Knowledge about Teaching Models and Modelling in the Context of Science Education”. In: *International Journal of Science Education* 29 (2007), pp. 1819–1846. DOI: 10 . 1080 / 09500690601052628.
- [5] C. J. Turner. “Impact of Professional Learning Community Design on Teacher Instruction”. Retrieved from PQDT Open. (Publication No. 3684227). PhD thesis. University of Arkansas, Fayetteville, 2015.
- [6] L. Stoll and K. Louis. *Professional learning communities: Divergence, depth and dilemma*. Open University Press, 2007.
- [7] S. M. Hord and Educational Resources Information Center (U.S.) *Professional Learning Communities : Communities of Continuous Inquiry and Improvement*. Tex: Southwest Educational Development Laboratory, 1997.

- [8] College voor Toetsen en Examens. *Examenprogramma informatica havo/vwo*. https://www.examenblad.nl/examenstof/informatica-havo-en-vwo-3/2020/f=/examenprogramma_informatica_havo-vwo.pdf. Retrieved December 17, 2020.
- [9] College voor Toetsen en Examens. *Examenprogramma informatica havo/vwo*. https://www.examenblad.nl/examenstof/informatica-vwo-2/2018/f=/inf_havovwo.pdf. Retrieved December 17, 2020.
- [10] SLO Nationaal expertisecentrum voor leerplanontwikkeling. *Karakterisering van de vernieuwing*. <https://slo.nl/handreikingen/havo-vwo/handreiking-se-info-hv/>. Retrieved December 17, 2020.
- [11] S. Grover and R. Pea. “Computational thinking in k-12: A review of the state of the field”. In: *Educational Researcher* 42 (2013), pp. 38–43. DOI: 10.3102/0013189X12463051.
- [12] M. McDonald and G. McDonald. “Computer Science Curriculum Assessment”. In: *Acm Sigcse Bulletin* 31 (1999), pp. 194–197. DOI: 10.1145/384266.299751.
- [13] J. L. J. Tolboom, J. Krüger, and N. Grgurina. *Informatica in de bovenbouw havo/vwo; Naar aantrekkelijk en actueel onderwijs in informatica*. SLO Nationaal expertisecentrum voor leerplanontwikkeling, 2014.
- [14] G. Kelly. *The Psychology of Personal Constructs: A theory of personality*. W.W. Norton and Company Inc., 1955.
- [15] H. M. Edwards, S. McDonald, and S. M. Young. “The repertory grid technique: Its place in empirical software engineering research”. In: *Information and Software Technology* 51 (2009), pp. 785–798. DOI: 10.1016/j.infsof.2008.08.008.
- [16] R. L. Herman. “The repertory grid technique: path to teacher description or teacher potential?” In: *Instructional Science* 24 (1996), pp. 439–459. DOI: 10.1007/BF00125579.
- [17] F. Fransella, D. Bannister, and R. Bell. “Constructs And Elements”. In: *A manual for repertory grid technique*. Retrieved June 10, 2020 from ProQuest Ebook Central, <https://ebookcentral-proquest-com.ru.idm.oclc.org/lib/ubnru-ebooks/detail.action?docID=219766>. Wiley, 2004, pp. 15–53.
- [18] L. Shulman. “Those Who Understand: Knowledge Growth in Teaching”. In: *Educational Researcher* 15 (1986), pp. 4–14. DOI: 10.2307/1175860.
- [19] P. L. Grossman. *The making of a teacher: Teacher knowledge and teacher education*. Teachers College Press, Teachers College, Columbia University, 1990.

- [20] S. Magnusson, J. Krajcik, and H. Borko. “Nature, Sources, and Development of Pedagogical Content Knowledge for Science Teaching”. In: 2002, pp. 95–132. DOI: 10.1007/0-306-47217-1_4.
- [21] P. Tamir. “Subject matter and related pedagogical knowledge in teacher education”. In: *Teaching and Teacher Education* 4 (1988), pp. 99–110. DOI: 10.1016/0742-051X(88)90011-X.
- [22] H. Jing-Jing. “A critical review of Pedagogical Content Knowledge’ components: nature, principle and trend”. In: *International Journal of Education and Research* 2 (2014), pp. 411–424.
- [23] *Management - The Perceptual Process*. <http://oer2go.org/mods/en-boundless/www.boundless.com/management/textbooks/boundless-management-textbook/index.html>. 2013.
- [24] *Maak Gratis Online Enquêtes & Vragenlijsten met Enquêtes maken*. <https://www.enquetesmaken.com/>. Retrieved January 16, 2020.
- [25] Skype. *Skype — Communication tool for free calls and chat*. <https://www.skype.com/en/>. Retrieved May 31, 2020.
- [26] *Rep Plus — Conceptual modelling*. <http://pages.cpsc.ucalgary.ca/~gaines/repplus/markdown-2/>. Retrieved January 16, 2020.
- [27] *RepGrid Manual - Eliciting, Entering, Editing and Analyzing a Conceptual Grid*. <http://pages.cpsc.ucalgary.ca/~gaines/Manuals/RepGrid.pdf>. Retrieved January 16, 2020.
- [28] ATLAS.ti Scientific Software Development GmbH. *What is ATLAS.ti*. <https://atlasti.com/product/what-is-atlas-ti/>. Retrieved January 28, 2020.

Appendix A

Repertory grid abbreviations

In this appendix, two tables are shown; the first table (see next page) shows the learning objectives used in this study with their corresponding symbol and subdomain. The second table shows the constructs used together with their corresponding symbol. For both the constructs and the elements, abbreviations are used within these study; these abbreviations are made bold in the descriptions of the elements and constructs in their respective tables.

A.1 Elements

Table A.1: Repertory grid elements (learning objectives)

Element	Learning objective	Corresponding subdomain
I	The student can translate a solution to a problem into an algorithm (i.e. in terms of steps, choices, and repetitions).	B1
II	The student can recognize and use standard algorithms (e.g. for determining the maximum, calculating the average, sorting).	B1
III	The student can reason about the correctness and efficiency of a given algorithm.	B1
IV	The student can (for a given algorithm) compare and contrast different options for data structures in terms of elegance and efficiency.	B2
V	The student can describe the behavior of a (physical) system as a finite automata (i.e. in terms of states and state transitions).	B3
VI	Given a grammar and input, the student can determine whether the input is correct .	B4
VII	The student can, for a given algorithm, develop a program in an imperative programming language .	D1
VIII	The student can structure a program so that others can easily understand and evaluate it.	D1
IX	The student can explain the structure and behavior of a given program.	D2
X	The student can design, implement and evaluate a digital product for a given situation.	A8

A.2 Constructs

Table A.2: Repertory grid constructs (perceptions of learning objectives)

Construct	Left pole of construct	Right pole of construct
A	I think it's easy to assess this learning objective.	I think it's difficult to assess this learning objective.
B	This learning objective can best be assessed using a written test .	This learning goal can best be assessed through practical work .
C	This learning objective can best be assessed with group work .	This learning goal can best be assessed individually .
D	Creating an assessment for this learning objective isn't much work .	Creating an assessment for this learning objective takes me a lot of time .
E	I can easily judge students' work unambiguously .	I think it's difficult to judge students' work objectively .
F	I can judge students' work quickly .	Judging students' work is time-consuming .
G	This learning objective particularly suits students with a NT (beta-science) profile .	This learning objective particularly suits students with a CM (alpha-cultural) profile .
H	For this learning objective, I can easily assess indepth understanding .	I think it's difficult to assess indepth understanding for this learning objective.
I	Generally, several correct answers are possible.	Generally, only one correct answer is possible.
J	I know little about the misconceptions and common common mistakes for this learning objective.	I know which misconceptions and common mistakes are involved with this learning objective.
K	Students' scores are generally fairly close to each other .	Students' scores are generally far apart .
L	I think it's difficult to adapt questions for new tests, with the same learning objectives and level of difficulty the same	I think it's easy to adapt questions for new tests, for the same learning objectives and level of difficulty

Appendix B

Development of research

B.1 Initial research plan

Originally, the overarching research (see section 1.1 'Context and overarching research') would investigate the effect of the Professional Learning Communities (PLC) meetings on the Pedagogical Content Knowledge (PCK) regarding assessment of the participants, more specifically by looking at the PCK on the domain Fundamentals from the reformed Computer Science (CS) curriculum in the Netherlands. In this research we would look at the baseline measurement, so at what the PCK assessment of the participants looks like at the start of the PLC meetings. Ideally, to determine this, we would have taken a look at all aspects of PCK according to Magnusson, Krajcik, and Borko (1999): what a teacher knows, believes, does, and the reasons for his actions.

In terms of data sources, however, we would only have used repertory grids (Curtis et al., 2008) which only indicate what the participants believe about the different learning goals and about the relationships between the different learning objectives. With this technique we would thus actually have investigated the perceptions of the participating teachers about the learning objectives. We would have used the repertory grids to look per teacher:

1. which learning objectives he finds to be (dis)similar regarding their assessment, and why
2. what is striking when looking at the specific assessments of the learning objectives (for example, many neutral perceptions for a specific learning objective)

In addition, we would have used the audio recordings of the PLC meetings, the by the participants created assessment questions and the retrospectives (in which the participants looked back at each separate

PLC meeting) as data sources to get an idea of how the PCK regarding assessment is likely to change. This research would however have been a side-research and thus not part of the main study, but rather a separate section with interesting results and their interpretation (not part of the research questions) in the discussion.

B.2 Interim research plan

For the interim research plan of this study - the research plan of the larger study had not changed - we would also take a look at how the Rep Plus software (“Rep Plus — Conceptual modelling”, n.d.) can be used to analyze the effect of the PCK regarding assessment of the teachers. The idea was that we would both look at the baseline measurement (as described in the first situation) and come up with advice for the researcher of the larger study (Renske Weeda) on how she can use the Rep Plus software herself. The main question should therefore be something of how the effect of the PLC meetings on PCK regarding assessment can be determined using the baseline measurement and the Rep Plus software.

B.3 Final research plan

In the final setup, there are more important data sources than just the repertory grids, namely also the prepared assessment questions of the teachers and how they have created them (based on an interview). The repertory grids remained the most important part though. At the end, due to time constraints, only the first part of this research plan was entirely executed. Therefore, this paper describes only the entirely executed part of this study, and the part that was not executed entirely is briefly described in section B.4 below.

In the final research, we look specifically at the individual lecturers in terms of repertory grids, and based on that we analyze to what extent a typification of the lecturers is possible (such as done in Henze, Van Driel, and Verloop, 2007). To make this typification, we look at:

- which learning objectives he feels to be (dis)similar regarding their assessment, and why
- what general characteristics can be found when looking at the outstanding perceptions of the (former) teachers regarding the assessments of the learning objectives (for example, many neutral ratings or no neutral ratings at all)

Furthermore, we look at the assessment questions that the teachers have prepared themselves and hold an interview with them to find out based

on what and how they have created their assessment question(s). This interview would then have been used to get an indication (for the larger research) where a possible shift or development lies in the perceptions of the teachers about the different learning objectives and what contributed to this change. These interviews therefore consist of, for example, questions such as "What insight(s) from the previous PLC meeting did you use as input or inspiration to create this assessment question?" and "What is special about (how you made) this assessment question in your opinion?".

Additionally, several questions are asked to find out the possible relationship between the assessment question made by a participant and how that participant looks at the learning objective corresponding to that assessment question. These questions are about the same perceptions as the ones used in the repertory grid from the first part of the study. Think of, for example, "Did you find it difficult to create an assessment question for this learning objective?".

B.4 Second main research question

As mentioned above, for the second research question, we wanted to take a look at what parts or aspects from the Professional Learning Community (PLC) possibly have an effect on these perceptions. For this research question, we would thus only have taken a look at the aspect 'what the reasons for teachers' actions are' of PCK. Although we have executed this second part of the study up to and including collecting the needed data by holding the interviews with the participants, not enough time was left to also perform the entire analysis on this data and writing the corresponding results and findings down. Therefore, we will summarize in this section what we have and would have done for this second research question.

B.4.1 Aim of study

To improve the quality of the upcoming and future PLC meetings, it is useful to know what the participants experience as helpful of the activities during the PLC meetings. To retrieve this information, we will them directly, but we will also investigate their experiences by, for example, asking about differences between how the participants performed activities before and after the PLC meetings. This information combined with the perceptions of the participants at the start of the PLC meetings, can be even more useful to improve the effectiveness of the PLC, because this information can for example be used to combine activities useful in the eyes of the participants with topics they have difficulties with. By combining this information, we will also have an indication of what factors/activities of the PLC meetings can possibly cause a shift in the perceptions of the participants of such PLC,

and thus also increase the knowledge base of what makes a PLC regarding the assessment of CS learning objectives effective for the participants.

B.4.2 Methodology

The participants for this second part of the research are the same people who also participated in the first part of this research (see section 4 Methodology). The tool Skype (n.d.) was used to collect the data in the same way as for the member checking interviews of the first part of the research mentioned in the last paragraph of section 4.3 'Collecting data'. This part of the interview would not have been transcribed, but immediately put in the analysis tool, namely the software tool AtlasTI (ATLAS.ti Scientific Software Development GmbH, 2020). This tool can, among others, be used for the coding of audio and video data. The user/researcher can choose what code groups and codes exist and link those codes to pieces of the data. Additionally, comments can be added to these codes, such as the transcription of the corresponding piece of data in order to have more context than only the code itself. After (part of) the data has been coded, AtlasTI gives the user/researcher more information about the coded data, such as how often which code occurs and in which documents. Therefore, AtlasTI is useful to get a nice overview of for the researcher interesting information of recorded interviews, without the need to transcribe the interview completely.

The procedure mentioned in section 4.2 'Methodology - General procedure' includes the procedure for how and when the data for this second research question is gathered, namely a semi-structured interview. The questions asked as data for the second research question (thus all questions except the member checking questions in these interviews) can be found in appendix G.2 'Leading interview questions'. As explained before, the software AtlasTI would have been used to analyze the collected data. The codes which would have been used to analyze the recorded interview, are and would have been based on the interview questions and the received answers. The codes which at least would have been used, can (in short) be found in the overview below:

- Beginning of developing assessment question
 - Looking more in-depth at the corresponding learning objective
 - Getting an idea for the context of the assessment question
 - Getting an idea for the assessment question in general
- Input/inspiration used for developing assessment question
 - Paper or other source of information
 - Previously made assessment question

- Useful of entire first PLC meeting
 - Use of template
 - Working together with other teachers/experts
- Difference in developing assessment question before and after first PLC meeting
 - Focus on learning objective
 - Thinking about variations
 - Thinking about misconceptions and common mistakes
 - Template in general
- Advantages use of template
 - Assessment model
 - Misconceptions and common mistakes
 - Variations
 - Focus on learning objective
 - Template as checklist
- Disadvantages use of template
 - Duration to complete/fill in
 - Limited freedom
 - None

Appendix C

Shared found perceptions

The following found perceptions about the assessment of the learning objectives from the fundamentally new (sub)domains in the recently reformed Computer Science curriculum in the Netherlands were (indicated to be) shared by at least two participants:

1. Almost all learning objectives can best be assessed individually;
2. Almost all learning objectives are difficult to create or adapt assessment questions for;
3. Almost all learning objectives are both suitable for students with a CM profile and students with a NT profile;
4. Almost all learning objectives are difficult to assess in-depth understanding for;
5. For almost all learning objectives they know little about misconceptions and common mistakes;
6. They have a strong wish that all learning objectives are suitable for all students;
7. Adapting questions is especially hard for grammars;
8. Students with a NT profile are generally better in learning objectives and questions dealing with exact sciences than students with a CM profile;
9. Design is, in general, more suitable for students with a CM profile than for students with a NT profile;
10. Keeping the same level of difficulty and finding the right level of difficulty is difficult;
11. Thinking of a new context is difficult;

12. Knowledge of misconceptions and common mistakes grow over time by experience;
13. Documents with misconceptions corresponding to a particular topic are useful for one's knowledge of misconceptions and common mistakes for that topic;
14. Learning objectives are more difficult to assess if one has no affinity with the corresponding topic;
15. Learning objectives are more difficult to assess if one has (almost) no theoretical background for the corresponding topic;
16. Design can best be assessed in group work, because that is how it happens in companies;
17. Written tests can best be taken individually;
18. The learning objective 'The student can design, implement and evaluate a digital product for a given situation' is different from the other learning objectives regarding its assessment;
19. The learning objectives 'The student can reason about the correctness and efficiency of a given algorithm' and 'The student can explain the structure and behavior of a given program' are very similar regarding their assessment;
20. The learning objectives 'The student can reason about the correctness and efficiency of a given algorithm' and 'Given a grammar and input, the student can determine whether the input is correct' are quite similar regarding their assessment;
21. Learning objectives from the same (sub)domain are often more similar to each other than to learning objectives from other (sub)domains regarding their assessment;
22. Learning objectives about implementation are more similar to each other than to more theoretical learning objectives regarding their assessment, and the other way around.

Appendix D

In-depth explanation of the results per participant

D.1 Characteristic perceptions

D.1.1 Charles

Charles sees each learning objective as difficult to assess in-depth understanding for (H:right). He indicated that he thinks asking 'the right question' on such kind of (meta)cognition level is always difficult, and that it differs per student whether a question assesses application or in-depth understanding; what students have learned differs a lot per school/city, and even in the same class it can differ slightly. Because he took the students from the different schools of the in the PLC participating (former) teachers into account, what each students has learned can differ quite a lot and therefore he felt that it is quite difficult to assess in-depth understanding for the learning objectives.

In the repertory grid, Charles also indicated to know little about misconceptions and common mistakes (J:left). Given that he took the differences between the students and what they have learned into account, we may assume that is also (at least part of) the reason that he feels to know little about the misconceptions and common mistakes of the students for almost each learning objective. This assumption seems even more reasonable when taking a look at the clustering of the constructs in Charles' focus cluster (see section E.2 for the focus clusters all participants); this clustering shows that Charles rated the constructs about knowing little about misconceptions and common mistakes (J) and thinking it is difficult to assess in-depth understanding (H) very similarly.

In addition, he feels that students' knowledge can best be assessed individually (C:right). He does however think that executing projects and topics dealing with users (e.g. creating a system or interface) can best be

done and assessed through group work (C:left), because in companies (these) projects are also almost never done individually as systems are so complex. He feels that describing a (physical) system as a (finite) automata (V) falls among these topics, and is therefore better to assess through group work. He also mentions that if one sees the automata learning objective (V) or programming (VII) as knowledge, it is also possible to assess it individually. Note that this opinion of Charles does not correspond with what he indicated in the questionnaire; in the questionnaire he indicated to see the automatas learning objective (V) as the only learning objective which is better to assess through group work (C:left) than individually (C:right) whereas in the interview he indicates to see several learning objectives as better to assess through group work (C:left), such as the one about imperative programming (VII), and digital products (X).

Furthermore, Charles indicated in the repertory grid that almost all learning objectives are as much suitable for students with a NT-profile as students with a CM-profile (G:neutral). He explained that he also sees the previously mentioned difference per student here in this case. However, he also indicated that students with a NT-profile are generally better in "exact sciences stuff" (translated to English) like algorithms, and that especially students with a CM-profile have often more difficulties in following programming steps: "even within the CM-profile students exist who are also very well able to follow the programming steps in Computer Science" (translated to English).

Charles also indicated that he thinks adapting questions for a new test where the learning objective and level of difficulty stay the same, is quite hard (L:left). He elucidated that in math you often can simply change a few values to create a new, similar question. In Computer Science, however, you need an entirely new context for which you can ask the same thing, if you want to create a similar, but new question. He told us that he thinks adapting questions for the domains B and D is especially hard compared to for example understanding how a computer works or compression. Regarding the specific learning objectives of the fundamentally new domains, he indicated that he finds grammars (VI) especially hard to adapt questions for, and he thinks that is perhaps caused by him not having an exact sciences background.

Additionally, Charles feels that for almost all learning objectives several correct answers exist (I:left), except for the learning objective about structuring a program (VIII) in some particular cases. For this last learning objective, he feels that the number of possible correct answers varies with the used programming language; for very strict programming languages it is quite possible that only one correct answer exists (I:right), but for other languages for example several orders for functions are possible and then you can have several correct answers (I:left).

Other outstanding aspects of Charles' repertory grid (which we could not discuss in the interview due to time constraints) are that Charles feels that:

1. creating an assessment takes a lot of time (D:right) for almost all learning objectives in the fundamentally new domains,
2. judging students' work is often time-consuming (F:right), and
3. in contrast to the other learning objectives, the learning objectives about the structure of a program (VIII and IX) are easy to assess (A:left).

D.1.2 David

David feels all learning objectives are better to assess individually (C:right) than through group work (C:left), except for the learning objective about structuring a program (VIII). He explained that he feels structuring a program well can best be learned through experience, or more specifically by having to work with a possibly bad structured program of another student. He thinks the other learning objectives are also especially possible to assess using a written test (B:left), and thus individually (C:left, "of course you make a written test individually" as translated into English). He also told us that whether he assess a learning objective using a written test (B:left) or through practical work (B:right) also depends on the material he has, so whether the material gives for example an example of a written test or an example of practical work.

Secondly, David said that he is of the opinion that all fundamentally new learning objectives in the core program are the same for each student no matter their profile (G:neutral), but that the way to achieving and assessing a learning objective differs for exact science students and not exact science students. He did however also say that programming (VII) is often better doable for NT students than CM students (G:left), which indicates that he (perhaps unknown) does see a difference in which learning objectives are more suitable for NT students (G:left) and which are more suitable for CM students (G:right).

Additionally, he feels that almost none of the fundamentally new learning objective is easy to assess (A:neutral/right). He explained that he has difficulties with thinking of contexts for new questions, and especially for questions assessing in-depth understanding (H). He mentioned that thinking of something new is often hard for him and that he therefore, when making questions, often uses previously made questions as inspiration. About adapting questions for new tests (L), he also told that the most difficult part for him is maintaining the same level of difficulty; changing a few

words or values does not adapt questions in his eyes, and if you really adapt a question (such as adding an option to a multiple choice question), the level of difficulty often changes. Therefore, he often finds it hard to adapt questions such that they assess the same learning objective with the same level of difficulty (L:left).

Thinking of a new context is not the only aspect of assessing in-depth understanding that David finds difficult (H:right); he also mentioned that he finds it hard to say whether a question does assess in-depth understanding or whether it is a difficult application question. Additionally, he thinks that it differs per student whether a question assesses insight or not, depending on whether they already have some in-depth understanding or not. Together, these aspects make creating questions that assess in-depth understanding difficult in his eyes for all learning objectives, in contrast to reproduction and application questions.

What David also indicated in the questionnaire, was that he knows (quite) little about the misconceptions and common mistakes (J:left) corresponding to the learning objectives of the fundamentally new (sub)domains. He explained that knowledge of misconceptions grows over time by seeing or looking at the answers of students to assessment questions and thus by experience. The novelty of the learning objectives is thus, in his opinion, the reason that he thinks to know little about the misconceptions.

D.1.3 James

In James' repertory grid, it can be seen that he rated quite a lot of learning objectives to be neutral regarding the distribution of the students' grades (K:neutral). He explained during the member checking interview that the students' scores are generally far apart (K:right) if a task assesses in-depth understanding, and that they are generally fairly close to each other (K:left) for tasks where the way the student represents their results determines part of their final score. The other learning objectives simply have a usual distribution of students' scores and therefore James rated them as neutral (K:neutral).

James also rated many learning objectives as neutral regarding whether they are more suitable for students with a CM or NT profile (G:neutral). He feels that most learning objectives are both suitable for students with a CM and NT profile, except for designing which he thinks is much more suitable for students with a CM profile than students with a NT profile (G:right). He indicated this in his repertory grid by rating the learning objective about designing, implementing and evaluating digital products as most suitable for CM students (G:right). He also indicated in his repertory grid that the learning objectives about data structures (IV), grammars (VI),

finite automatas (V), and correctness and efficiency (III) are better suitable for students with a NT profile (G:left).

James' ratings for the learning objective about imperative programming (VII) are (not considering for the construct (G) mentioned in the last paragraph) very distributed. He explained that he has a specific way to assess this learning objective which he always uses. He calls his assessment method an 'individual skill assignment' (in Dutch: "individuele vaardigheidsopdracht"). When filling in the repertory grid, he already indicated his use of these skill assignments and the decision was made that these assignments fall under individual (C:right), practical (B:right) work. Next to these skill assignments James does not use practical work (B:right) to assess topics. Such individual skill assignment consists in the case of imperative programming (VII) of three main tasks: reading a flowchart, translating a problem description (solution) into an algorithm (I), and programming an algorithm (VII). He explained that he currently assesses almost no knowledge of for example standard algorithms in these skill assignments; the focus is way more on letting the student think of a solution by themselves. He does however think he can adapt the skill assignment easily to also assess this knowledge by adding a control question.

Regarding his individual skill assignments, he also explained that he assess data structures (IV) also using such assignment; this way of assessing this learning objective ensures in his eyes that assessing in-depth understanding is easy (H:left) and having only one correct answer possible (I:right). He explained that the (formulation of the) question is often so specific that the correct answer(s) is also the most obvious caused by the three tasks split in the individual skill assignments. The question is however whether the student determines what data structure is most efficient and elegant (as mentioned in learning objective IV) by themselves in such assignment, as acknowledged by James.

In his repertory grid, James indicated to find the learning objective about digital products (X) difficult to assess, not only in general (A:left), but also for in-depth understanding specifically (H:right), regarding adapting questions (L:left) and regarding his knowledge of misconceptions and common mistakes (J:left). He explained that he does not know how to measure or assess design skills. He also told that he does not have affinity with design and misses a theoretical background or guidelines for that topic, and that he therefore does not feel comfortable assessing design skills and knowledge and only knows little about the corresponding misconceptions (J:left). He also explained that judging this learning objective costs him little time (F:left) due to not knowing corresponding criteria and that judging such assessment is for him therefore guesswork and thus a qualitative worthless assessment (in his own words).

Lastly, James also told us during the member checking interview that he feels quite uncomfortable with assessing grammars (VI), because he also misses the theoretical background for that topic and has a lack of conviction that it benefits the students. He does thus see a relation between having a theoretical background of a topic and how comfortable one feels with assessing that topic.

D.1.4 John

John rated the question whether the learning objective is better suitable for students with a NT or CM profile as neutral (G:neutral) for almost all learning objectives; only for the learning objectives about correctness and efficiency (III) and standard algorithms (II) he indicated that those are better suitable for students with a NT profile (G:left). During the MemberChecking interview, he explained that these ratings are more a thought than a wish to make (the assessment of) each learning objective suitable for each student. He tries to achieve this wish by using for example guided discovery teaching, or, in the case of algorithms, he divides the assessment questions in two tasks, namely formulating the algorithm in Dutch language and creating a flowchart for it (including values and variables). He mentioned that he is not certain to which extent he already has achieved this wish.

Secondly, he indicated in the repertory grid that creating an assessment question for the learning objective about structuring a program (VIII) takes him less time (D:more left) than for the other fundamentally new learning objectives. He explained that his assessment questions for this learning objective often simply consists of improving a given a poorly structured program; he takes a program and makes it less structured by replacing meaningful variable names with 'a', 'b', 'c', etc, and removing all line endings and tabs. Creating such question does not take a lot of time. By looking at the students' work, you can easily see what they are doing well and whether they know when a program is well structured, so he told us. He feels that other learning objectives can be more difficult to make a assessment question of, such as for the learning objective of translating a solution into an algorithm (I), because in such cases it is very difficult to find the right level of difficulty for the question and can therefore take much time to create an assessment question for (D:right) in his opinion.

Later in the member checking interview, he explained that the time it takes to create such question (D) depends on how this learning objective about translating a solution into an algorithm (I) is assessed. He feels it takes almost no time (D:left) if you assess it through practical work (B:right) because the teacher can just ask the students to think about something, ask him for approval, and do it. If you use a written test (B:left) however, you

also have to take the amount of time of the written test and the level of difficulty into account, and he feels that takes him a lot of time (D:right). It takes him this much time despite how much he thinks to know about the misconceptions and common mistakes (J:right) and that he thinks in-depth understanding is easy to assess for this understanding (H:left).

About his knowledge of the misconceptions (J) for this learning objective (I) and for the learning objective about imperative programming (VII), he explained that he has given an introductory programming course for a long time and therefore knows what goes wrong at the start. As example, he mentioned the difference between static and not static. He told us that such mistakes are also often mentioned in documents with misconceptions in programming. Regarding the assessment of the misconceptions, he explained that it is fairly easy to assess whether a student has such misconception by for example asking 'what does the keyword static mean?'. Such question is easy to judge (F:left) and still really indicates whether the student understands the concept or not, in his opinion.

Furthermore, John indicated in his repertory grid that only two learning objectives are better to assess using a written test (B:left) than through practical work (B:right), namely the ones about reasoning about correctness and efficiency (III), and recognizing and using standard algorithms (II). He feels that learning objectives about creating or reasoning are best to assess with open questions and thus (in his eyes) through practical work (B:right), and that other questions like checking whether a given algorithm gives a correct result, can be assessed more simply by asking a closed question (given an algorithm and an input, what is the result) and are therefore better to assess using a written test (B:left). If he has to assess questions about creating or reasoning using a written test (for example due to a high number of students), he tries to make his questions somewhere between open and closed, so that the students still have to think about something themselves. An example of such question is a drag and drop question.

Additionally, in Johns repertory grid it can be found that he rated how difficult it is to adapt questions for new tests for almost each learning objective as neutral (L:neutral). As mentioned before, he filled in this rating because it depends on how you assess it (using a written test or practical work, B). He also rated the most constructs as neutral for the learning objective about structuring a program (VIII). However, due to the time constraints of the member checking interview, no more information was collected about these two characteristic perceptions of John.

D.1.5 Michael

Michael indicated in his repertory grid that creating an assessment task for the learning objective about translating a solution into an algorithm (I)

takes him quite a lot of time (D:right), although he finds it quite easy to create an assessment question (A:left and L:right) and to assess for example in-depth understanding (H:left) for this learning objective. He explained that he quickly has an idea for an assessment question, but that creating the final assessment task also involves thinking about how difficult it is for students to interpret the task, whether you do or do not a lot of text for extra difficulty, and the formulation of the question. In Michaels opinion, the formulation of the question can make the question more difficult and thus more suitable for either NT or CM students (G). Additionally, he thinks that creating an assessment question for this learning objective (I) costs him especially quite a lot of time (D:right), because he always implements the algorithm beforehand to decrease the chance of a mistake in the question and to increase his knowledge of the level of difficulty of the question. He feels that you only really know the level of difficulty if you implement the algorithm yourself.

Perhaps the most outstanding aspect of Michaels repertory grid is that he rated the question about how suitable each learning objective is for students with a NT or CM profile as neutral (G:neutral) for each learning objective. He thinks that you can teach Computer Science in such a way that it is suitable for everyone, and that you also should do that (by for example taking the formulation of the question into account, as mentioned before). He did however indicate that if you make something too abstract (such as automatas (V) combined with grammars (VI), and (especially the lambda calculus part of) functional programming), students with a CM profile will find the topic more difficult than students with a NT profile. Therefore, he feels that more abstract learning objectives are more suitable for students with a NT profile than students with a CM profile (G:left). He emphasized that he filled in neutral (G:neutral) as it should be like that, and that in his opinion learning objectives which you can not make suitable for everyone should be thrown away.

He also explained that automatas (V) do not need to be taught in an abstract way; in his opinions automatas (V) can be taught and assessed quite easily by using imaginative examples such as coffee machines. Therefore, he indicated in his repertory grid this learning objective (V) is quite easy to assess in-depth understanding for (H:left) and creating assessment tasks does not take much time (D:left). He does however think teaching and assessing the learning objective about automatas (V) becomes much more difficult if you make it more abstract and use more difficult examples by for example combining grammars (VI) with it. Therefore, it depends on the level of difficulty and in-depth understanding expected from the students and the level of abstractness needed to be used, how hard it is to assess this learning objective (V) in his opinion. That is the reason that he indicated in his repertory grid that the learning objective about automatas (V) is

quite difficult to assess (A:right). Additionally, he indicated regarding this theoretical, abstract aspect of Computer Science that he probably has more difficulties with these learning objectives (automatas (V), and grammars (VI)), because he has done nothing with these topics since his graduation (about 30 years ago). He does however think he can have it in hand quickly because he is graduated theoretically. Therefore, he thinks there is a relation between experience and how difficult teaching and assessing a topic or learning objective is.

Furthermore, Michael indicated in his repertory grid that he finds it more difficult to create assessment tasks (e.g. A:right, H:right, and L:left) and judge students' work (E:right, and F:right) for the learning objective about comparing different data structures in terms of elegance and efficiency (IV) than for almost all other learning objectives. He explained that he finds assessing the students' understanding of the efficiency of data structures quite easy, because he has quite some experience with that aspect. He shows this by mentioning an example assessment question at once during the member checking interview, and explains that making many variations on that assessment question is also easy to do. Consequently, he explained that he finds assessing students' knowledge about the elegance (of for example data structures) much more difficult and that he can not at once come up with a good assessment question during the interview. He also explained that he feels many practical, non-theoretical computer scientists have no affinity with elegance (and therefore often postpone that subject) whereas theoretical and mathematical computer scientists do have affinity with elegance. He does thus think there is a difference between theoretical and non-theoretical computer scientists regarding for example there affinity with elegance. In this typification, he sees himself as part of the non-theoretical computer scientists despite his theoretical study background.

Michael also indicated to find creating assessment tasks (e.g. A:right, H:right, and L:left) and judging students' work (E:right, and F:right) more difficult for the learning objective about digital products (X) than for almost all other learning objectives. First of all, he explained that the term 'digital product' is not clear for him; is a digital product a robot, an automata, a computer, or something else? Secondly, he mentioned to have neither affinity nor experience with such digital products and that he therefore finds teaching and assessing these learning objectives less fun and more difficult. He acknowledged that he feels there is a relation between having less affinity with a learning objective and finding that learning objective more difficult to teach and/or assess. He did mention to have affinity with algorithms (II, and perhaps I and/or VII) and (the efficiency of) data structures (IV), for which he indeed indicated (in his repertory grid and member checking interview respectively) to find assessing those learning objectives fairly easy.

Lastly, he indicated in his repertory grid that all fundamentally new

learning objectives can best be assessed individually (C:right) and using a written test (B:left). For each learning objective, he gave exactly the same ratings for these constructs. During the member checking interview, Michael simply explained that he does not really believe in group work and that he therefore also does not really use group work within his company. Although nothing more was mentioned about these constructs during the interview, the similarity of the ratings seem to indicate that Michael feels that a relation exists between whether these two constructs (B and C)

D.1.6 William

When looking at Williams repertory grid, it immediately stands out that he rated only one learning objective-construct combination as neutral, namely whether the learning objective about imperative programming (VII) can better be assessed through practical work or using a written test (B:neutral). During the member checking interview, William indicated that for assessing this learning objective using practical work and a written test both have its advantages and disadvantages. Michael wants to get an objective picture of the programming knowledge and skills of the student, and give them a chance to show what they know and are able of. By using a written test (B:left), he feels he can see what an individual student knows, because the student can not ask others for help and all students work on the test for the same amount of time. Through practical work (B:right), you cannot really look at each specific individual, and it can also for example happen that a student loses a lot of time with debugging caused by a forgotten semi-colon. Thereby, one gets a less objective picture of the knowledge and skills of the student. Using practical work is in such case not the best option. However, if one also wants to assess skills like debugging, using practical work can also be the best option. In Williams eyes, assessing the learning objective about imperative programming is thus doable both through practical work and using a written test (B:neutral), depending on what one wants to assess exactly. Additionally, his explanation shows that William feels there is a relation between written tests (B:left) and individual work (C:right), and between practical work (B:right) and group work (C:left).

Furthermore, William rated one learning objective very different from the other learning objectives, namely the learning objective 'The student can design, implement and evaluate a digital product for a given situation' (X). In contrast to the other learning objectives, he rated this learning objective as better suitable for student with a CM profile than students with a NT profile (G:right), better to assess using group work than individually (C:left), and better to assess through practical work than using a written test (B:right). William explained that even though all students need to learn to program and apply the design process, for students with a CM profile technology is less prominent and therefore deepening regarding the technical

aspect of Computer Science is of less importance to them. Designing for example digital products is of more importance for students with a CM profile. He finds assessing students' knowledge of design processes is quite difficult, because most important are the way of working (namely iterative based on collected feedback) and the reasoning of the students. In addition, he explained that he thinks this learning objective can better be assessed with group work (C:left) and through practical work (B:right), because in practice designing is also often done in a group with other experts. William does thus feel that how an activity happens in practice is of importance for how that activity can best be assessed.

Three learning objectives stand out of Williams repertory grid for the same reason; William rated the learning objectives about translating a solution into an algorithm (I), recognizing and using standard algorithms (II), and determining whether input is correct given a grammar and an input (VI) as easy to assess (A:left), but also as difficult to assess in-depth understanding (H:right) and adapting questions for (L:left) and as knowing little about the corresponding misconceptions and common mistakes (J:left). He explained that he finds for example assessing the learning objective about standard algorithms quite easy (A:left), because one can simply describe a bubble sort and ask the students to transform that into a flowchart, or ask the students where the standard algorithm is shown in a given algorithm (in the form of either pseudocode or a flowchart). He does however finds it rather difficult to find several algorithms which are similar regarding their level of difficulty but have a different purpose or context (L:left).

William explained about these learning objectives that he finds adapting questions difficult (L:left), because you have to think of a context which is appealing to the students. When you ask a question without context, the question is often not interesting for the students. When you do use a context, the question becomes more interesting and relevant for the students in his eyes, because they want to see what they can achieve with their knowledge about a topic. Additionally, he thinks that contexts like determining the shortest path from home to school is also not interesting for students, because they will find it anyway. Therefore, such more relevant context is also not always ideal for an assessment question. William feels thus that thinking of a context which is appealing to students is rather difficult, and that, in addition to ensuring the same level of difficulty, makes adapting questions difficult (L:left) in his opinion.

Furthermore, William finds judging students' work for these three learning objectives sometimes quite difficult (E:right and/or F:right), because a student can give an answer going in the right direction but with some mistake(s) making their answer only partly correct. He did not indicate this opinion in his repertory grid. Lastly, he feels to know only little about the misconceptions and common mistakes of these learning objectives

(J:left), because he has only one source for the mistakes students make in flowcharts (namely a study of Ebrahim) for example. In the direction of pseudocode instead of flowcharts more is known about the misconceptions, because they are quite similar to the misconceptions of programming and a lot is known about programming misconceptions. He does however feel that creating a question which provokes misconceptions and thus assesses them explicitly is (still) very difficult, both when using flowcharts and when using pseudocode. To conclude, he thinks that an assessment question without context is easy to create (A:left) for the learning objectives about translating a solution into an algorithm (I), standard algorithms (II) and grammars (VI), but that it becomes more difficult when you want to assess more in-depth understanding (H:right) or when you need to make two different questions regarding their context with the same level of difficult which are appealing to the students (L:left).

D.2 (Dis)similar perceptions of learning objectives

D.2.1 Charles

Charles seems to make a distinction between learning objectives about implementation and more theoretical learning objectives; he rated the learning objectives about imperative programming (VII), digital products (X), standard algorithms (II), data structures (IV) and describing the behavior of a system as a finite automata (V) as best to assess using group work (C:left) and not time-consuming to judge (F:neutral) whereas he rated the learning objectives about correctness and efficiency (III), explaining structure and behavior (IX), translating a solution into an algorithm (I), grammar (VI) and structuring a program (VIII) as best to assess individually and time-consuming to judge. An in-depth explanation of the groupings as well as a visual overview can be found in section D.3.1.

Within the group with learning objectives about implementation, two learning objectives were rated very similarly by Charles, namely the one about imperative programming (VII) and the one about designing, implementing and evaluating a digital product (X). Charles characterized these learning objectives about concrete implementation as best to assess with practical work (B:right).

Charles also rated two elements within the group of more theoretical learning objectives very similarly. These two learning objectives are reasoning about the correctness and efficiency of a given program (III) and explaining the structure and behavior of a given program (IX). Both these learning objectives are not only quite theoretical, but also contain the aspect of understanding and describing part of a given program. These two learning

objectives were characterized by being best to assess using a written test (B:left), difficult to judge (E:right) and easy to assess (A:left).

D.2.2 David

Whereas Charles' grouping of learning objectives seemed to have (almost) no relation with the (sub)domains to which the learning objectives correspond, Davids clusters showed that he sees the learning objectives of domain A and D as more closely related to each other than to the learning objectives of domain B. All learning objectives of domain A and D - thus the ones about imperative programming (VII), structuring a program (VIII), explaining the structure and behavior of a given program (IX), and digital products (X) - can be found in one cluster, implying that he perceived these as being similar regarding their assessment. He rated these learning objectives as difficult to assess (A:right), and as more difficult to assess in-depth understanding for (H:more right), and having students' scores which are further apart (K:more right) than the quite similarly rated learning objectives from domain B.

These 'quite similarly rated' learning objectives from domain B comprise the learning objectives about translating a solution into an algorithm (I), recognizing and using standard algorithms (II), and describing a (physical) system as a finite automata (V). David rated both these learning objectives from domain B and the ones from domain A and D as difficult to judge objectively (E:right) and suiting students with a CM profile (G:neutral/right). From these elements, he rated the learning objectives about algorithms (I and II) very similarly, characterized by better to assess using a written test (B:more left) and more time to create an assessment for (D:more right) than the other learning objectives about algorithms and programs.

Quite different from these learning objectives regarding assessment, according to David, are the learning objectives about correctness and efficiency (III), grammars (VI) and data structures (IV). He rated these learning objectives as easily to judge unambiguously (E:left) and suiting students with a NT profile better than students with a CM profile (G:left). He rated two of these learning objectives about the math or reasoning behind Computer Science as being very similar regarding assessment, namely the ones about correctness and efficiency (III) and grammars (VI). In contrast to the element about data structures (IV), he saw these learning objectives as having generally only one correct answer possible (I:right) and as not much work to create an assessment for (D:left).

D.2.3 James

James sees two groups of learning objectives as being very similar regarding assessment. In his groupings there seems to be no relation between

(sub)domains and the groups. Two learning objectives are not part of one of these groups, which implies that James rated them to be quite dissimilar compared to the learning objectives in the groups, but also not similar to each other. These learning objectives are the ones about imperative programming (VII) and digital products (X).

The first group of similar learning objectives, according to James, consists of the learning objectives about reasoning about the correctness and efficiency of a given program (III), describing the behavior of a system as a finite automata (V), explaining the structure and behavior of a given program (IX), and recognizing and using standard algorithms (II). In contrast to the other elements, he rated these learning objectives as easy to assess (A:left), easy to adapt questions for (L:right), easy to judge objectively (E:left), quickly to judge (F:left), and having generally only one correct answer possible (I:right). Moreover, he rated the describing learning objectives within this group (III, V and X) as being very similar, characterized by them being easy to assess in-depth understanding for (H:left) and more work to create an assessment for (D:more right) than the learning objective about recognizing and using standard algorithms (II).

The second group comprises the learning objectives about comparing and contrasting data structures (IV), structuring a program (VIII), translating a solution to an algorithm (I), and grammars (VI). These elements are rated similar in the sense that James sees them all as difficult to assess (A:right), not easy to judge objectively (E:neutral/right) and difficult to adapt questions for (L:left). James rated the learning objectives about data structures (IV) and structuring a program (VIII) to be especially very similar, because they are very suitable for both CM and NT students (G:neutral) and have generally several correct answers possible (I:left), according to his ratings.

D.2.4 John

Johns ratings showed a clear relation between the existing groups of similarly rated learning objectives and the (sub)domains to which these learning objectives correspond. The only learning objective from domain A, namely the one about digital products (X), has a cluster of its own, characterized by being best to assess through group work (C:left) and easy to adapt questions for (L:right).

Furthermore, all learning objectives from domain D can be found in one cluster. This group is characterized by being not difficult to assess (A:left/neutral), not difficult to judge objectively (E:left/neutral) and quickly to judge (F:left). This group also contains the only learning objective not in the same cluster as the other learning objective from its (sub)domain, namely translating a solution into an algorithm (I) from subdomain B1.

Moreover, he rated this learning objective to be very similar to the learning objective about explaining the structure and behavior of a given program (IX), characterized by being better to assess individually (C:more right) and more easy to assess in-depth understanding for (H:more left) than the other elements in this group. The relation between the learning objective from subdomain B1 and domain D seems to be that they are both about (not web) programming.

All other elements from domain B can be found in the group characterized by having generally only one correct answer possible (I:right). This group contains a clear subgroup including all learning objectives from subdomains B2-B4 and none from subdomain B1. John saw these learning objectives as not quickly to judge (F:more right), better to assess individually (C:more right), and more difficult to assess in-depth understanding for (H:more right) than the learning objectives from subdomain B1 in this group.

D.2.5 Michael

Michael rated two learning objectives to be very similar to each other and dissimilar to all other learning objectives, namely the learning objectives about comparing and contrasting data structures (IV) and designing, implementing and evaluating digital products (X). In contrast to the other learning objectives, he rated these learning objectives as difficult to assess (A:right), difficult to judge objectively (E:right), and time-consuming to create an assessment for (D:right). In addition, he rated these learning objectives as time-consuming to judge (F:right) and difficult to assess in-depth understanding for (H:right).

The other learning objectives involve more actual (not web) programming and explaining programs. These are rated by Michael as easy to assess (A:left), easy to judge unambiguously (E:left) and quickly to create an assessment for (D:left). Within this huge group, Michael rated the learning objectives about explaining a program (II, III, V, VI, VII and IX) to be more similar to each other than the learning objectives about programming itself, namely translating a solution into an algorithm (I) and structuring a program (VIII). He sees the learning objective about explaining a program as having generally only one correct answer (I:right) and he also thinks to know more about their misconceptions and common mistakes (J:more right).

Michael rated two of the learning objectives about explaining a program more similar to each other than the other elements, namely reasoning about the correctness and efficiency of a given program (III) and explaining the structure and behavior of a given program (IX). There is however not one identifiable difference between these learning objectives and the others.

D.2.6 William

William sees the learning objective from domain A to be very different from the learning objectives from domain B and D; he rated the learning objective about digital products (X, domain A) as best suitable for CM students (G:right), best to assess through group work (C:left) and practical work (B:right), and difficult to judge objectively (E:right) whereas he rated the learning objectives from domains B and D as best suitable for NT students (G:left), best to assess individually (C:right) and using a written test (B:left), and easy to judge unambiguously (E:left).

Within the group of domain B and D, William rated several groups of learning objectives to be similar to each other regarding assessment. For example, he rated the learning objectives about imperative programming given an algorithm (VII) and explaining the structure and behavior of a program (IX) - both about the relation between an algorithm and a program - very similar, characterized by having generally several correct answers possible (I:left) and easy to assess in-depth understanding for (H:left).

Additionally, he rated the learning objectives about efficiency, namely reasoning about the correctness and efficiency of a given program (III) and comparing and contrasting data structures in terms of elegance and efficiency (IV), very alike. The only real difference between these learning objectives is that William rated the learning objective about correctness and efficiency (III) as quite difficult to assess in-depth understanding for (H:right) whereas he rated the learning objective about data structures (IV) as easy to assess in-depth understanding for (H:left).

Lastly, he rated the elements about recognizing and using standard algorithms (II) and checking whether input is correct given a grammar (VI) very similar. William did not only see no real difference between these learning objectives regarding assessment, but also characterized these learning objectives (in contrast to the other learning objectives from domain B and D) as being both easy to assess (A:left) and having students' scores which are generally close to each other (K:left).

D.3 Description of Focus cluster

D.3.1 Charles

The cluster analysis of Charles' raw grid (see Figure E.7) showed two large groups or clusters of closely matching elements. The first group consists of the learning objectives about standard algorithms (II), data structures (IV), automatas (V), imperative programming (VII), and digital products (X). The second group consists of the learning objectives about translating a solution into an algorithm (I), correctness and efficiency (III),

grammars (VI), structuring a program (VIII), and explaining the structure and behavior of a program (IX). To understand the grounds on which Charles discriminated between these two groups of learning objectives, we examined his ratings of these learning objectives on the various constructs. The main difference between these groups is that the first group is seen as better to assess with group work (C:left) whereas the second group is seen as better to be assessed individually (C:right). In addition, most of the learning objectives of the first group work were rated as neither quickly nor time-consuming to judge (F:neutral) whereas the elements in the second group which were generally seen as time-consuming to judge (F:right).

In the second group, one clear subgroup of two very similarly rated elements can be seen, namely the learning objectives 'The student can reason about the correctness and efficiency of a given program' (III) 'The student can explain the structure and behavior of a given program' (IX). The main difference between the ratings of these elements and the other elements in the second group is that learning objectives III and IX are seen as best to assess using a written test (B:left) instead of through practical work (B:right) as well as quite difficult to judge objectively (E:right) instead of easy to judge ambiguously (E:left). On the first sight, Charles' raw grid also shows that the learning objectives about data structures (IV) and finite automatas (V) from the 'first' group are quite similar to these learning objectives about correctness and efficiency (III) and explaining the structure and behavior of a program (IX). However, these elements from the 'first' group are not only rated as best to assess with group work (C:left), but also as difficult to assess (A:right), whereas Elements III and IX are seen as best to be assessed individually (C:right) and easy to assess (A:left).

Within the first group, the learning objectives 'The student can, for a given algorithm, develop a program in an imperative programming language.' (VII) and 'The student can design, implement and evaluate a digital product for a given situation.' (X) are rated quite similarly. Compared to the other learning objectives in the first group, Charles agreed entirely with the construct 'This learning goal can best be assessed through practical work' (B:right) whereas he did not totally agree with this construct for the other elements in the first group. Moreover, Charles even indicated to see the learning objective 'The student can (for a given algorithm) compare and contrast different options for data structures in terms of elegance and efficiency.' (IV) as best to assess using a written test (B:left) instead of through practical work (B:right).

In the figure below, an overview of which learning objectives Charles rated similarly and why. If learning objectives are shown together in an oval, they are rated similarly. The rectangle corresponding to that oval shows the main characteristics of the learning objectives in that group according to

Charles. The colors and lines are only used to make the overview more clear.

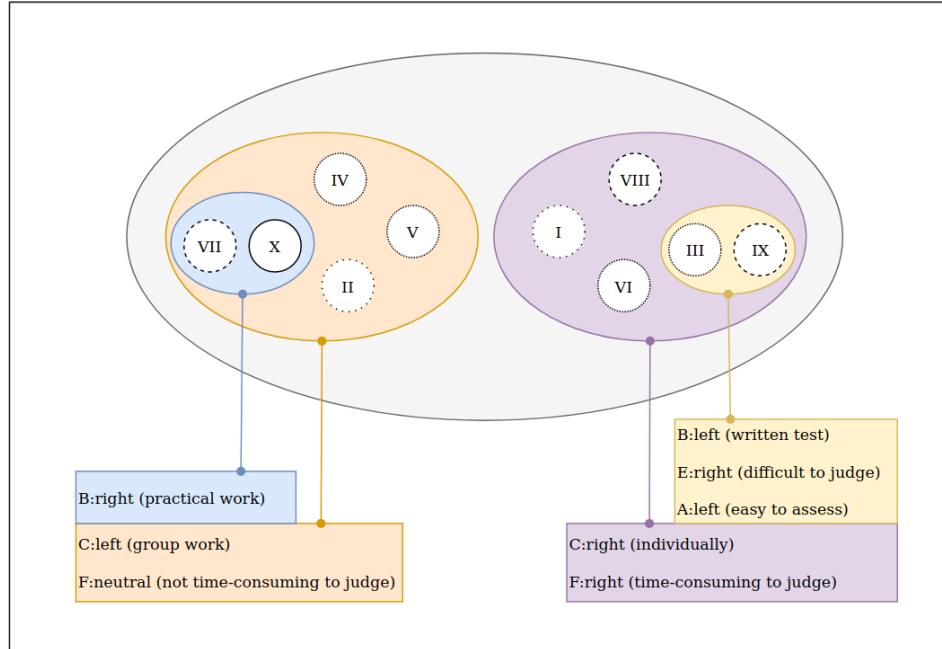


Figure D.1: Overview of Charles' grouping of learning objectives and main grounds

D.3.2 David

In Figure E.8, the Focus Cluster analysis of Davids raw grid is shown. In this analysis, two large groups of closely matching elements can be seen, each of which consists of several subgroups. Of these large groups, one consists of the three elements about correctness and efficiency (III), data structures (IV) and grammars (VI). All elements in this group are learning objectives from Domain B. The other large group consists of all other elements (seven in total).

To understand the grounds on which David discriminated between these two groups, we examined his ratings of these learning objectives on the various constructs. The main differences between these groups are that elements in the smaller group are seen by David as easily to judge unambiguously (E:left) and particularly suiting students with a NT profile (G:left) whereas the elements in the larger group were rated as difficult to judge objectively (E:right) and not particularly suiting students with a NT profile more than students with a CM profile (G:neutral/right).

The group with three elements contains one clear subgroup consisting of

the learning objective about correctness and efficiency (III) and grammars (VI). Compared to the learning objective about data structures (IV, the other element in the group), these elements are seen as having, in general, only one correct answer possible (I:right) instead of several correct answers (I:left). In addition, David rated the learning objectives in the subgroup (correctness & efficiency, III and grammars, VI) as not much work to create an assessment for (D:left) whereas the learning objective not in the subgroup (data structures, IV) is rated as neutral by David and thus as not taking either a lot of time or as not much work [to create] an assessment for (D:neutral).

The group with seven elements contains various subgroups. The first subgroup contains the learning objectives about translating a solution into an algorithm (I) and about standard algorithms (II), both from subdomain B1. Compared to the other elements in the group, these elements are rated as taking more time to create an assessment for (D:more to the right), less suitable to assess using a written test (B:more to the left) instead of through practical work (B:right), and less easy to adapt questions for (L:more to the left). These differences are all quite subtle but together ensure the creation of this subgroup.

The second subgroup contains the learning objectives about imperative programming (VII), structuring a program (VIII), explaining structure and behavior of a program (IX), and designing, implementing and evaluating a digital product (X). None of these elements are from Domain B, instead they are part of domains A and D. Compared to the other elements of the larger group, these elements are generally seen by David as more difficult to assess indepth understanding for (H:more to the right) and as having students' scores which are generally further apart (K:more to the right) than the students' scores of the other elements. In addition, these elements are more often rated as being more difficult to assess (A:right) and as best to assess through practical work (B:right).

This second subgroup consists of two even smaller subgroups, namely a subgroup with the learning objectives about structuring a program (VIII) and digital products (X), and a subgroup with the learning objectives about imperative programming (VII) and explaining the structure and behavior of a program (IX). Both elements from this last subgroup are from domain D. The main difference between these small subgroups is that the elements in the first subgroup are both rated by David as certainly best to assess through practical work (B:right) whereas the elements in the second subgroup are rated as absolutely best to assess using a written test (B:left).

In the figure below, an overview of which learning objectives David rated similarly and why. If learning objectives are shown together in an oval, they are rated similarly. The rectangle corresponding to that oval shows the main

characteristics of the learning objectives in that group according to David. The colors and lines are only used to make the overview more clear.

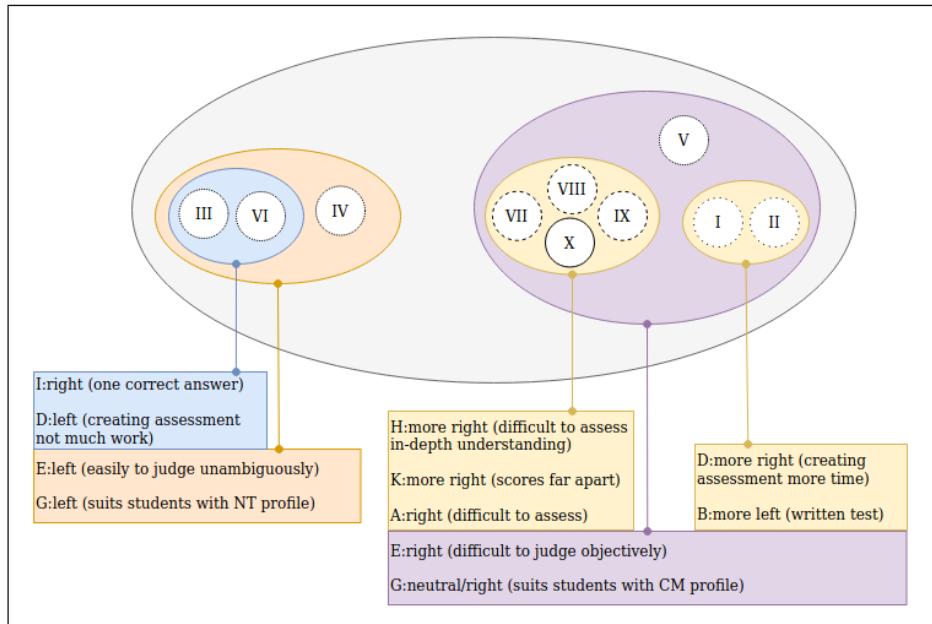


Figure D.2: Overview of Davids grouping of learning objectives and main grounds

D.3.3 James

The Focus cluster analysis of James' raw grid (see Figure E.9) shows two clear clusters: the first group consists of the learning objectives about standard algorithms (II), reasoning about correctness and efficiency (III), finite automatas (V) and explaining structure and behavior of a program (IX), and the second of the learning objectives about translating a solution into an algorithm (I), data structures (IV), grammars (VI) and structuring a program (VIII). The learning objectives imperative programming (VII) and digital products (X) are not part of one of these clusters (and are also not rated similarly). This observation implies that James saw these learning objectives as quite different in assessment as compared to the other learning objectives. James rated the elements in the first group, in contrast the other learning objectives, as easy to assess (A:left), adapt questions for (L:right), and judge objectively (E:left), as well as quickly to judge (F:left) and having generally one one correct answer possible (I:right). The learning objectives in the second group are characterized, according to James, by being difficult to assess (A:right), difficult to adapt questions for (L:left), and not easy to judge objectively (E:neutral/right).

The Focus cluster analysis of James' raw grid also showed that the first group contains a clear subgroup consisting of the learning objectives about correctness and efficiency (III), finite automatas (V), and explaining the structure and behavior of a program (IX). Compared to the other learning objectives, these learning objectives (III, V and IX) are about explaining, reasoning and describing which all involve describing as an essential part of the learning objective. In all other learning objectives, describing is not necessarily part of the learning objective as formulated in the repertory grid. The presence of this grouping in James' analysed grid shows that James perceived describing learning objectives to be very similar with respect to each other, no matter what exactly was to be described.

To understand the ground on which James discriminated between these types of learning objectives, his ratings of the describing learning objectives on the various constructs. In addition to the other characteristics of the first group, it was found that James saw the describing learning objectives mainly as easy to assess in-depth understanding for (H:left), and as more work to create an assessment for than for the other learning objectives (D:more right).

Additionally, the Focus cluster of James showed that the second cluster also contains a subgroup. This subgroup contains the learning objectives about comparing and contrasting different options for data structures (IV) and structuring a program (VIII). In contrast to the other elements in the second group, James rated these elements as suiting both students with an NT and a CM profile (G:neutral) instead of better suiting students with a NT profile (G:left), and as having in general several correct answers possible (I:left) as opposed to having only one correct answer possible in general (I:right).

In the figure below, an overview of which learning objectives James rated similarly and why. If learning objectives are shown together in an oval, they are rated similarly. The rectangle corresponding to that oval shows the main characteristics of the learning objectives in that group according to James. The colors and lines are only used to make the overview more clear.

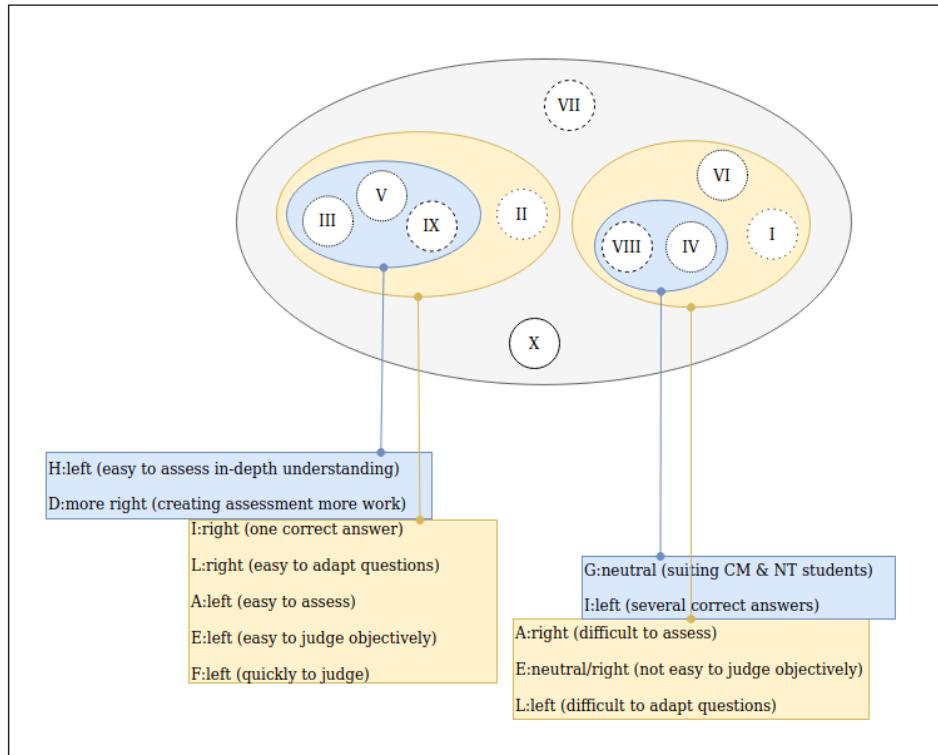


Figure D.3: Overview of James' grouping of learning objectives and main grounds

D.3.4 John

The cluster analysis of Johns raw grid (Figure E.10) showed three main groups consisting of the following learning objectives:

- (i) – (X) digital products
- (ii) – (II) standard algorithms
 - (III) correctness and efficiency
 - (IV) data structures
 - (V) automatas
 - (VI) grammars
- (iii) – (I) translating a solution into an algorithm
 - (VII) imperative programming
 - (VIII) structuring a program
 - (IX) explaining the structure and behavior of a given program

The groups (ii) and (iii) both comprise one or more subgroups which will be explored later on. Group (i) consists of only elements from Domain A, group (ii) of elements only from Domain B, and group (iii) contains all elements from Domain D and one element from Domain B, namely the learning objective about translating a solution into an algorithm (I). This grouping implies that John saw the learning objective about digital products (X) as very different from all other learning objectives. In addition, John saw the learning objective about translating a solution into an algorithm (I) as quite different from the other learning objectives from Domain B, and as more related to the learning objectives from Domain D.

To understand the grounds on which John discriminated between these three groups, we examined his ratings of these comprising elements on the various constructs. The only learning objective in group (i) about digital products (X) is, in contrast to the elements from the other groups, seen as best to assess with group work (C:left), easy to adapt questions for (L:right). Furthermore, the learning objective about digital products (X) is seen as generally having several correct answers possible (I:left) whereas the elements from group (ii) are seen as generally having only one correct answer possible (I:right). In contrast to the elements from group (iii), the elements from group (i) and (ii) are seen as time-consuming to judge students' work for (F:right), difficult to assess (A:right) and difficult to judge objectively (E:right).

Group (ii) contains one subgroup which consists of the learning objectives about data structures (IV), automatas (V), grammars (VI). These learning objectives are part of subdomains B2, B3 and B4 whereas the other elements from group (ii), namely the learning objectives about standard algorithms (II) and correctness and efficiency (III), are both from subdomain B1. Compared to these elements, the learning objectives about data structures (IV), finite automatas (V) and grammars (VI) are seen as less suitable to assess individually (C:more to the right), more difficult to assess in-depth understanding for (H:more to the right), and more time-consuming to judge students' work for (F:more to the right). In addition, the learning objectives about standard algorithms (II) and correctness and efficiency (III) are seen by John as more suitable for students with a NT profile (G:more to the left).

Group (iii) contains a subgroup which contains a subgroup itself. The inner subgroup consists of the learning objectives about translating a solution into an algorithm (I) and explaining the structure and behavior of a given program (IX), the outer subgroup additionally contains the learning objective about imperative programming (VII). Compared to the elements in the inner subgroup (I and IX), the learning objective about imperative programming (VII) is rated as more difficult to assess in-depth understanding for (H:more to the right) and more suitable to assess

individually than with group work (C:more to the right). Only one element from group (iii) is not part of the outer subgroup, namely the learning objective about structuring a program (VIII). Compared to the elements in the outer subgroup, this element is seen as taking more time to create an assessment for (D:more to the right), more difficult to assess (A:more to the right) and even more – namely entirely - suitable to assess individually (C:right).

In the figure below, an overview of which learning objectives John rated similarly and why. If learning objectives are shown together in an oval, they are rated similarly. The rectangle corresponding to that oval shows the main characteristics of the learning objectives in that group according to John. The colors and lines are only used to make the overview more clear.

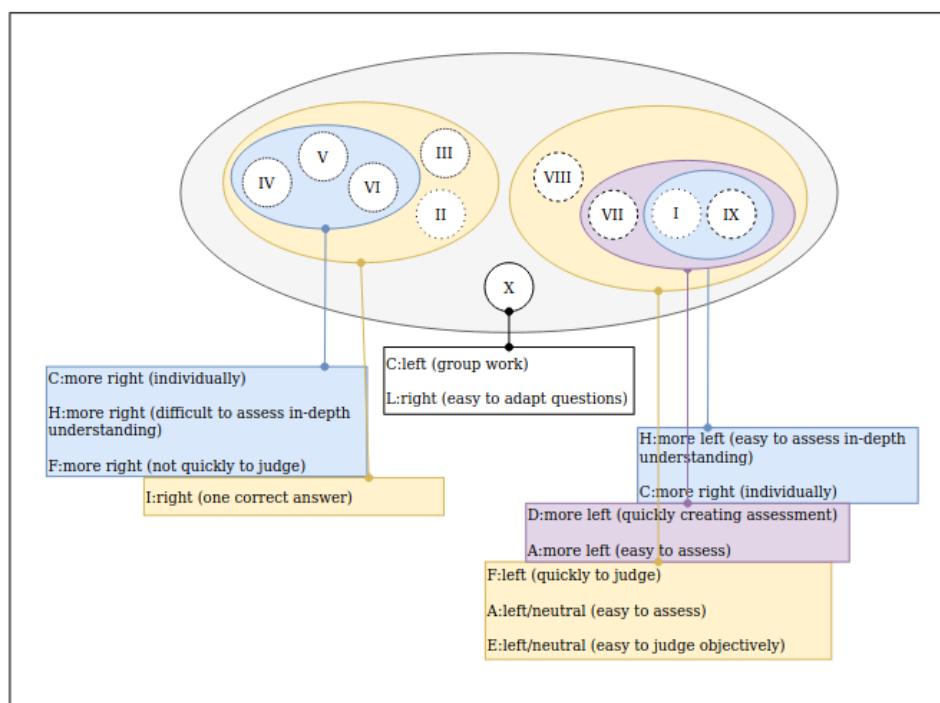


Figure D.4: Overview of Johns grouping of learning objectives and main grounds

D.3.5 Michael

The Focus Cluster analysis of Michaels raw grid is shown in Figure E.11. This analysis shows two main clusters: one cluster consists of the learning objectives about data structures (IV) and digital products (X), the other cluster contains all other elements. To understand the grounds on which Michael discriminated between these two clusters of learning objectives. A

lot of differences exist between the two groups; the elements in the first group are, in contrast to the elements in the second group, seen by Michael as difficult to assess in-depth understanding for (H:right) and time-consuming to judge students' work for (F:right). In addition, the elements in the small group are rated by Michael as difficult to assess (A:right), difficult to judge objectively (E:right) and taking a lot of time to create an assessment for (D:right) whereas most elements in the large group are rated as easy to assess (A:left), easily to judge unambiguously (E:left) and not much work to create an assessment for (D:left). Furthermore, Michael rated the learning objectives about data structures (IV) and digital products (X) as more difficult to adapt questions for new tests for than all other elements (L:more to the left).

The large main group contains one clear subgroup consisting of the learning objectives about standard algorithms (II), correctness and efficiency (III), finite automatas (V), grammars (VI), imperative programming (VII), and explaining structure and behavior (IX). To understand the reasons on which he made a distinction between this subgroup and the other elements, we will again take a look at the differences between this participants' ratings of this subgroup and the other elements. Michael believes to know more about the misconceptions and common mistakes of the learning objectives from the subgroup than of the learning objectives not in the subgroup (J:more to the right). Additionally, Michael rated most elements in the subgroup as generally having only one correct answer possible (I:right) whereas he rated the other elements as generally having several correct answers possible (I:left).

The above mentioned subgroup also contains a subgroup. This inner subgroup consists of the learning objectives about reasoning about the correctness and efficiency of a given program (III) and explaining the structure and behavior of a given program (IX), which are rated very similarly. Although there is not one identifiable difference with the other elements in the outer subgroup, they are rated exactly the same for all constructs except for constructs I and A; both elements are rated by Michael as easy to assess (A:left) and having only one correct answer possible (I:right). The learning objective about explaining structure and behavior (IX) is however even more easy to assess (A:more to the left) and has some more cases where several correct answers are possible (I:left) than the learning objective about correctness and efficiency (III), according to Michael.

In the figure below, an overview of which learning objectives Michael rated similarly and why. If learning objectives are shown together in an oval, they are rated similarly. The rectangle corresponding to that oval shows the main characteristics of the learning objectives in that group according to

Michael. The colors and lines are only used to make the overview more clear.

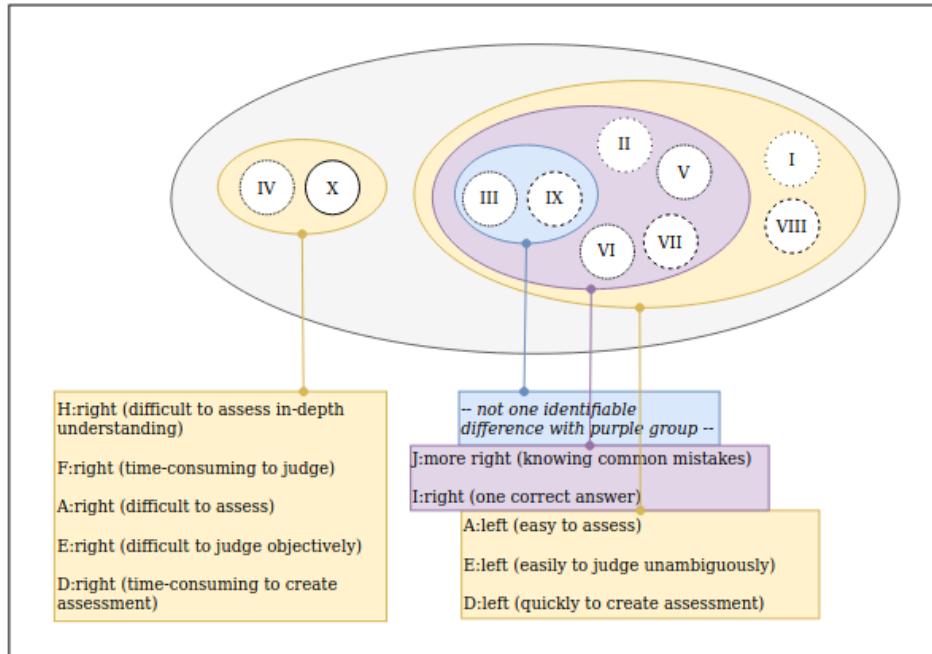


Figure D.5: Overview of Michaels grouping of learning objectives and main grounds

D.3.6 William

In Figure E.12, the Focus Cluster analysis of Williams raw grid is shown. In this analysis, the learning objective about digital products (X) is rated as very different from all other elements. To understand the ground on which William discriminated between this learning objective and the other elements, we examined the differences between his ratings of Element X and the other elements on the various constructs. Learning objective X is rated by William as particularly suited for students with a CM profile (G:right), best to assess with group work (C:left), difficult to judge students' work objectively for (E:right), and best to assess through practical work (B:right). In contrast to the learning objective about digital products (X), the other learning objectives are seen by William as particularly suited for students with a NT profile (G:left), best to assess individually (C:right), easy to judge students' work unambiguously for (E:left), and best to assess using a written test (B:left).

The other elements - thus all elements except for the learning objective about digital products (X) - in the analysis are divided into three large

clusters. The first cluster consists of the learning objectives about imperative programming (VII) and explaining structure and behavior (IX), the second of the learning objectives about correctness and efficiency (III), data structures (IV), automatas (V) and structuring a program (VIII), and the third group consists of the learning objectives about translating a solution into an algorithm (I), standard algorithms (II) and grammars (VI). In contrast to almost all elements from the second and third group, the elements in the first group are seen as easily to assess in-depth understanding for (H:left) and having generally several correct answers possible (I:left). The third group differs especially from the other groups in the difficulty to assess it (A); the elements in the third group are seen by William as easy to assess (A:left) whereas the elements in the other clusters are rated as hard to assess (A:right). The elements in the second group are similar in the sense that their constructs are rated the opposite of the classifying constructs of the first group, the third group, and Element X about digital products.

The second group contains a subgroup consisting of the learning objectives about reasoning about the correctness and efficiency of a given program (III) and comparing and contrasting different options for data structures (IV). William believes to know little about the misconceptions and common mistakes (J:left) for these learning objectives according to his ratings. In contrast, he believes that he knows which misconceptions and learning objectives are involved (J:right) for the other learning objectives in the second group, namely the learning objectives about describing the behavior of a (physical) system as a finite automata (V) and structuring a program (VIII).

The third group also contains a clear subgroup. This subgroup contains the learning objectives about standard algorithms (II) and grammars (VI). In contrast to the other element in the third group, namely the learning objective about translating a solution into an algorithm (I), the elements in the subgroup are seen by William as having students' scores which are generally fairly close to each other (K:left).

In the figure below, an overview of which learning objectives William rated similarly and why. If learning objectives are shown together in an oval, they are rated similarly. The rectangle corresponding to that oval shows the main characteristics of the learning objectives in that group according to William. The colors and lines are only used to make the overview more clear.

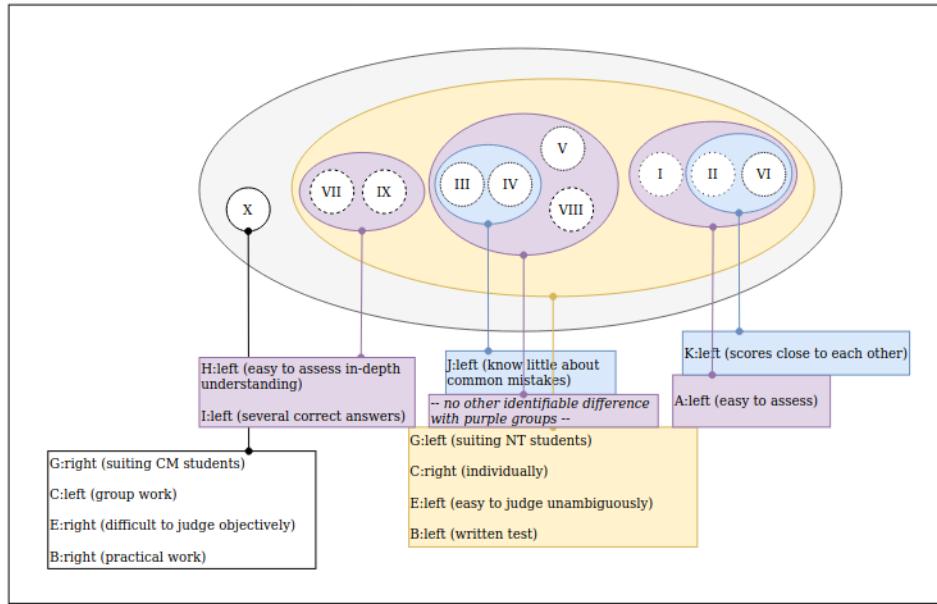


Figure D.6: Overview of Williams grouping of learning objectives and main grounds

Appendix E

Repertory grids of participants

E.1 Raw repertory grids

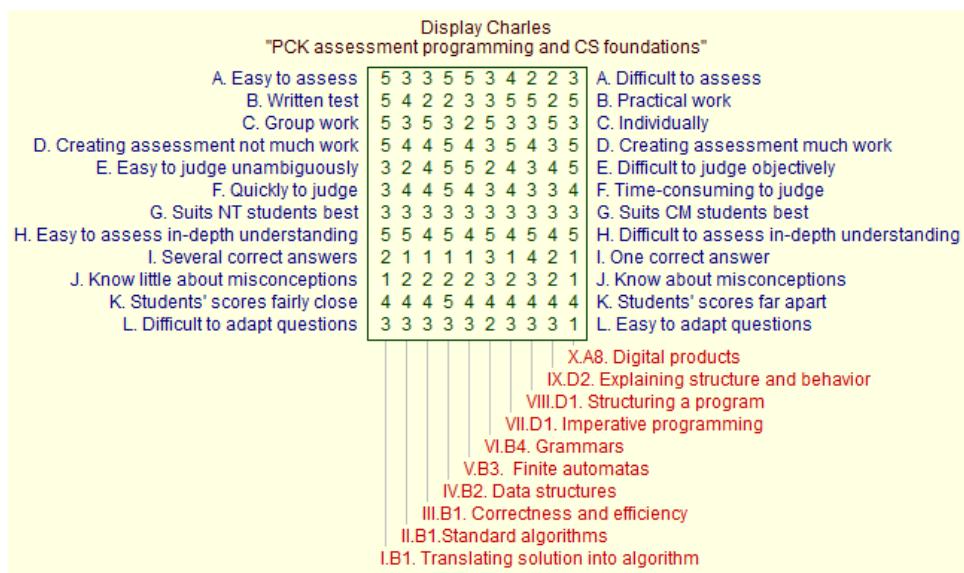


Figure E.1: Charles' repertory grid

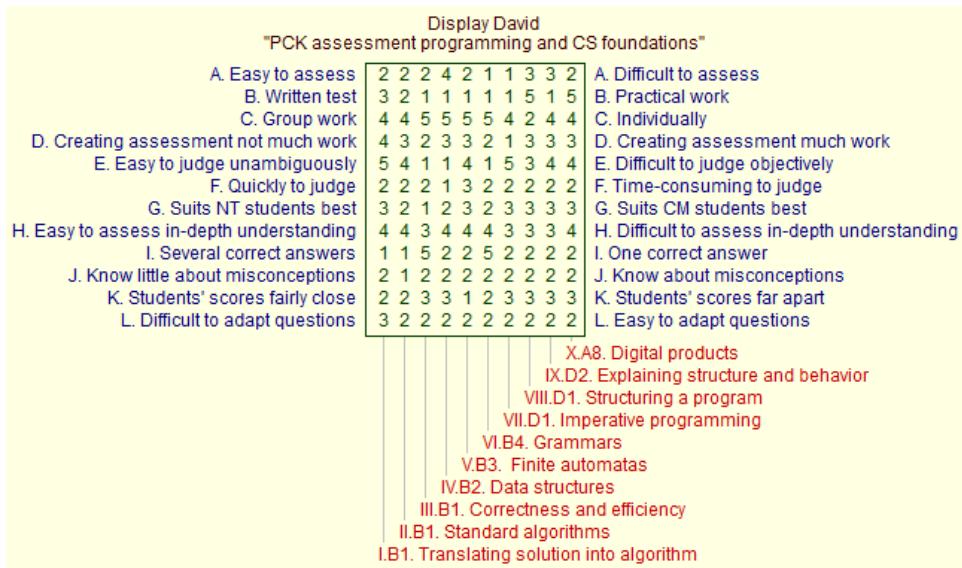


Figure E.2: Davids repertory grid

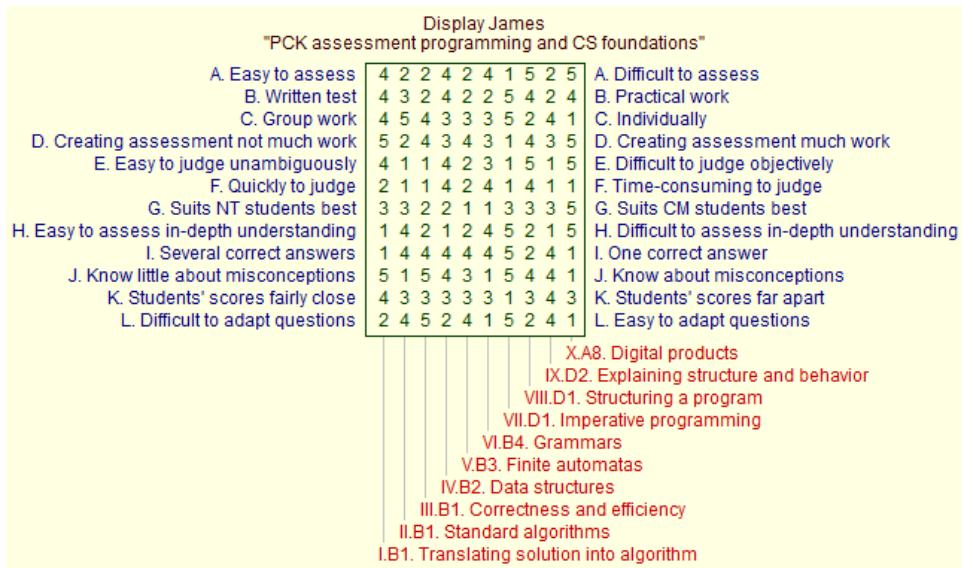


Figure E.3: James' repertory grid

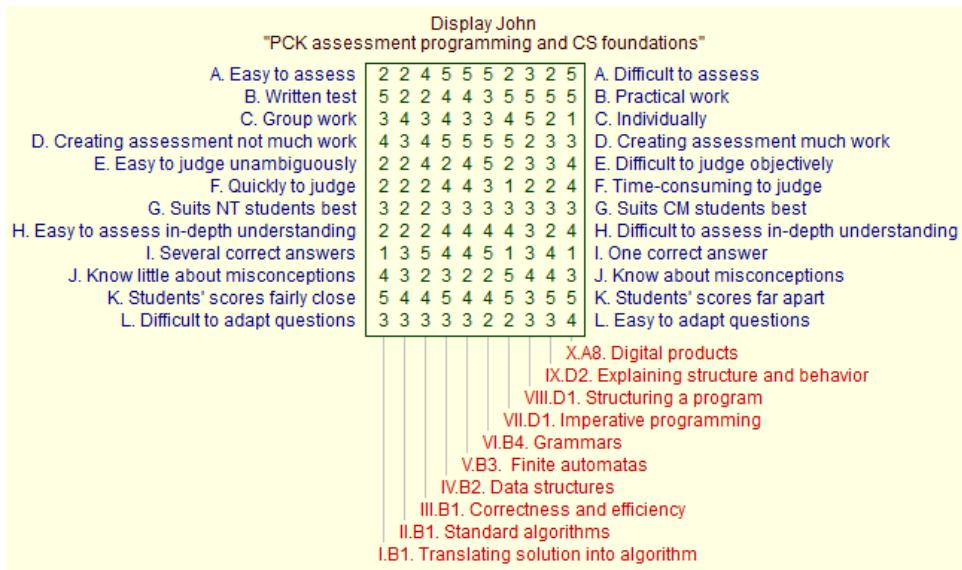


Figure E.4: Johns repertory grid

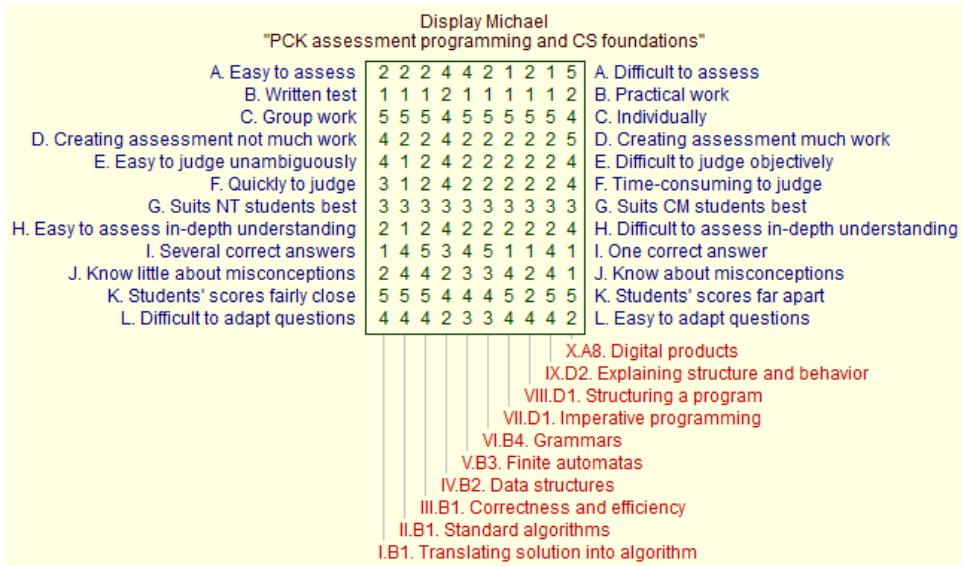


Figure E.5: Michaels repertory grid

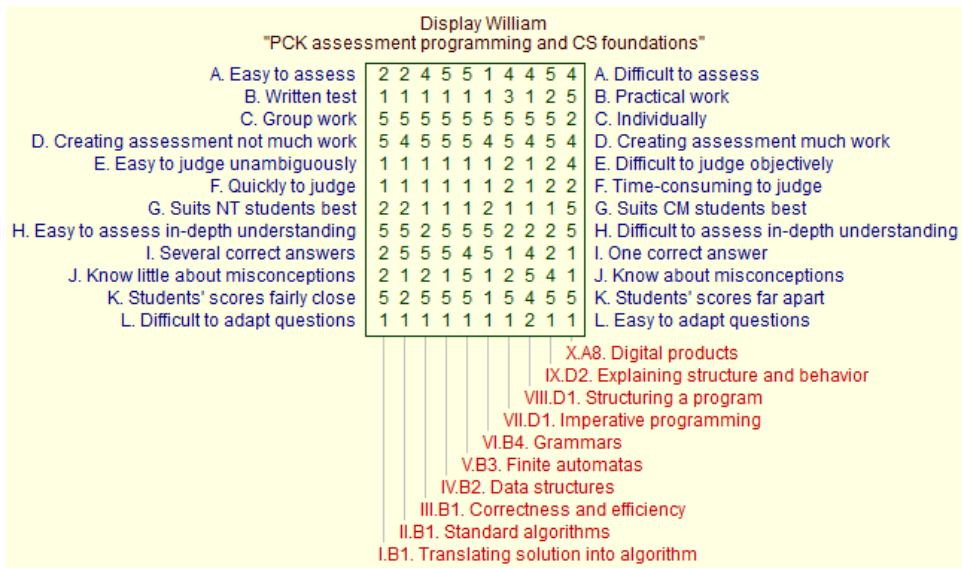


Figure E.6: Williams repertory grid

E.2 Focus cluster

As explained in section 4.3, each participant rated ten learning objectives of the fundamentally new (sub)domains of the reformed CS curriculum from a value of 1 to a value of 5 in terms of twelve bipolar constructs: score 1 means ‘Agree entirely with left side’ (i.e. agree with the construct on the left side of the grid), and score 5 means ‘Agree entirely with right side’ (i.e., agree with the construct on the right side of the grid). The feature Focus Cluster of the software WebGrid Plus puts similarly rated elements and the similarly rated constructs close together. Groups or clusters are indicated by the curved lines on the right side of the grid, which connect certain constructs, or elements. The percentages, ranging between 60% and 100%, on the upper right side of the grid, indicate how much is shared between certain constructs, or elements. The software program prints the scores in different shades ('1', white; '3', grey; and '5', dark) as explained previously in section 4.4. The created focus clusters of the participants (after the member checking) are shown below for reference.

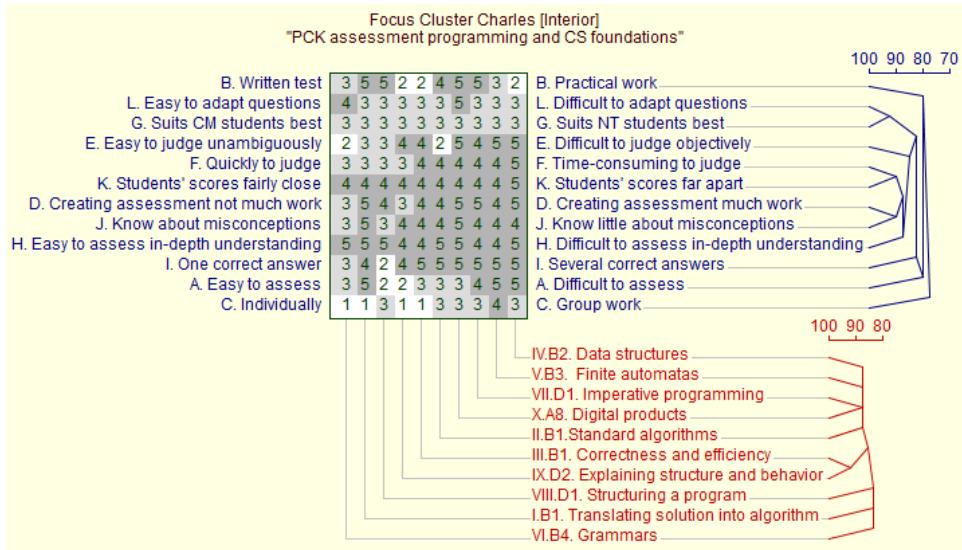


Figure E.7: Focus Cluster of Charles' repertory grid

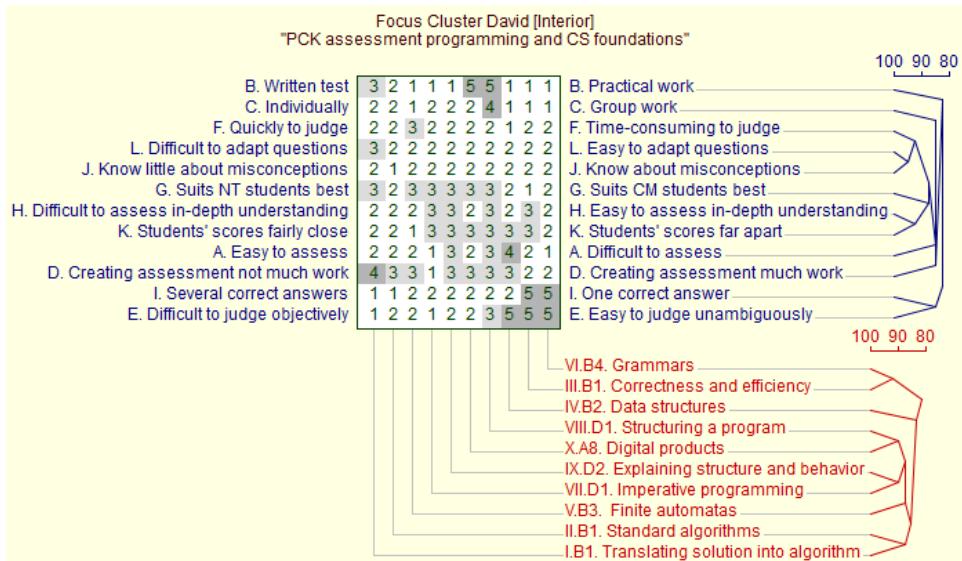


Figure E.8: Focus Cluster of Davids repertory grid

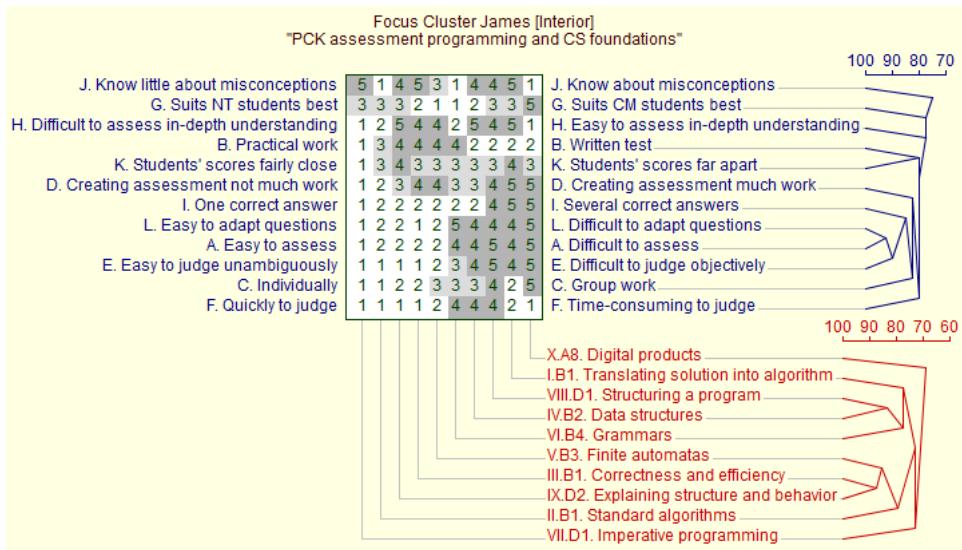


Figure E.9: Focus Cluster of James' repertory grid

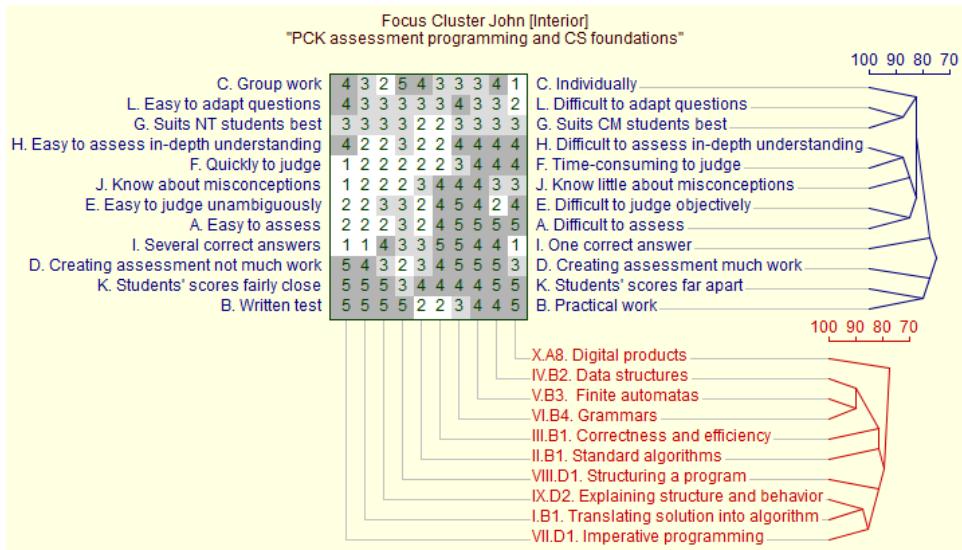


Figure E.10: Focus Cluster of Johns repertory grid

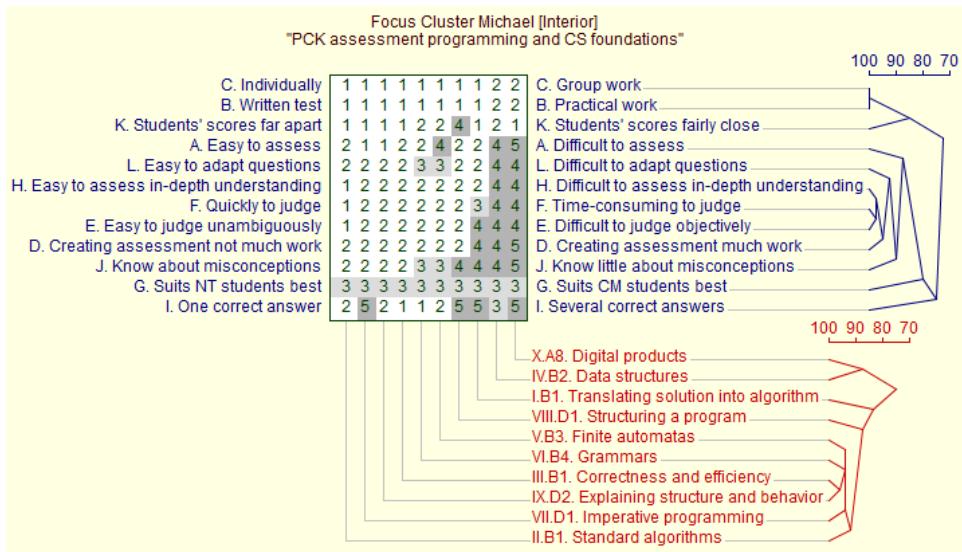


Figure E.11: Focus Cluster of Michaels repertory grid

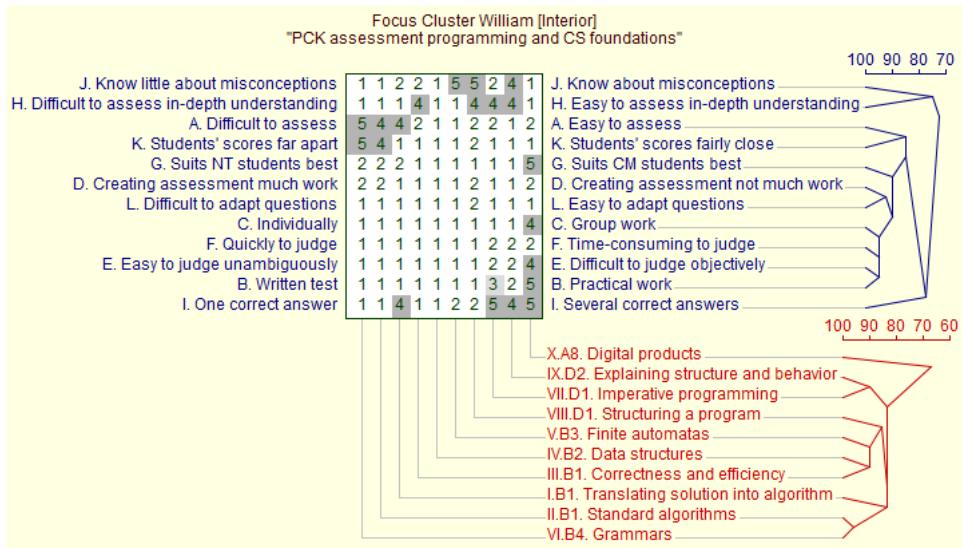


Figure E.12: Focus Cluster of Williams repertory grid

Appendix F

Online questionnaire

Professionele leergemeenschap - onderzoek Docentenkennis (PCK) over toetsing

31 %

Beantwoord de volgende vragen met het volgende leerdoel in gedachten:
De leerling kan een oplossingsidee voor een probleem uitwerken als een algoritme (dwz. in termen van stappen, keuzes en herhalingen). *

	Geheel eens met linkerzijde	Meer eens met linkerzijde	Neutraal	Meer eens met rechterzijde	Geheel eens met rechterzijde	
Ik vind het makkelijk om dit leerdoel te toetsen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Ik vind het lastig om dit leerdoel te toetsen
Dit leerdoel kun je het beste schriftelijk toetsen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Dit leerdoel kun je beter aan de hand van een praktisch werk toetsen
Dit leerdoel kan het beste groepsgewijs getoetst worden	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Dit leerdoel kun je het beste individueel toetsen
Het opstellen van toetsing hiervoor is weinig werk	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Het opstellen van toetsing kost veel tijd
Leerlingenwerk is makkelijk eenduidig te beoordelen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Leerlingenwerk is lastig objectief te beoordelen
Ik kan het werk van leerlingen snel beoordelen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Het beoordelen van leerlingenwerk is tijdrovend
Dit leerdoel past vooral bij een NT profiel	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Dit leerdoel past vooral bij een CM profiel
Ik kan voor dit leerdoel gemakkelijk inzicht toetsen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Ik vind het toetsen van inzicht voor dit leerdoel lastig
Doorgaans zijn er meerdere correcte antwoorden mogelijk	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Doorgaans is er maar één goed antwoord mogelijk
Ik weet weinig over de misconcepten en veelgemaakte fouten bij dit leerdoel	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Ik weet welke misconcepten en veelgemaakte fouten hierbij zijn
De scores van leerlingen liggen doorgaans redelijk dicht bij elkaar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	De scores van leerlingen liggen doorgaans ver uit elkaar

Appendix G

Semi-structured interviews

G.1 Interview questions used to elicit constructs

1. Do you find it important to assess students? Why?
2. What are difficulties/limitations are connected with the assessing of CS?
3. Which factors influence how you assess?
4. What are assessment methods?
5. What are your specific ways of ascertaining students understanding or confusion?
6. How do you assess your students?
7. Which dilemmas do you have when you make an assessment (task)?
8. What do you take into consideration when making an assessment?
9. What do you find hard when making an assessment?
10. What do you find hard when scoring an assessment?

G.2 Interview about perceptions and developing assessment task

Introduction about questions regarding results of repertory grid (MemberChecking)

Each result was described after which the participant was asked to elucidate this result.

Introduction about questions regarding how the assessment question was made

1. Hoe of waarmee bent u begonnen, toen u deze toetsopdracht ging maken?
2. Waar heeft u de input of inspiratie vandaan gehaald voor het maken van deze toetsopdracht?
 - (a) Heeft u gebruik gemaakt van materialen zoals papers of andere informatiebronnen? Zo ja, wat en waarvoor?
 - (b1) Zou u een voorbeeld kunnen geven uit de toetsopdracht waar u gebruik heeft gemaakt van uw eigen kennis of ervaringen om de toetsopdracht te maken?
 - (b2) Waar heeft u nog meer gebruik gemaakt van uw eigen kennis of ervaringen?
 - (c1) Zou u een voorbeeld kunnen geven uit de toetsopdracht waar u gebruik heeft gemaakt van inzichten uit de vorige PLC bijeenkomst om de toetsopdracht te maken?
 - (c2) Waar heeft u nog meer gebruik gemaakt van inzichten uit de vorige PLC bijeenkomst?
3. Heeft u het maken van deze toetsopdracht anders aangepakt dan vóór de afgelopen PLC bijeenkomst? Zo ja:
 - (a1) Wat is het belangrijkste verschil?
 - (a2) Wat heeft u nog meer nu anders gedaan?
 - (b) Per verschil:
 - i. Wat heeft ertoe geleid dat u dit anders bent gaan doen? Wat is het belangrijkste voordeel om het zo te doen?
 - ii. Verwacht u dit vanaf nu vaker op deze manier te gaan doen? (Waarom niet?)
4. Tijdens de eerste bijeenkomst is voorgedaan hoe een toetsopdracht gemaakt kan worden met behulp van een template. U lijkt dit template geheel/niet/... gevuld te hebben. Klopt dit?
 - (a) Deed u vóór de afgelopen PLC bijeenkomst al gebruik maken van een dergelijk template om toetsopdrachten mee te maken? Zo nee:
 - i. Hoe deed u toetsvragen maken vóór de afgelopen PLC bijeenkomst (bijvoorbeeld vragen van internet of uit toetsmateriaal halen of ...)?
 - (b1) Wat vindt u het belangrijkste voordeel van het gebruiken van dit template?

- (b2) Welke andere voordelen ziet u aan het gebruiken van dit template?
- (c1) Wat vindt u het belangrijkste nadeel van het gebruiken van dit template?
- (c2) Welke andere nadelen ziet u aan het gebruiken van dit template?
- (d) Verwacht u vanaf nu vaker een dergelijk template te gebruiken voor het maken van toetsopdrachten? Waarom (niet)?

Introduction about the content of the made assessment question and the opinion of the participant on that

1. Toetst uw toetsopdracht kennis, inzicht of beide?
 - (a) Kan u toelichten hoe uw toetsopdracht inzicht toetst?
 - (b) Hoe makkelijk of moeilijk vond of vindt u het om inzicht te toetsen in uw toetsopdracht?
2. U noemt deze miconcepten of veelgemaakte fouten bij uw toetsopdracht.
 - (a) Hoe makkelijk of moeilijk vond of vindt u het om misconcepten en veelgemaakte fouten te bedenken bij deze toetsopdracht?
 - (b) Bent u tevreden over de manier waarop u de misconcepten/veelgemaakte fouten toetst?
3. Zal het beoordelen van leerlingenwerk in uw ogen eerder subjectief of objectief zijn? Waarom?
4. Hoeveel tijd heeft het maken van deze toetsopdracht u gekost? Relatief veel of weinig vergeleken met hoeveel tijd het u kostte om toetsopdrachten voor dit leerdoel vóór de afgelopen PLC bijeenkomst?
 - (a) Waardoor komt dat denkt u?
 - (b) En zal dat de volgende keer ook weer zo veel tijd kosten of gaat het dan sneller, denkt u?
5. Hoe makkelijk of moeilijk vond u het om variaties op deze toetsopdracht te maken, waarbij het leerdoel en de moeilijkheidsgraad hetzelfde blijven? (Waardoor komt dat denkt u?)

If time left

6. Hoeveel tijd verwacht u dat het kost om het leerlingenwerk van deze toetsopdracht te beoordelen? Relatief veel of weinig vergeleken met het nakijken van toetsopdrachten voor dit leerdoel van vóór de afgelopen PLC bijeenkomst?

- (a) En zal dat de volgende keer ook weer zo veel tijd kosten of gaat het dan sneller, denkt u?
7. Verwacht u dat bij deze toetsopdracht de meeste leerlingen een 7 halen, of dat er ook vier en negens tussen zullen zitten?
 8. Verwacht u dat deze toetsopdracht beter past bij leerlingen met een NT profiel of bij leerlingen met een CM profiel?

G.3 Neat transcription of MemberChecking

In this appendix, the transcriptions of the semi-structured interviews with each of the participants is shown. These transcriptions are in Dutch as both the researcher (R) and each interviewee (I, participant) are Dutch. The way of asking questions for the interviews differs slightly between the interviews, because they were improved over time (to get even more interesting results).

G.3.1 Charles

[00:36]

R: U heeft in de online vragenlijst, tijdens de eerste bijeenkomst dus, aangegeven dat u het voor elk leerdoel moeilijk vindt om inzicht te toetsen. Kan u dit toelichten?

I: Moeilijk vindt om?

R: Inzicht te toetsen.

I: Ja, inzicht toetsen. Het makkelijkste is natuurlijk om te vragen ‘Is het A of B?’, dat is altijd makkelijk, maar het toetsen van inzicht, dan zit je natuurlijk op een ander – hoe noem je dat – niveau van – hoe heet dat ook alweer – van meta-cognitie, en ik vind dat altijd moeilijk om daarbij de goede opdracht te vragen. Bovendien is het ook zo dat wat voor de ene leerling een vorm van inzicht is, voor een andere vaak een vorm van toepassing of van reproductie is, omdat ze op een andere manier natuurlijk les gehad hebben of zichzelf bekwaamd hebben in die leerstof, dus ja, wanneer is het inzicht en wanneer is het geen inzicht, dat hangt ook een beetje af van wat de persoon zelf geleerd heeft of onderwezen heeft gekregen. Dus daarom vind ik het altijd een beetje moeilijk om te zeggen van ‘dit is een inzichtsvraag’.

R: Oké, dus het gaat u – zeg maar – meer om van wanneer iets inzicht is dan om, stel je zou precies weten wat iemand als voorkennis gehad, vindt u het dan nog steeds moeilijk om inzicht te toetsen denkt u?

I: Wanneer bedoel je?

R: Wanneer u precies weet wat de voorkennis is van iemand, dus wanneer iedereen exact dezelfde voorkennis heeft, zeg maar.

I: Ja, dan zou het makkelijker zijn, maar dat weet je niet.

R: Oké, dat snap ik, maar even om duidelijk te krijgen

I: Kijk, als ze allemaal hetzelfde programma gedaan zouden hebben; de leerlingen die ik in de klas heb, daarvan weet ik wat hen aangeboden is en wat ze zouden moeten weten, dus dan is het in ieder geval makkelijker om van deze vraag is een vorm van inzicht, maar als je, zoals we natuurlijk in die PLG bezig zijn om vragen op te stellen als voorbeeld voor anderen, ja ik weet niet wat ze op een school in een stad verderop allemaal gedaan hebben, dus op die manier vind ik dat dan moeilijk.

R: Oké, dan is dat helder.

[03:26]

R: U heeft in de vragenlijst voor ieder leerdoel ‘neutraal’ ingevuld bij de vraag of het leerdoel beter past bij leerlingen met een NT-profiel of bij leerlingen met een CM-profiel.

I: Ja.

R: Kan u dit toelichten of begreep u die vraag niet?

I: Nou, ik kan dat wel toelichten. Ook hier hangt het weer erg van de persoon af. NT-leerlingen zijn over het algemeen beter in dingen wat, nou ja bèta-dingen, dus algoritmen en dergelijke, dat is wat eenvoudiger voor hen om dat te doorgronden, maar ik heb heel gemêleerde klassen met NT, EM en CM erin, en die CM-leerlingen die kunnen, de een kan heel goed overweg met een begrip als ‘algoritme’ en een ander niet. Dus ja, ik vind dat dan moeilijk om te zeggen van nou dat is beter of niet, die kunnen dat beter aan dan anderen.

R: Omdat het dus zoveel verschilt per leerling?

I: Ja. We zijn gelukkig geen van allen hetzelfde, en zelfs in het CM-profiel zijn er leerlingen die ook heel goed de programmeerstappen in informatica kunnen volgen.

R: Oké, dan is ook dat helder.

[04:59]

R: Een van de percepties in de vragenlijst was ‘Ik vind het makkelijk of lastig om een vraag aan te passen voor een nieuwe toets, waarbij het leerdoel en de moeilijkheidsgraad hetzelfde blijven’. Ook hierbij heeft u voor de meeste leerdoelen ‘neutraal’ ingevuld. Kan u dit toelichten?

I: Ja, misschien was ik wel erg makkelijk op dat moment, dat kan ook nog hè, maar het aanpassen van een vraag... Het is geen wiskunde, het is geen rekenen, ik mag natuurlijk niet zeggen wiskunde, maar bij rekenen ga je de getallen veranderen en dan heb je een nieuwe opdracht; bij wiskunde ga je x en y andere waarden geven en dan heb je een andere opdracht. Bij informatica is dat niet zo; dat is wat moeilijker om op hetzelfde gebied een vraag te maken of aan te passen. Je moet dan vaak al weer – hoe zal ik dat zeggen – een nieuwe situatie hebben waarin je hetzelfde kunt vragen. Begrijp je een beetje wat ik bedoel?

R: Ja.

I: En ik was bijvoorbeeld van de week bezig met lineair/binair zoeken; dat

kan je op een heleboel manieren doen, maar uiteindelijk is natuurlijk het principe hetzelfde, maar hoe kan ik nou makkelijk een nieuwe vraag maken over datzelfde principe, dat is best wel een ding, daar moet je toch even goed over nadenken.

R: Ja. Maar wat u nu aangeeft is dat u het eigenlijk best moeilijk vindt?

I: Ja, eigenlijk wel ja.

R: Is dat dan bij alle leerdoelen zo, of zijn er ook leerdoelen waarbij u denkt van nou bij die is het wel wat makkelijker?

I: Ja, er zijn wel een aantal dingen waar het, nou laat ik zeggen waar we nu mee bezig zijn, dat [domein] Grondslagen en ook domein D, dat vind ik heel moeilijke dingen. Die algemene zaken, zoals verschillende beeld-bestanden of compressie of hoe werkt een computer, ja, dat zijn over het algemeen best dingen die je, ja daar kan je heel snel iets nieuws bij maken is mijn idee, maar er zijn een paar domeinen, en vooral dit B – daarom ben ik ook mee gaan doen met die PLG, om daar wat beter in te worden, om daarin wat meer vaardigheid te krijgen qua het maken van vragen.

[08:00]

R: Ja, oké. En zijn er dan ook nog leerdoelen die u extreem moeilijk vindt om vragen aan te passen, zeg maar specifieke leerdoelen van - je hebt bijvoorbeeld algoritmen, datastructuren, grammatica's - zijn er dan bepaalde delen die u nog moeilijker vindt dan anderen?

I: Ja, grammatica's, dat vind ik moeilijk.

R: Kunt u ook uitleggen waarom?

I: Misschien omdat ik niet een bèta-achtergrond heb, [dat] zou best kunnen.

R: Oké, want u denkt dat u er wel voldoende voorkennis voor heeft?

I: Jawel

R: Want dat is zeg maar wat je vaak hoort, dat net als bijvoorbeeld grammatica's en automaten bijvoorbeeld minder achtergrond hebben dan bij andere delen, dus daarom vroeg ik het even.

I: Ja, dat heeft er wel mee te maken, zeker.

R: Oké.

[09:00]

R: Dan de volgende vraag: een van de leerdoelen in de vragenlijst was 'De student kan een programma zo structureren dat anderen het eenvoudig kunnen begrijpen en evalueren'. Dit is het enige leerdoel waarbij u heeft aangegeven dat er in het algemeen maar één goed antwoord mogelijk is in plaats van meerdere goede antwoorden. Kan u dit toelichten?

I: Even denken, hoor. Ja, ik denk dat, als het goed is, heb je in je programma er commentaar in gezet. En aan de hand van de commentaren zou een ander moeten kunnen zien hoe/of het een en ander in elkaar zit en hoe/of het werkt.

R: En waarom denkt u dan dat er dan maar één goed antwoord mogelijk is?

I: Nou, daar zit ik nu eigenlijk een beetje over te twijfelen. Het hangt er een beetje vanaf, het hangt natuurlijk ook van de taal af, natuurlijk.

Sommige talen zijn heel erg strikt, en als je daar de volgorde niet aanhoudt, dan gebeurt er dus niks. En andere talen zijn wat vrijer en dan kun je misschien, nou ja met functies of voor... nog wel wat schuiven, en dan hoeft het niet allemaal precies hetzelfde te zijn. Dus misschien dat ik toen wat snel ingevuld heb.

R: Oké, want u geeft dus eigenlijk aan dat het ook echt aan de programmeertaal ligt hoeveel antwoorden er mogelijk zijn?

I: Ja

R: Oké

[11:26]

R: Een ander leerdoel in de vragenlijst was 'De student kan het gedrag van een fysiek systeem beschrijven als een eindige automaat'. En dit is het enige leerdoel waarbij u heeft aangegeven dat het beter met groepswerk getoetst kan worden dan individueel. Kan u dit toelichten?

I: Nou nee, ik weet niet waarom ik op dat moment dat antwoord gegeven heb.

R: Oké, want doet u over het algemeen alles individueel toetsen, of denkt u dat alles over het algemeen beter individueel getoetst kan worden dan met groepswerk?

I: Nee. Ik heb de indruk dat echt kennis-toetsing, dat je dat het beste individueel kunt doen, maar een groot gedeelte – dat is bij mij in de les minstens de helft – is het uitvoeren van projecten, en ik vind eigenlijk dat je dat niet individueel mag doen, want dat is ook niet zoals het werkt in de echte wereld, zullen we maar zeggen. Systemen worden ondertussen zo complex, dat kun je niet meer in je eentje doen, dus je moet echt wel met meerdere mensen aan dergelijke projecten werken, en ik vind ook dat dat zo hoort.

R: Oké. Zijn er dan ook specifieke onderdelen, zoals ik ook net noemde van programma's maken of datastructuren, grammatica's, automaten, zijn er dan bepaalde onderdelen waarvan u denkt dat kan echt beter met groepswerk, of in projecten, getoetst worden? Of is dat denkt u bij allemaal zo, wanneer u inzicht toest?

I: Vooral de zaken die te maken hebben met gebruikers, [zoals] het maken van een systeem, het maken van een interface, die dingen moet je volgens mij echt met groepswerk doen, en inzicht in programmeren en automaten misschien, ja, als je dat als kennis wilt zien, dan kan je dat individueel toetsen. Ik denk dat er meer zaken zijn die je als groepswerk zou moeten doen in informatica, ik noem maar wat: UX design en dat soort dingen, en dan een app erover bouwen of een webpagina of wat dan ook, ja dat moet je dan wel samen doen. Maar als ik het heb over een webpagina bouwen, het is natuurlijk wel zo dat ieder individueel groepslid moet weten hoe HTML en CSS in elkaar zit.

R: Ja

I: Is dat ongeveer een antwoord wat je...

R: Dat lijkt me helder inderdaad; dat lijkt me ook wel logischer.

I: Ja

[14:48]

[End of the MemberChecking part of the interview with Charles]

G.3.2 David

[00:17]

R: Er is maar één leerdoel waarbij u heeft aangegeven dat deze beter getoetst kan worden met groepswerk dan individueel. Dat leerdoel is ‘De student kan een programma zo structureren dat anderen het eenvoudig kunnen begrijpen en evalueren’. Kan u dit toelichten?

I: Even kijken, dat was een leerdoel uit domein D, of niet? Of uit B? Even denken bij welk subdomein hij hoort.

R: Dat is inderdaad uit domein D, D1.

I: Ja. Kan je hem dan nog een keer verwoorden? Want ik heb hier alleen de lijst van domein B uitgebreid voor me.

R: Als het goed is, stond die daar ook tussen of niet? Ik heb een bestandje gestuurd met alle leerdoelen en percepties; daar staat hij in.

I: Oh help, ik heb dat bestandje met leerdoelen uit de gedeelde map gehaald.
[short conversation about where the correct file can be found]

[03:12]

R: Het ging over het leerdoel ‘De student kan een programma zo structureren dat anderen het eenvoudig kunnen begrijpen en evalueren’.

I: Ja, ik zie hem staan. En ik had daarop aangegeven dat dat het beste in groepsverband is.

R: Ja, en dat is het enige leerdoel waarbij u dat heeft aangegeven. Kan u dit toelichten?

I: Ja, tuurlijk. Als ik denk aan mijn eigen lessen, dan zou dit bijvoorbeeld een opdracht kunnen zijn van dat ik een opdracht geef waarin ze, nee. Ik weet waarom ik dat gedaan heb; ik heb ooit een keer een test gedaan met de klas, toen hadden ze een opdracht gekregen om in C# een dartbord te maken; een programma om de score van een darterstijd bij te houden, dus met drie tekstvakken voor de pijlen en dan een keertje op opslaan en dan ging hij gewoon veilig naar de ene helft, nou op die manier. Toen had ik iedereen de opdracht gegeven dat ze eerst de layout moesten maken, dus alleen de layout en verder niets, dus grafische dingen. Toen hebben ze dat ingeleverd. Ik heb toen daarna alles geshuffled, en toen naar iedereen een willekeurig document opgestuurd, en toen gezegd van nou ga dat nu maar afmaken - dus dat je dan dus niet met je eigen beginprogramma begonnen bent, dus dingen als naamgeving van variabele elementen die had je misschien zelf netjes gedaan, maar iemand anders had daar een hele chaos van gemaakt -, en op die manier heb ik ze geleerd dat het dus blijkbaar heel handig is om dingen te structureren, je programma zo te structureren dat andere mensen er ook makkelijk mee aan de slag kunnen. Ja, dat was eigenlijk een soort groepsopdracht die we hebben gedaan, waarbij het denk ik voor iedereen helder was dat je goed moet nadenken over naamgeving van variabele elementen. En met dat in mijn achterhoofd heb ik die vraag dus beantwoord, nou ja, dat gaat heel makkelijk in een groep, omdat je dan

ook echt perfect ziet van als je dit programma niet goed structureert dan is het gelijk helder waarom dat toch wel handig is om te doen. Ik kan wel een heel blij verhaal op hangen voor de klas tegen de leerlingen van ‘ja, je moet je programma structureren en dit en dat’, [maar] als je ervaart als het niet zo is, dan leer je er toch meer van. Dat was toen heel mooi gedaan, een HAVO 4 klas was dat. *[David has to laugh]* Dat zorgde wel voor frustratie als mensen gewoon een heel brak ding kregen van iemand anders.

[05:26]

R: En kan je dan ook toelichten waarom je bij de andere leerdoelen dan hebt aangegeven dat je ze beter individueel kan toetsen?

I: Die andere zijn vooral ook vaardigheden die je bijvoorbeeld ook op papier zou kunnen toetsen. Bijvoorbeeld als je zegt- ik pak een willekeurige - ‘De leerling kan bij een gegeven grammatica en invoer bepalen of de invoer correct is’, dat hoef je niet in een groep te doen, dat kan je gewoon met pen en papier doen. Maar juist dat ene leerdoel was echt iets, dat moet je gewoon in de praktijk meemaken; dat moet je met anderen doen; dat lukt je niet in je eentje.

[06:00]

R: Dus je vindt dan ook dat er een samenhang is tussen geschreven toetsen en individueel werk, als ik het zo hoor?

I: Ja. Een toets maak je sowieso individueel natuurlijk, dus daar zit automatisch samenhang tussen. Het enige wat je dan beter kan kijken van kun je iets individueel doen of in groepswerk, en niet zozeer of dat dan een geschreven toets is of dat je als leerling zelf een stuk praktisch werk oplevert.

[06:38]

R: Oké, dan gaan we verder met de volgende vraag. U heeft in de vragenlijst voor de meeste leerdoelen neutraal ingevuld bij de vraag of het leerdoel beter past bij leerlingen met een NT profiel of bij leerlingen met een CM profiel. Kan u dit toelichten of begreep u de vraag misschien niet?

I: Oké, je moet dus opletten wat je hebt ingevuld blijkbaar. Ik begreep – denk ik – de vraag wel, dat zal niet de onderliggende reden geweest zijn dat ik daar voor neutraal heb gekozen. Ik denk, de leerdoelen die zijn voor iedere leerling in principe gelijk, alleen de manier waarop je met een leerdoel omgaat, daarin kun je wel differentiëren tussen profielen. In principe moet elke leerling een leerdoel behalen, maar de weg daarnaartoe die is anders denk ik. Je probeert onderscheid te maken tussen N-leerlingen en M-leerlingen. Ik denk dat ik daarom vooral voor neutraal heb gekozen.

R: Dus u denkt dat het niet zozeer een leerdoel is die het best bij een profiel past, maar meer hoe je het toetst?

I: Ja, en het ging natuurlijk hier vooral om het kernprogramma en het kernprogramma is in principe voor elke leerling, dus je zou dan beter kunnen zeggen van de keuzethema’s daarin kun je beter differentiëren op profiel dan op het kernprogramma, want ja, het kernprogramma is nou eenmaal

voor iedereen verplicht. Dan kan het zijn dat een leerling, bijvoorbeeld een N&T-leerling zal programmeren beter liggen dan een C&M-leerling, ja daar zijn ook allemaal overeenkomsten, maar dat neemt niet weg dat ook die C&M leerling ook moet kunnen programmeren.

R: Ja oké, dan is dat helder.

[08:38]

R: U heeft in de online vragenlijst in het algemeen op maar weinig plekken ingevuld dat u het geheel eens bent met een van beide kanten. U heeft dit bijvoorbeeld wél gedaan bij de vraag of een leerdoel beter getoetst kan worden met een geschreven toets of praktisch werk, maar bij veel andere vragen liggen uw antwoorden vaak in het midden. Kan u dit toelichten?

I: Ja, dat is een mooie, ik heb het niet voor je uitgekozen. Ik denk dat dat voor een deel wel komt omdat je, ja die uitersten zijn zo zwart-wit dan zeg maar, dus dat je dan daarvoor kiest dat komt dan meer op mij over dat het enige manier is waarop je dat kan doen, terwijl als je dan een balletje er vanaf gaat zitten, betekent dus dat je wel die richting op neigt, maar dat er nog wel variaties op mogelijk zijn. En bij welk leerdoel had ik dat dan wel specifiek aangegeven, bij welk punt?

R: Waarop u wel hebt ingevuld dat u het geheel eens bent met een van beide kanten?

I: Ja.

R: Of een leerdoel beter getoetst kan worden met een geschreven toets of praktisch werk.

I: En was dat dan specifiek bij een speciaal leerdoel of niet?

R: Even kijken...

I: In principe kan je, misschien heeft het te maken met bijvoorbeeld Domein D van programmeren, dat kan je in mijn eigen beter met praktisch werk toetsen dan met een geschreven toets, terwijl bijvoorbeeld een onderdeel als algoritmiek dat zou je beter als geschreven toets kunnen doen, maar dat neemt niet weg dat het ook praktisch kan, dus daarom denk ik van dan ga ik net iets van het gemiddelde af zitten. Ik vermoed dat dat mijn keuze daarin is geweest van ja, programmeerdingen, domein D, dat moet je eigenlijk gewoon praktisch doen. Ja, het kan ook met een geschreven toets, maar nee dat voelt voor mij niet goed. Ik zou dat gewoon praktisch doen, terwijl eigenlijk heel domein B daar zou mijn voorkeur liggen voor een geschreven toets, maar dat neemt dan niet weg dat ik er ook praktische opdrachten voor zou kunnen bedenken om het af te toetsen.

[10:48]

R: Ja, want ik zat dus te kijken wat de precieze getallen [beoordelingen] waren, maar u heeft zeg maar bij dat groepswerk of geschreven test, heeft u alleen bij 'De student kan de oplossing van een probleem vertalen naar een algoritme', dat is de enige waarbij u neutraal heeft aangegeven, en op een aantal zaten alle anderen op geheel eens met de ene kant of geheel eens met de

andere kant.

I: Oké, nou die ene van neutraal dat is - denk ik dan - omdat we daar een heel mooie praktijkopdracht voor hebben, maar die je ook theoretisch kan doen. Dat verklaart dan waarschijnlijk mijn neutraal. En de rest, die heb ik dan blijkbaar op die andere gezet, omdat ik daar gewoon materiaal voor heb en het op die manier op dit moment die leerdoelen af toets. Dat moet het geweest zijn, denk ik.

R: Oké, dan is dat ook duidelijk.

[11:46]

I: [moet lachen] Ik moet denken dan heb je in februari dat we al die dingen invullen en dat moet je er nu nog heel hard over nadenken, dat is wel mooi.

R: Ja, dat snap ik, maar soms zijn het dan gewoon dingen van 'oh heb ik dat ingevuld, dat is eigenlijk wel raar, want ik vind dit'. Daarom vind ik het belangrijk om dan toch samen terug te vragen van 'is dit ook wel logisch?'

I: Nee, ik vind het prima, maar ik moet dan wel even wat graven na afgelopen tijd.

R: Ja, ik snap het.

[12:12]

R: U heeft voor vrijwel alle leerdoelen aangegeven dat u het enerzijds makkelijk vindt om het leerdoel te toetsen, maar dat u het anderzijds bijvoorbeeld wel moeilijk vindt om inzicht ervoor te toetsen, dat u vrij weinig weet over de bijbehorende misconcepten en veelgemaakte fouten, en dat u het vrij moeilijk vindt om vragen aan te passen voor nieuwe toetsen waarbij de moeilijkheidsgraad hetzelfde blijft. Kan u dan toch toelichten waarom u het dan toch makkelijk vindt om het leerdoel te toetsen?

I: Op zich, een vraag bedenken voor een leerdoel is niet heel ingewikkeld. Een reproductievraag is natuurlijk altijd makkelijk om te maken en ik heb vooral aangegeven inzichtsvragen zijn lastig omdat ja, dat is natuurlijk gewoon de moeilijkste categorie vragen. Reproductievragen, die kan je zo uit je mouw schudden; je pakt gewoon het boek erbij, je haalt een paar woorden weg uit een alinea, en je hebt al een vraag zeg maar. Dat is niet heel ingewikkeld. Toepassingsvragen, daar kom je over het algemeen ook nog wel mee weg. Maar echt een goede inzichtsvraag, die zijn gewoon lastig om te maken, en dan lastig in de zin van 'wanneer is een vraag nou echt een inzichtsvraag' of 'wanneer is een vraag nou eigenlijk gewoon een lastige toepassingsvraag'. Ik denk dat daar ook wel een grijs gebied tussen zit, tussen wanneer een vraag nou echt inzicht vereist, en misschien is het ook persoonlijk voor een leerling. Een heel goede leerling uit je klas, die noemt een gemiddelde inzichtsvraag misschien wel een toepassingsvraag omdat ze al dat inzicht hebben, terwijl andere leerlingen die juist niet hebben. Ja, ik denk dat dat de achtergrond is geweest; van ja, op zich vragen maken voor een leerdoel, dat is te doen; dat is je werk als docent. Daar kan je lang of kort over praten, maar dat moet je gewoon doen. Dat lukt tot op zekere hoogte wel. Alleen de categorie

inzichtsvragen, die vind ik persoonlijk gewoon lastig om te maken.

[14:06]

R: Ja, want u heeft ook voor geen enkel leerdoel aangegeven dat u het makkelijk vindt om dat leerdoel te toetsen, maar kan u dan ook verklaren wat de reden daarvoor is, dat u dat zo moeilijk vindt?

I: Ja, dat vind ik een lastige vraag. Voor inzichtsvragen moet je ook best creatief zijn, voor het bedenken van contexten, en op een of andere manier ben ik dat gewoon niet. Ik vind het gewoon lastig om nieuwe dingen te bedenken en ook als ik vragen ga bedenken, dan spiek ik toch meestal bij wat heb ik in het verleden gemaakt en daar maak je dan een variatie op, dus echt iets nieuws maken, dat is gewoon mijn zwakke plek. Dat is niet alleen bij vragen maken, maar bij allerlei dingen in het leven, denk ik, dus ja ik vind nieuwe dingen zelf bedenken gewoon lastig. Dat heeft hier ook wel mee te maken.

[15:05]

R: Ja, want u heeft bijvoorbeeld ook aangegeven dat u het voor de meeste leerdoelen vrij moeilijk vindt om vragen aan te passen voor nieuwe toetsen, maar u zegt net eigenlijk dat u dat best wel vaak doet. Kan u dat dan ook verklaren?

I: Nou ja, aanpassen, nee aanpassen is iets anders dan spieken in het verleden wat andere mensen doen en aan de hand daarvan zelf iets nieuws bedenken; dat is niet zozeer aanpassen, maar meer het idee overnemen en daar dan mee aan de slag gaan en zelf uitwerken.

R: Dus u gebruikt het dan meer als inspiratie?

I: Ja, voor een groot deel is het inspiratie. En vragen aanpassen waarbij de moeilijkheidsgraad hetzelfde blijft; meestal als ik aan een vraag ga sleutelen, dan wordt hij of veel makkelijker of veel moeilijker in mijn ervaring, en niet dat hij dan gelijk blijft. Ja, je kan een paar woorden vervangen, maar is het dan een andere vraag, daar kan je over discussiëren. Maar de moeilijkheidsgraad moet gelijk zijn gebleven, dat is meer iets wanneer je twee versies wil maken voor een klas, dat je dan een paar woorden of een paar getallen anders maakt, dat vind ik niet echt aanpassen, nee. Als je echt gaat sleutelen aan een vraag, dan, dus bij een meerkeuzevraag als je er een extra optie bij zet, dan is dat meestal wel of het goede antwoord of een optie die niet makkelijk is. Dat vind ik inderdaad wel lastig, om het niveau te bewaken bij het aanpassen.

R: Oké.

[16:24]

R: U heeft voor alle leerdoelen in de vragenlijst aangegeven dat u vrij weinig tot weinig weet over de bijbehorende misconcepten en veelgemaakte fouten. Kan u dat toelichten?

I: Ja, dat komt voor een groot deel doordat het een nieuw examenprogramma is. Ik sta al sinds 2010 voor de klas en toen hadden we een ander

examenprogramma. Daarbij weet je na een aantal jaar wel wat leerlingen vinden of niet vinden van leerdoelen. Terwijl dit; we hebben het eerste jaar (2018) tests gebruikt in de klas, in een paar klassen; Dit schooljaar zijn we echt volledig aan de slag gegaan met het nieuwe examenprogramma. Ja, dat moet gewoon groeien, die misconcepten; je moet gewoon een aantal toetsen gemaakt hebben en daar dan ook echt op de vraag een gemaakt hebben, dat je dan achteraf ziet van ja als ik dit denk, dan denkt de leerling iets anders; zo bouwt jouw eigen lijst met misconcepten eigenlijk gestaag op. Ik denk dat dat gewoon een kwestie is van tijd. Geef het een paar schooljaren, en dan weet je van de meeste leerdoelen wel wat de leerlingen er wel of niet van vinden.

R: Ja. Dat klinkt wel alsof je er van tevoren niet echt over nadenkt wat mogelijke misconcepten en veelgemaakte fouten bij een vraag zijn. Klopt dat?

[17:31]

I: Ja, dat zou kunnen. Meestal als je een toets maakt dan is dat, toetsen maken en nakijken is nooit echt heel leuk werk; daarom heb ik me ook zeker aangemeld hiervoor, omdat ik denk dat ik daar dan iets meer vaardigheid in krijg. Soms is het maken van een toets ook gewoon van dan heb je nog een halfuurtje over en dan denk je ik raap snel nog even een toets bij elkaar en dan zien we wel wat die doet. Vroeger had je dan op zich de ervaring dat je wist wat je dan niet moest vragen en nu ben je daar gewoon weer in aan het zoeken, en dat komt dus inderdaad van soms dan neem je zelf gewoon te weinig tijd om na te denken over een vraag, over dus dat er bepaalde misconcepten kunnen zijn en dat zie je dan in de praktijk; dan kan je in de beoordelingen dan wel recht trekken, als je ziet dat een hele klas een bepaalde vraag fout maakt.

[18:21]

I: Per vraag hou ik bij wat de score is in een heel groot Excel bestand. Leerlingen hebben altijd zoiets van maar dat kost toch twee keer zoveel nakijktijd; het kost dan ook wel even, maar ik heb dan wel mooi inzicht in welke vraag ze goed of fout maken en of ik volgend jaar zo'n vraag wel of niet zo kan hergebruiken. Dus mijn lijst met misconcepten, die bouw ik gestaag op als ik al die Excel-bestanden bekijk. Het is wel zo dat als ik nu, als ik het [Excel-bestand] dit jaar goed bij houd, dat ik dan als ik volgend jaar een toets zou moeten maken, dan kijk ik dan terug naar wat kwam er afgelopen jaar uit; En vragen die dan gewoon gemiddeld gemaakt worden, neem ik dan gewoon een-op-een over. Vragen die heel slecht gemaakt worden, zal ik dan moeten aanpassen, en vragen die heel goed gemaakt worden misschien ook wel.

R: Ja.

[19:03]

I: Nou, een lang antwoord. Ik denk daarvan de samenvatting is dat ik persoonlijk soms te weinig tijd neem om een toets te maken en dat door het

nieuwe examenprogramma, je lijst met misconcepties vanzelf gaat groeien door ervaring.

R: Oké.

[19:16]

[End of the MemberChecking part of the interview with David]

G.3.3 James

[00:34]

R: Ik vroeg me af, we hadden een aantal subdomeinen waar we geïnteresseerd in zijn en dat zijn voornamelijk algoritmiek, uhm datastructuren, automaten, grammatica, programmeren, en ik heb daar ook nog een ontwerpding. Als je naar die dingen kijkt, waar voel je je het meest vertrouwd bij om te toetsen?

I: Mag ik dat ook de andere kant op doen, waar ik me niet vertrouwd bij voel?

R: Ja

I: Design en implement, daar voel ik me heel ongemakkelijk bij. Grammatica's voel ik me ongemakkelijk bij. Automaten kom ik wel uit. Datastructuren dat gaat al weer wat beter, en voor de rest gaat het allemaal prima.

[01:28]

R: Waarom vind je deze dan zo lastig?

I: Nou, ik heb zelf helemaal niets met ontwerp; daar mis ik een theoretische basis van, dus daar heb ik ook uit mezelf weinig richtlijnen of criteria die ik daarbij zou kunnen hanteren. Ik mis daar een kader. Ik mis daar gewoon zelf een kader bij.

R: Hopelijk komen we daar met onze groep ook wel verder uit.

[01:55]

R: Als je nou kijkt naar programmeren, want je zegt zelf dat is wel iets waar je zelf wel zoiets van hebt van daarvan weet je het wel,

I: Oh sorry nog eentje aanvullen, dus dat is voor wat betreft ontwerp, wat versta je onder ontwerp?

R: Ja, vaak is een ontwerp iets van teken een stroomdiagram voor dat je het gaat bouwen. Tenmiste, zo zou ik het zien, want er staat hier "kan voor een gegeven situatie een digitaal product ontwerpen, implementeren en evalueren"

I: Ja oké, als het gaat om het ontwerpen van een oplossing in termen van je krijgt een probleem voorgelegd, daar moet je een oplossing voor bedenken, dat lukt op zich wel, want dat zijn dingen die correct zijn. Als je praat over het ontwerp van je moet iets vormgeven, bijvoorbeeld een website vormgeven, die vormgeving, dat ontwerp, daar voel ik me heel ongemakkelijk bij. Volgens mij is dat niet deze definitie.

R: Nee. Volgens mij is dit ook meer van kan jij een oplossing, een programma schrijven die wat doet.

I: Want, integendeel, dat is juist wat in mijn aanbod juist heel sterk vertegenwoordigd is. Daar draait het veel meer eigenlijk om bij mij.

[03:04]

I: Dan voel ik me het minst gemakkelijk bij grammatica's en dat heeft ook, dat heeft absoluut te maken met de magere theoretische basis die ik zelf

heb en het gebrek aan overtuiging dat ik leerlingen daarmee zou moeten kunnen bedienen.

R: Dat ze er echt wat aan hebben?

I: Ja precies

R: Dan zou het waarschijnlijk ook wel een beetje wat minder gemotiveerd worden en moeilijker om inzichtsvragen voor te bedenken?

I: Nou kijk, ik heb me er ook nog nooit in verdiept.

R: Heb je daar al toetsen in afgenoem?

I: Nee

[03:42]

R: Je had het ook over dat je verschillende individuele vaardighedspractica, vaardigheidstoetsjes gaf. Je gaf daar ook een voorbeeld van ‘dan gaan de leerlingen eerst anderhalf uur aan een ontwerp werken’.

I: Ja, als je nu het [deel] over programma-ontwerp en programmeren afrond, dan bestaat dat uit drie stappen. Eerst het lezen van een oplossing, dus een schematische oplossing als bijvoorbeeld een stroomschema of flowchart of een programmastructuurdiagram, die krijgen ze en dan moeten ze zeggen wat die doet door aan te geven wat er getoond wordt; wat waardes zijn, wat de resultaatwaarden zijn die getoond worden en daarmee laten ze zien dat zo’n oplossing kunnen lezen als eerste stap. Als ze dat hebben gedaan, dan krijgen ze een opdracht om iets te ontwerpen. Ze krijgen een probleemomschrijving. Daarvoor moeten ze een schematische oplossing maken, hetzij een flowchart, hetzij een programmastructuurdiagram. Dat leveren ze vervolgens in en dan krijgen ze van mij een ontwerp en dat moeten ze vertalen naar een programmeertaal. Dat zijn de drie stappen.

[04:47]

R: En dit zijn allemaal individuele opdrachten?

I: Ja, dit moeten ze individueel doen.

R: Hoeveel tijd hebben ze voor ieder ding?

I: Ze hebben daar in totaal drie uur de tijd voor. Dat is waar ik dus tegenaan loop.

R: Zoveel kan je niet programmeren in dat anderhalf uurtje of die twee uurtjes[, bedoel je]?

I: Het programmeren op zich, dat [kost] heel weinig tijd, want de basis is bij mij gewoon het kunnen vertalen van sequenties, selecties, en iteraties naar een programmeertaal.

R: Dus dat is een hele simpele opdracht?

I: Het is waarschijnlijk iets van de gebruiker voert twee woorden in en jouw programma moet bepalen of die twee woorden een anagram zijn van elkaar of zoets, en de oplossing daarvoor bedenken is best wel lastig, dus alleen dat, als je zo’n ontwerp krijgt en zou moeten vertalen naar een programma, dat zijn niet meer dan een stuk of twintig programmeerregels of zo, dus dan gaat het echt om of je die taalconstructie ook beheerst en daarmee is

eigenlijk het programmeren is heel erg simpel; dat probeer ik ook [nog] zo simpel mogelijk te laten zijn in deze fase.

R: Je geeft een schema en daar komt inderdaad nog maar één antwoord eigenlijk uit, ja

I: Ja, je zou bij wijze van spreken, als je een generator hebt als Structurizer bijvoorbeeld, [de code door die generator kunnen laten coderen, want] die genereert daar automatisch code voor. Dat mogen zij dan niet doen, ze moeten het zelf coderen. Zo simpel is het eigenlijk.

[06:09]

R: Ja en dat, die eerste stap, daar kunnen wel veel meer oplossingen uitkomen?

I: Ja, die van die oplossingen, dáár zit de uitdaging in. Daar zit ook een stukje creativiteit in.

R: En doe je ook nog iets met PO's of zo, iets dat ze langere tijd moeten programmeren aan een opdracht?

I: Nee

[06:25]

R: Want je gaf ook al aan dat je het moeilijk vond om bijvoorbeeld een criteria, de kwaliteitscriteria [te bepalen] en of je code helder kan oopschrijven. Ik zal hem er even bijpakken. Even kijken, “Een leerling kan een programma zodanig structureren dat deze door anderen gemakkelijk te begrijpen en te evalueren is.”. Als jij alleen een paar ‘if-then-else’-jes [laat gebruiken in het programmeren], [hoe behandel je dit leerdoel dan?]

I: Ja maar kijk, dat ik die moeilijk vind, waarschijnlijk omdat ik er nu nog helemaal geen aandacht aan [besteed]. Ze maken dat ontwerp en ik beoordeel of het een correcte oplossing is en hoe toegankelijk hij is voor anderen, dat zit er in.

R: Volgens mij wordt hier ook bedoeld van een programma zodanig gestructureerd dat je functies gebruikt, goede naamgeving, dat je ‘if-then-else’-jes netjes opzet, dat je niet een hele rits van rare geneste dingen, of dat je een ‘for’-loop gebruikt waar dat kan, maar als je dat met een stroomdiagram al behoorlijk de richting hebt dichtgetimmerd, dan denk ik dat je daar ook heel weinig variatie in krijgt.

I: In ieder geval bij [het] programmeren niet. Bij programmeren is het gewoon eenduidig, want kijk, ik knip het programmeren ook op in twee stukken dus het eerste stuk richt zich eigenlijk op het maken van een algoritme voor een klein probleempje. Pas bij Programmeren 2 wat zij in het tweede leerjaar krijgen, hetzij HAVO 5 of VWO 5, daar ga ik iets meer de diepte in en dan ga ik bijvoorbeeld met subroutines werken en ook met complexere datastructuren, dus dat zit bij mij niet in de basis van het programmeren.

R: En dat toets je ook gewoon op hetzelfde soort manier?

I: Ja, alleen dan in het kraken van een algoritmisch probleempje, dus

bijvoorbeeld het verzinnen van een oplossing voor twee woorden of die wel of niet anagram zijn van elkaar. Het oplossen van het probleem is niet meer de crux van de opdracht die ze hebben, maar meer bijvoorbeeld hoe kan je een probleem opknippen in deelproblemen, die deelproblemen leveren dan subroutines op, dus daar ligt veel meer het accent op hoe kan je hele grote probleem vertalen naar deelprobleempjes. Het heeft iets meer nadruk op functionele compositie en het technisch gebruik van subroutines met parameter overdracht, return-waardes, functiewaardes.

R: Het feit dat ze dat gebruiken is volgens mij wel een vertaling van [dit leerdoel].

I: Ja, dat komt daar in de buurt. En dan eigenlijk die twee dingen samenbrengen, het nadenken over het oplossen van problemen en dat op een gestructureerde manier doen, dan kom je weer bij mij eigenlijk bij de derde module die ze krijgen en dat is algoritmiek. En dan gaan we aan de slag om inderdaad complexere problemen op te lossen en dan moet je al die kennis en vaardigheden van Programmeren 1 en Programmeren 2 kunnen toepassen en dan gaat het over de voor jou bekende problemen als minimale spannende boom, kortste pad, [enzovoort].

R: En hoe toets jij dat?

I: Conceptueel; tot nu toe heb ik dat steeds gewoon gedaan met een toets, een papieren, theoretische toets en ze moeten van een van de opdrachten, en dan spreken we af welke opdrachten, dan moeten ze daar natuurlijk een code voor opschrijven en dan mogen ze zelf bepalen welke programmeertaal ze pakken.

R: En dan gewoon zo'n graaf uitprogrammeren?

I: Ja, en dat kan ook, want bij Programmeren 2 hebben ze wat complexe datastructuren gehad, zoals bijvoorbeeld arrays en dat soort dingen.

R: Wel een lekker uitdagende opdracht

I: Ja, dat is zo. Dat is ook niet voor alle leerlingen, dat is echt alleen voor de leerlingen die daarvoor kiezen, en dat is allemaal een beetje voorsorteren richting het kiezen.

R: Die module kiezen?

I: [Ja,] algoritmiek, dat is een keuzemodule; Programmeren 2 dat is verplicht.

[11:03]

R: Datastructuren, hoe doe je dat?

I: Dat zit bij mij in Programmeren 2. Programmeren 1 gaat bij mij om de elementaire datastructuren, gewoon getalletjes, komma, getalletjes, woorden, karakters, booleans, dat zit allemaal in Programmeren 1, dus die programmeeropdrachten die ze krijgen bij Programmeren 1 zijn ook relatief eenvoudig qua datastructuren. Ik kan dus bij Programmeren 1 bij mij ook nog niet gaan werken met bijvoorbeeld het zoeken van een minimale waarde in een lijst van gegevens, want ze hebben nog geen lijst gehad en dat komt pas bij Programmeren 2.

R: Hoe toets je dat?

I: Dat is een individuele vaardigheidsopdracht die ze daar ook weer krijgen, dus dat is ook weer dat ze een aantal uur de tijd krijgen om gewoon iets te programmeren.

R: En dan kijk je of ze een lijstje kunnen maken en of ze daar dingen in kunnen zetten en uit kunnen halen?

I: Ja en daarbij specifieer ik wat die gedaan moet hebben en de oplossingsrichting beschrijf ik ook al, dus dan hoeven ze niet na te denken over hoe je in vredesnaam dat zou kunnen aanpakken, alleen daar gaat de nadruk over hoe je dat nou kunt vertalen naar een programmeertaal. Het maken van zo'n schematische oplossing als een programmastructuurdiagram of een flowchart, dat kunnen ze dan zelf doen. Ik beschrijf in woorden hoe de oplossing eruitziet. De tussenstappen van de schematische weergave kunnen ze dan zelf doen, maar uiteindelijk leveren ze op de programmacode die daarbij hoort.

[12:35]

R: Ja, want je gaf aan dat [je bij] datastructuren best wel makkelijk inzicht kan toetsen, en dat er één antwoord meestal mogelijk is.

I: Ja, en dat heeft alles te maken met de vraagstelling die ik [gebruik]. De vragen die ik stel zijn wel zo specifiek dat [de correcte antwoorden] wel de meest voor de hand liggende oplossingen [zijn] en ik ben daar nog niet tegenaan gelopen dat de leerlingen met iets aankwamen van 'goh wat moet ik hier nou mee'. Maar dat heeft dus alles te maken met het formuleren van een vraag; vaak is het antwoord dan al zo helder dat de formulering van de vraag eigenlijk al uit [gaat] van het antwoord wat ik wil.

R: Ja, maar dat komt ook omdat je dat zo mooi hebt opgesplitst.

I: Ja, dat geeft me in ieder geval een instrument óm het goed in kaart te brengen. Of ik het nou goed zie, dat weet ik nog niet. Ik mis nog vanalles maar ook ten aanzien van het examenprogramma zit er ook nog wel wat uitdaging in hoor. Ik mis nog wel wat.

R: Want zo'n datastructuur, volgens mij zou een leerling moeten bepalen, zelf moeten bepalen wat voor 'n datastructuur efficiënt en correct is, maar ja, als er maar een voor de hand liggend is, dan ja.

I: Ja, klopt.

[14:16]

I: Richting bijvoorbeeld automaten – dat is ook een van de dingen die erbij staat – dat is bij mij nu nog onvoldoende belicht en zeker in het stuk van het programmeren, dus dat zou ik ook nog

R: Het programmeren van data-, van die automaten?

I: Nou, automaten ja

R: Oké, je doet niks met physical computing of zo?

I: Jawel, maar physical computing is een keuzemodule

R: Ja, ooh, oh, ja, ja, jij vind alles, jij laat al je leerlingen heel veel dingen

kiezen

I: Ja, kiezen, als je geen physical computing kiest, dan heb je geen automaten.

R: Ja, nee, ja, ja

I: Ja

R: Ja dat is lastig

I: En ik vind dat een hele logische plek om automaten in te behandelen

R: Ja

I: Ja, het is gepresenteerd als een keuzemodule, dus

R: Ja, oké interessant

[15:07]

R: Toen je het ding [de enquête] invulde, toen kreeg je heel veel pagina's met vragen, op een gegeven moment kwam je bij de laatste en toen viel het me op dat je overal neutraal hebt ingevuld.

I: Ja, volgens mij ging dat over een onderwerp waar ik niet sterk in zit, maar dat heeft alles te maken met dat ik de vraag niet goed gelezen heb, 'ontwerp', daar heb ik me door laten misleiden. Want [ik doe] gewoon nauwelijks iets met dat onderwerp, dus ik zag het woord 'ontwerp' staan en dacht ja oké
R: Oké, kunnen we daar misschien nu even nog naar kijken?

I: Voordat ik het [vergeet], moet ik even nog [iets vragen]. Heel vaak stond er in de vragenlijst of het geschikt was voor NT'ers of CM'ers. Die vond ik heel moeilijk om te beantwoorden omdat, als ik koos voor neutraal, dan zeg je eigenlijk is het niet uitgesproken voor NT'ers of voor CM'ers maar voor iedereen en ik wist niet wat je bedoelde met CM'ers. Is dat specifiek juist voor CM'ers dus niet NT'ers, of is dat óók heel erg geschikt voor CM'ers?

R: Ja

I: En ik heb als het óók heel erg geschikt is voor CM'ers neutraal ingevuld.

R: Oké, heel goed

I: Dus als iets specifiek voor CM'ers is en dus niet voor NT'ers, dan zou ik geneigd zijn om het bij CM neer te zetten.

R: Is goed

[16:36]

R: Als wij hiernaar kijken, lees nog maar even een keertje voor.

I: De leerling kan voor een gegeven situatie een (digitaal) product ontwerpen, implementeren en evalueren.

R: Ik vind het moeilijk om dit te toetsen, op een schaal van 1 tot 5?

I: De vraag is wat hiermee wordt bedoeld; ik heb de betekenis van die vraag nog niet helemaal scherp. Kijk, als het is dat je 'n probleem voorgelegd krijgt, ik noem maar weer even als voorbeeld zo'n 'ik wil bepalen of een woord een palindroom is' bijvoorbeeld, maak daar wat voor, ontwerp daar eens een oplossing voor. Daar kan ik wat mee. Als je een algemeen probleem, een maatschappelijk probleem of een functioneel probleem binnen

je organisatie benoemt, ‘we willen gaan bijhouden hoeveel uren leerlingen te laat komen bij ons op school, ontwerp daar een systeem voor, dan vind ik het lastig worden.

R: Het gaat om de vaardigheden; het hoort niet echt specifiek bij user interactie en dat soort dingen waar die GUI’s in voorkomen. Het is eigenlijk weer zoets van het gaan over allemaal, het is een vaardigheid, het gaat eigenlijk over alle domeinen heen.

I: Ja, en als het over alle domeinen heen gaat, vind ik het lastig, want dan praat je zo van ontwerp, algemene ontwerpvaardigheden die zitten heel lastig. Daar heb ik heel veel moeite mee.

R: Wat maakt dat zo lastig?

I: Ja, misschien ben ik wel echt informaticus, ik vind dat heel lastig. Ik heb daar nog geen beeld bij, hoe ik dat meetbaar zou moeten maken. Ik weet dat gewoon niet.

R: Het is ook lastig, want het is net zo als dat je een bepaald kleuren kiest, wat is mooi, wat is niet mooi.

I: Ja, daar loop ik een beetje tegenaan. Ik heb daar gewoon een beetje moeite, ik heb daar geen instrumentarium voor, dus het kan misschien heel goed testbaar zijn, maar ik weet het niet, ik heb het nog niet. Ik heb er ook nog geen tijd aan besteed om dat goed in kaart te brengen.

R: Dus op een schaal van 1 tot 5, 5 is moeilijk, wat zou je het geven?

I: Nou ik zou het nu heel moeilijk vinden. 5 is heel moeilijk hè?

R: Ja

I: Ik zou het heel moeilijk vinden omdat ik daar gewoon, ik zou nu gewoon niet weten hoe in algemene termen.

[18:53]

R: Zou je het met praktisch werk of een schriftelijke toets afnemen?

I: Dat zou ik wel met praktisch werk afnemen.

R: En als 5 is praktisch werk en schriftelijk een 1?

I: 4 denk ik.

R: Ik zou het individueel of groepsgewijs [afnemen]?

I: Groepsgewijs, 5.

R: Het maken van de toets kost me heel veel tijd?

I: Ja, want ik moet goede ideeën hebben, dus eigenlijk het genereren van ideeën, goedbruikbare ideeën zou me veel tijd kosten. Als ik de ideeën eenmaal heb, dan kost het maken niet meer zoveel tijd.

R: Dan typ je het zo gewoon even uit?

I: Ja

R: Maar [zoals] de casus die je zojuist noemde, zoets bijvoorbeeld?

I: Nou kijk, die casus die ik noemde; Qua grotere problemen formuleren die echt bruikbaar zijn; soms heb ik weleens een ideetje, maar dan gaandeweg blijkt [dat] daar allemaal complicaties bij [zitten]; achteraf blijkt [het] helemaal niet zo'n gelukkig idee. Als het gaat om weer zo'n klein

probleempje richting het programmeren, verzin een oplossing voor het bepalen voor, of een woord een palindroom is, van dat soort algoritmische dingetjes, daar heb ik genoeg ideeën bij, dus dat kost me niet zoveel tijd.

R: En zoieters als dit, op schaal van 1 tot 5?

I: Groter dat vind ik heel moeilijk, dat vind ik heel lastig, 5.

R: Het objectief beoordelen van het werk?

I: Ja, dat vind ik heel moeilijk, 5.

[20:22]

R: Hoeveel tijd kost het je om dat soort dingen te beoordelen?

I: We hebben laatst een project gehad met ook iets waar ik totaal onbekend mee was, iets van ‘psychonic py’ waar ik geen criteria van had of wat dan ook. Dit is ook iets waar ik op dit ogenblik geen criteria voor zou hebben en dan ga je het beoordelen. Als je geen criteria hebt, dan is het ook vrij makkelijk te beoordelen.

R: Een beetje meer natte vingerwerk?

I: Ja, klopt, om het te beoordelen is het eenvoudig, maar het is een kwalitatief waardeloze beoordeling.

R: Kost het je dus veel tijd?

I: Nee

R: Dus is dat het dan een 1 of een 2?

I: Een.

R: Uhm uh, CM of uh NT-profiel?

I: Nou, hiervan heb ik dit, als het gaat om het ontwerp, dat zou CM, dat zou zeer nadrukkelijk richting CM gaan.

R: En helemaal een 5 of uh, 4?

I: Uhuhuh, nou uh, als je het op een schaal van NT naar CM doet, dan zou ik hem wel helemaal aan de kant van CM neerzetten.

R: Oké, een 5.

I: Ja, ik denk het wel.

R: Vind je het moeilijk om inzicht te toetsen? Hoe moeilijk vind je het om inzicht te toetsen?

I: Op dit onderwerp vind, zou ik het wel moeilijk vinden, ook wel 4 of 5, denk ik.

R: Maak een keuze.

I: 5.

R: Één antwoord mogelijk of zijn er meerdere antwoorden mogelijk?

I: Liefst heb ik meerdere antwoorden mogelijk, uitgesproken heel veel.

R: Uitgesproken heel veel?

I: Ja

R: Weet je wat de bekende misconceptions of fouten zijn die leerlingen hierbij maken?

I: Helemaal niet, geen flauw idee.

R: Op een schaal van 1 tot 5?

I: Helemaal niet.

[22:12]

R: Cijfers liggen helemaal ver uit elkaar of dicht bij elkaar?

I: Zit dit in de handreiking van de SLO binnen het kernprogramma?

R: Ja, [domein] A.

I: Dus alle leerlingen zouden dit moeten doen?

R: Alle leerlingen moeten het doen. Je zei net [dat je het] beoordeelt het met een beetje natte vingerwerk, maar betekent dat dat het iedereen zevens, van een zes tot een acht krijgt?

I: Nee

R: Of ben je iemand die bijvoorbeeld ook iemand een 3 zou kunnen geven?

I: Ja

R: Oké, dus je kan een 3 kunnen krijgen, maar ook een 9, of een 8.

I: Ja

R: En hoe extreem op een schaal van 1 tot 5? 5 is liggen heel ver uit elkaar.

I: 4. Sorry, nee 3.

R: Dus 3 ligt er een beetje middenin.

I: Ja, voor het grootste deel.

R: En ik vind het makkelijk om nieuwe vragen te maken voor nieuwe toetsen met hetzelfde leerdoel?

I: Dat vind ik moeilijk dus.

R: Daar hadden we het net ook al over, echt een 5?

I: Ja

[23:26]

R: Het viel me op dat sowieso qua cijfers ver uit elkaar en dicht bij elkaar, dat je [daarbij] bij alle leerdoelen ongeveer een beetje neutraal zat.

I: Bij het beoordelen?

R: Ja, bij het beoordelen.

I: Ja, dat betekent dus dat volgens mij loopt het niet extreem uit elkaar, maar wel wat.

R: Dus iedereen haalt gemiddeld een 6 of een 7 en er zitten er een paar boven en onder?

I: Ja

R: Zie jij verschil tussen welke domeinen je toetst of zie je dat niet?

I: Ja, wanneer het wat meer gaat om informaticatechniek, dan liggen de cijfers wat meer bij elkaar. Wanneer leerlingen wat meer afhankelijk zijn van inzicht, dan liggen de cijfers wat verder uit elkaar.

R: En wat bedoel je met techniek dan? Informaticatechniek?

I: Nou bijvoorbeeld SQL. Als ik 'n toets over SQL geef, dan liggen de resultaten meer bij elkaar.

R: Heel herkenbaar

I: Daar zijn leerlingen wat minder afhankelijk van hun algemene vaardigheden. Als ik ze bijvoorbeeld een presentatie laat maken over informatiestromen binnen de school, dan kan het gebrek aan inhoud bij sommige leerlingen nog weleens wat worden gecompenseerd door dat je het goed in kaart hebt gebracht of helder hebt verwoord. En in het eendoordeel krijg je dan wat minder punten voor de techniek en wat meer punten voor de wijze waarop je het vertelt, en dan komen de cijfers daardoor iets meer bij elkaar, dus dan kunnen de leerlingen iets beter een gebrek aan kennis compenseren.

R: Ja, maar dat is ook prima, want dat zijn vaak ook vaardigheden die getest moeten worden.

I: Ja, maar uiteindelijk levert één product maar één beoordeling op.

[25:30]

[short interruption]

[25:48]

R: Er was nog een vraag over standaardalgoritmes, zoals de min en de max bepalen, dat soort dingen. Hoe vindt je het om de inzicht daarvan te toetsen? Dus of een leerling daar inzicht in heeft? Ik weet niet meer wat je precies had geantwoord. Ik kan wel even kijken, dus hoe toets je dat überhaupt dan?

I: Ja, weet je, dat zit bij mij er heel zwak in, want kennis over de standaardalgoritmes zit er heel weinig in, want de nadruk bij mij in het materiaal zit nog steeds in het zelf ontwikkelen van een oplossing. Bij mij zijn de opgaves die ze krijgen, eigenlijk allemaal problemen die ze voorgelegd krijgen waar ze een oplossing voor moeten bedenken. Bij mij in het materiaal op dit ogenblik zit nog heel weinig ‘dit algoritme doet dat’, dus de eerste besprekking van een algoritme in een verhaal en de standaardoplossing daarvoor, dus dat is bij mij op dit ogenblik onderontwikkeld.

R: Oké, als je dat uh een cijfer van de diepgang hè, de diepgang van kennis van iemand die daar, uhm, hoe makkelijk vind je het om dat te beoordelen?

I: Nou dat lijkt me héél erg makkelijk om te beoordelen.

R: Vertel

I: Dat doe ik nu bijvoorbeeld met sorteeralgoritmes wel, denk ik, en dat doe ik aan de hand van een voorbeeld, naar mijn gevoel is het heel makkelijk. Wat ik doe, is ik geef ze een rij getallen en ik vraag aan de leerlingen vertel me nu even met dit sorteeralgoritmes welke vergelijkingen worden uitgevoerd. En door dat te zeggen zie ik of de leerling het algoritme begrijpt. Stel dat het een standaardalgoritme zou zijn om van een woord te bepalen of het een palindroom is of niet, of ik geef zelfs het algoritme waarbij de letters met elkaar worden vergeleken van buiten naar binnen toe totdat je bij het midden bent aangekomen, en als alle letters gelijk waren aan elkaar, dan is het een palindroom. Dan zou ik een vraag kunnen stellen van ja uh maar hoe had dit wat sneller gekund. Dat is een controlevraag waaruit bij heel veel

standaardalgoritmes blijkt, dat het heel makkelijk is om een controlevraag te stellen waar het begrip uit blijkt, dus daar ben ik nog nooit tegenaan gelopen dat inzicht in een algoritme dat dat een probleem heeft opgelost.

R: Oké

[28:20]

[einde van het MemberChecking interview met James]

G.3.4 John

[00:17]

R: In de vragenlijst was er een perceptie die ging over of u een leerdoel beter vindt passen bij leerlingen met een NT-profiel of bij leerlingen met een CM-profiel. Daarbij heeft u eigenlijk voor vrijwel ieder leerdoel ‘neutraal’ ingevuld. Kan u dat verklaren?

I: Ja. Nou, het gaat eigenlijk vooral om die algoritmen hè, en ik denk eigenlijk van – en dat is meer een wens dan een gedachte – want van eigenlijk al die dingen rondom algoritmen vind ik van daar moet je een manier voor vinden dat het voor iedereen toegankelijk is, dus daarom; het is meer ingevuld vanuit het idee van een ontwikkelaar van ‘ik wil niet voor een profiel iets maken; ik wil voor iedereen iets maken’.

R: Dat begrijp ik.

I: Ja

R: En in hoeverre denkt u dan van dat dat – tot nog toe, zeg maar – al lukt om leerdoelen, toetsopdrachten passend te maken bij zowel leerlingen met een NT-profiel als leerlingen met een CM-profiel?

I: Nou, bijvoorbeeld, we zijn gewoon dingen aan het proberen hè, dus ik weet nog niet of het werkt, maar een van de dingen die ik gedaan heb bijvoorbeeld is – ik weet niet of je ook in InformaticaActief kunt kijken bijvoorbeeld – want zo af en toe doe ik dingen formuleren op verschillende manieren, dus bijvoorbeeld er is een onderwerp dat gaat over bijvoorbeeld zoekalgoritmen en sorteeralgoritmen en die zijn zo gemaakt dat niet alle algoritmen gegeven worden, maar eigenlijk dat meer dingen worden – dat noemen ze ‘guided discovery teaching’ - van je helpt ze een beetje om het algoritme zelf te ontdekken, en geleidelijk aan ben ik er naartoe gegaan dat ik, het formuleren van algoritmen in het Nederlands vind ik gewoon helemaal prima, maar dan zit er een tweede opdracht bij en die houdt eigenlijk in van maak nu het stroomdiagram zó, dus met getallen en variabelen erin, van dat je hem goed kan programmeren, en met andere woorden die is dan gericht op leerlingen die eigenlijk direct willen gaan coderen, maar dat doen ze dus niet, en in beide gevallen ben je met die algoritmiek bezig, dus zulk soort dingen zitten erin, en of dat nou werkt, ja dat zal de tijd ons leren. Dat is natuurlijk ook een interessant onderzoeksonderwerp.

R: Ja, oké.

[03:06]

R: Een van de leerdoelen in de online vragenlijst is de volgende: ‘De student kan een programma zo structureren dat anderen het makkelijk kunnen begrijpen en evalueren’. Voor dit leerdoel heeft u aangegeven dat het u minder tijd kost om een toetsopdracht te maken dan voor de meeste andere leerdoelen. Kan u dat verklaren?

I: Ja. Nou, ik heb wel eens zulk soort opdrachten gemaakt, en eigenlijk waren dat opdrachten in de stijl van je hebt een duidelijk slecht gestructureerd

programma en je moet, je kunt namelijk gewoon een bestaand iets nemen en dan gewoon het programmaatje, ik heb wel eens een programmaatje gemaakt en dan alle variabelennamen vervangen door a, b, c, d, en alle regeleindes eruit halen, alle tabs eruit halen. Nou dat is vrij onleesbaar en heel gemakkelijk te maken, en dan is de opdracht voor de leerlingen van maak dat eens in orde, en dan kun je redelijk snel zien wat ze goed doen, dus die vraag is heel makkelijk te maken, en je kunt er ook gelijk aan aflezen of ze een beetje een idee hebben wat goed gestructureerd is, dus daar/dat is wel, en soms is het heel moeilijk om vragen te bedenken als van stel je gaat een vraag bedenken om een algoritme te bedenken, dan is dat heel lastig om iets te vinden wat – weet je wel – of niet veel te moeilijk is of je geeft het antwoord al half weg of – weet je wel – dat. Sommige dingen zijn redelijk eenvoudig en juist zulk soort dingen zijn, denk ik, redelijk eenvoudig.

R: Oké

[04:49]

R: Dan de volgende vraag: U heeft voor slechts twee leerdoelen aangegeven dat ze het best getoetst kunnen worden met praktisch werk in plaats van met een geschreven toets¹. Dat zijn de leerdoelen ‘De student kan redeneren over de correctheid en efficiëntie van een gegeven algoritme’ en ‘De student kan standaardalgoritmes herkennen en gebruiken’. Kan u dat verklaren?

I: Dan zou ik even moeten kijken. Wat waren die anderen ook alweer ? Dat waren degene in dat lijstje wat je in de mail gezet had?

R: Ja, [dat] klopt.

I: Nou eigenlijk, weet je dat is zo’n, laat ik gewoon zeggen wat ik denk en dat is namelijk dat zo gauw je iets doet van ‘maak iets’ of ‘beredeneer iets’, dan kom je altijd op dat terrein van dat je geen gesloten vragen wilt hebben, dus dat je meer een praktische opdracht en zulk soort dingen hebt. En andere leerdoelen zit bijvoorbeeld ook iets in van kijken of een gegeven algoritme tot een goede uitkomst komt. Dat kan je veel simpeler toetsen. Dan kan je gewoon zeggen van ‘hier heb je een algoritme, hier heb je een input, wat komt er uit is de vraag’; dan kan je gewoon een simpele gesloten vraag maken. Maar al die dingen die redeneren of maken in zich hebben, dan heb ik grote twijfels of je dat met gesloten vragen kan doen.

[06:20]

I: Aan de andere kant is het ook zo van als je die dingen die ik nu gemaakt heb bekijkt, dat is eigenlijk een poging om daar een beetje tussen in te kruipen, omdat je niet helemaal gesloten vragen hebt, en ook niet helemaal open vragen, en toch nog iets waarbij leerlingen zelf echt nog iets moeten bedenken.

R: Is het correct dat uit wat u nu net zegt, dat ik goed begrijp dat u vindt

¹As can be seen in the Focus cluster of John (see Figure E.10), the reverse is the case. During the interview, the researcher accidentally restated this finding incorrectly. The answer of John and thus the results were not affected due to this mistake.

dat een gesloten vraag beter bij geschreven toetsen horen [passen], en open vragen meer bij praktische?

I: Ja. En eigenlijk vind ik sowieso dat bij veel van dat algoritmiek, bij programmeren bijvoorbeeld – ik weet niet of je, jij studeert ook informatica of...

R: Ja, ik studeer ook informatica

I: Ja, dan heb je misschien ook ooit een eerstejaarsvak gehad met programmeren in C# of zulk soort dingen?

R: Ja

I: Daar was misschien een soort tentamen bij?

R: Ja

I: En dat was helemaal verschrikkelijk? Ik heb dan zo'n soort vak gegeven, ik vond dat ook verschrikkelijk. Ik vond dat gewoon bizar dat het zo moet. Aan de andere kant waren er zo'n a 120 studenten, dus het kon niet anders. Ik kon niet zeggen van ‘lever maar een leuk programma in’ en dan ga ik 120 programma's als docent [nakiiken]. Dan heb je ook nog universiteiten waar ze 300 eerstejaars hebben, dan wordt dat helemaal hopeloos.

R: Ja, dat kost heel veel tijd.

[07:49]

I: Ze compenseren dat wel een beetje met studentassistenten, maar je weet als je ooit zo'n tentamen gedaan hebt, hoe verschrikkelijk dat is. Dat gaat eigenlijk nergens over, en het wordt nog erger. Ik heb bijvoorbeeld nu overgenomen een vakje game design voor docenten in opleiding. Dat zijn er gelukkig een stuk of acht, en daar zat ook een tentamen bij, en volgens mij was ik gezakt voor dat tentamen, als ik dat zelf had moeten maken, en heb ik uiteindelijk gewoon gezegd van ‘Nou, we doen geen tentamen. Jullie maken maar een ontwerpdocument voor een eigen te bedenken game rekening houdend met een aantal messen’, en dat vind ik veel realistischer. En ondanks alle bezwaren die er aankleven van er wordt gekopieerd en weet ik veel wat, uiteindelijk geeft het een beter beeld, denk ik, en is ook gewoon meer toegesneden op het aard van het vak. Dus als je gewoon gesloten toetsen maakt bij maak-vakken, *[he shrugs]* ja.

R: Oké, dat is helder dan.

[09:01]

R: Een van de leerdoelen in de vragenlijst is ‘De student kan de oplossing van een probleem vertalen naar een programma’. Hiervoor heeft u aan de ene kant aangegeven dat u het vrij makkelijk vindt om inzicht te toetsen en dat u vrij veel weet over de misconcepten en veelgemaakte fouten bij dit onderwerp, maar u heeft ook aangegeven dat het maken van een toets hiervoor u vrij veel tijd kost. Kan u dat verklaren?

I: Nou ja, je hebt natuurlijk misschien een beetje gerommel met in[vullen]; je bent niet helemaal consequent, dus voor een rechtzaak zou de vragenlijst geen stand houden. Maar het bedenken van een open opdracht voor een

maak-vak is voor docenten vaak heel makkelijk, want ik zeg namelijk gewoon, bijvoorbeeld bij een project deden we dat is bedenk een leuk iets, sluit even kort met mij – een soort go/no go van dat het niet veel te weinig/veel te veel is – en ga het doen. Daar heb je als docent niet zoveel werk aan. Aan de andere kant, als je dus meer een tentamenvorm moet doen, dan wordt het dus krankzinnig moeilijk, omdat je bij alles rekening moet houden van dit moeten ze in een uur opschrijven, en is het niet te veel, is het niet te weinig, kunnen ze niet totaal de mist in gaan – dus dat ze iets heel anders erbij gaan bedenken – en in de tijd dat ik dat programmeervak gaf, zag je dat ook met enige regelmaat gebeuren dat je een stukje code vroeg van vijf regels, en dat werd niet helemaal begrepen, en je krijgt er 50 terug, omdat ze iets heel anders hebben bijbedacht eigenlijk, en dat is dus heel lastig. Dus vandaar die splitsing die erin zit. Dus als je het dan gaat proberen dicht te timmeren, zo'n toets, dan wordt het heel moeilijk.

R: Ja, oké.

[11:04]

R: Een van de andere leerdoelen is ‘De student kan voor een gegeven algoritme een programma ontwikkelen in een imperatieve programmeertaal’. Hierover heeft u gezegd dat u vrij veel weet over de misconcepten en veelgemaakte fouten.

I: *[he nods in agreement]* Ja.

R: Maar ook dat u het toetsen van inzicht hiervoor vrij moeilijk vindt.

I: Ja, dus dat toetsen, dat is eigenlijk bijna hetzelfde als de vorige, want – en daarbij was ik even die misconcepten vergeten – voor die toetsen geldt eigenlijk precies hetzelfde verhaal als net, en die misconcepten, ja daar, ik heb dus heel lang gewoon een eerstejaars programmeervak gegeven, dan kun je dus op een gegeven moment een hele rij opstellen van dingen die in het begin fout gaan. Een van de hele, op zich als je OO programmeren [OO: ‘object oriented’, object georiënteerd] doet, is het verschil tussen static en niet-static. Dus als je aan honderd studenten dat vak geeft, dan zijn er dus ieder jaar zo'n dertig of veertig die in hun eerste programma volledig de mist in gaan, omdat ze iets niet goed, een variabele bijvoorbeeld, maken die vanuit de main – een static methode – wordt aangeroepen, dan gaat die compiler zeuren van ‘je kunt niet aan een instance variabele vanuit de main’, en die zegt dan ‘zal ik de variabele static voor je maken?’, en dan krijg je dus van die programma’s waarin alles static is en die het niet meer doen, en dat merken ze dan pas na een tijdje vaak. Dus zulk soort misconcepten die zie je voortdurend gebeuren. Die zie je – denk ik – als je een standaardlijstje neemt – je hebt weleens van die documenten gezien van ‘Misconceptions’ - die staan daar ook allemaal in. Dat zijn dan vaak dingen die in zo'n tentamen, die je wel redelijk kunt vragen, en ik doe dus altijd – om in dezelfde context te blijven – heel vaak had ik dan een vraag van ‘wat betekent het keyword static?’ en dan was een van de alternatieven ‘je mag hem niet veranderen’ en

in plaats van dus een klasse variabele en iets van een instantie, dus hij is voor iedereen gelijk, en dat heb ik dan ook weleens als een open vraag waaraan je redelijk kunt zien of iemand het begrepen heeft en die toch makkelijk na te kijken is, dan is het een open vraag, dus je kijkt erna en je weet het, of ze het doorhebben.

R: Ja, oké.

[13:50]

[End of the MemberChecking part of the interview with John]

G.3.5 Michael

[Before the interview itself starts, tells Michael explicitly that he does not teach at the moment, but that he has taught programming and mathematics for five years at the "HTS Amsterdam"]

[During the introduction of the interview, Michael tells the following to the researcher about the learning objectives]

[02:11]

I: Het merendeel ervan, dat is wel grappig, met name de gebieden waar ik me in de opgaven op focus, dat herinner ik me. Die andere dingen herken ik wel, maar herinner ik me niet.

[Start of the actual interview]

[02:33]

R: In de online vragenlijst had u eigenlijk bij alle leerdoelen aangegeven dat ze het beste individueel getoetst kunnen worden. Doet u dan ook alles individueel toetsen, in de tijd waarin u les heeft gegeven?

I: Ja, dit is een beetje zo'n situatie van ja op dit moment geef ik geen les, dus die vraag kan ik moeilijk beantwoorden.

R: Hoe zou u het nu doen dan?

I: Als ik nu les zou geven, zou ik het individueel doen. [Het is] mogelijk dat een van de – ik ben van plan om de lerarenopleiding nog te doen – mogelijk dat ik daar dingen hoor en leer die me van inzicht doen veranderen, maar ik geloof niet zo in groepswerk, eerlijk gezegd. [Dat] doe ik in mijn eigen bedrijf ook niet.

R: Is dat dan ook de reden dat u – volgens de online vragenlijst – vindt dat leerdoelen het beste met een schriftelijke toets getoetst kunnen worden?

I: Correct. Ja.

R: Oké.

[03:45]

R: De volgende is meer een stelling. Bent u het eens of oneens met de volgende stelling: ik vind het even moeilijk om toetsopdrachten te maken en leerlingenwerk na te kijken voor het leerdoel 'De student kan verschillende datastructuren vergelijken en evalueren in termen van elegantie en efficiëntie' als voor de meeste andere leerdoelen?

I: Ik zit even te denken, efficiëntie en elegantie zei je, hè?

R: Ja, elegantie en efficiëntie bij verschillende datastructuren.

I: Of ik die moeilijker vind, nee, die vind ik niet moeilijker, nee.

R: Is niet moeilijker dan ook niet makkelijker?

I: Ik denk dat het wel makkelijker is, omdat ik daar gewoon meer focus op heb. Dan moet ik ook eerlijk zeggen, daar heb ik ook meer ervaring mee, dus ik neig naar gemakkelijker, ja.

R: Dat is dan wel opvallend, want in de online vragenlijst

I: Oh ja, wacht, ik weet wat je wil gaan zeggen. Dat wordt veroorzaakt

door elegantie. Efficiëntie van datastructuren is niet zo moeilijk, denk bijvoorbeeld aan hoeveel vergelijkingen moet je doen in een gebalanceerde boom; als jij, dus stel je zoekt iets, en je hebt die zoekgegevens in een gebalanceerde boom opgeslagen, hoeveel vergelijkingen moet je doen om een knoop te vinden die in die boom zit versus hoeveel vergelijkingen heb je nodig in een gestructureerde lijst. Ja, zo kun je hier heel veel op variëren, dat is vrij gemakkelijk. Over elegantie van datastructuren, dat vind ik een stuk lastiger, dus daar ben ik mijn hoofd ook nog over aan het breken. Wat Renske en Martin ook zeggen is van ‘nou, laten we eerst even voor elegantie gaan en efficiëntie later doen’, dus ik denk dat zij hetzelfde denken van ik kan heel makkelijk vijf, zes, zeven, acht, negen, of tien voorbeelden van efficiëntie uit mijn mouw schudden, maar het lukt me niet om tien voorbeelden van elegantie te noemen.

[06:26]

I: Ja, want probeer het eens; je zit in het derde jaar, je hebt al aardig wat datastructuren door de hand gehad, geef eens een voorbeeld van een onelegante datastructuur.

R: Ja, dat ligt ook heel erg aan het voorbeeld dat je neemt, want afhankelijk van de situatie zijn ook andere datastructuren eleganter of niet.

I: En met de situatie bedoel je dan het onder handen hebbende probleem?

R: Ja.

I: Ja.

R: Maar goed, het ligt dus vooral aan de elegantie?

I: Ja, correct.

[07:08]

R: Zijn er dan meer van dat soort dingen die u lastiger vindt dan andere, zoals dus bijvoorbeeld die elegantie?

I: Ik sluit het niet uit, maar ik kan er zo geen – ik zit even te kijken – ja waar ik natuurlijk al – even kijken, hoe lang ben ik daar al afgestudeerd; ah, nog net geen dertig jaar, 29 jaar geleden – ik heb in die 29 jaar niet zo heel veel gedaan met grammatica's, en talen en die eindige automaten; nou ben ik theoretisch afgestudeerd, dus dat heb ik zo weer in de hand. Maar ik kan me voorstellen dat ik in die hoek, dat ik daar wat meer moeite, dat ik daar wat langer over zou moeten puzzelen, dus aan de eindige automatenkant en de taaltjeskant, en daar zou ik ook wel wat meer moeite hebben.

R: Ja.

[08:24]

R: Nu we het over de automaten hebben, het viel me op u heeft aangegeven in de online vragenlijst dat u het toetsen hiervan vrij moeilijk vindt, maar u heeft wel aangegeven dat u het vrij makkelijk vindt om inzicht hiervoor te toetsen, dat u verwacht dat het maken van toetsopdrachten hiervoor u vrij weinig tijd kost, en ook dat het nakijken van leerlingenwerk u niet veel tijd kost. Kan u dat verklaren?

I: Hoe ik daarbij denk - ik snap het dualisme - je kunt heel moeilijke dingen verzinnen en je kunt het vrij [makkelijk houden], ja, het probleem is vaak dat talen en eindige automaten over een kam worden geschoren; je kunt namelijk een taal als een eindige automaat zien, maar een eindige automaat kan ook iets, als ik een voorbeeld geef van een eindige automaat, dan heb ik het altijd over – god, waar heeft Renske het nou altijd over, nou ben ik het kwijt – zelf heb ik het altijd over een koffiezetapparaat, dus een koffieautomaat zoals die ook bij jullie [informatica studenten aan de Radboud Universiteit] op de faculteit staat; jullie hoeven er geen geld in te doen, jullie hoeven alleen een keuze te maken en dan maakt ‘ie het), en dat is een mooi voorbeeld van een eindige automaat. Als je zo’n eindige automaat hebt, dat spreekt tot de verbeelding, en dan kun je met mensen, met leerlingen, heel gemakkelijk zeggen van ‘Wat voor toestanden zou die automaat hebben?’, nou ja, kapot, in onderhoud, ik ben bezig een bestelling in te geven, ik ben bezig geld in te geven, alles is goed: hij gaat over naar de staat van het product maken, of in de staat van een error: de koffie is op, of de melk is op, of er is onvoldoende waterdruk, en zo hebben mensen vrij veel fantasie. Maar jij weet ook, denk ik, dat eindige automaten ook heel veel abstracter uitgelegd kunnen worden en je er ook heel veel moeilijkere dingen mee kunt beschrijven, en afhankelijk van wat ze willen kan het heel moeilijk worden. Maar ik vind het ook, zoals ik net zei, koffiezetautomaat, snoepjesautomaat, en de deur van een publiek toilet waar je geld in moet doen, dat zijn allemaal voorbeelden die je heel gemakkelijk kunt modelleren met een eindige automaat die tot de verbeelding spreekt.

[11:13]

I: Als jij grammatica’s en eindige automaten gaat combineren, dan wordt het in een klap een stuk abstracter, dan kun je ook veel moeilijkere dingen gaan doen, en dan zeg ik van nou, dan wordt het toch echt wel wat pittiger.
R: Ja, oké.

I: Dat is een beetje wat ik bedoel.

[11:34]

R: Dan is er nog een andere stelling, zoals net. Bent u het eens of oneens met de volgende stelling: ik vind het even moeilijk om toetsopdrachten te maken en leerlingenwerk na te kijken voor het leerdoel ‘De student kan een digitaal product ontwerpen, evalueren en implementeren voor een gegeven situatie’ als voor de meeste andere leerdoelen, dus een digitaal product ontwerpen, evalueren en implementeren.

I: Oh, ik heb geen flauw idee wat ik daarop heb geantwoord. Ik stoei een beetje met ‘een digitaal product’ implementeren en dat kan ook een robot zijn of een automaat of een – wat kan het nog meer zijn – een computer, een...

R: Het kan van alles zijn.

I: Ja, het kan echt van alles zijn. Kijk, ik zit in mijn hoofd – en vandaar

ook mijn voorkeur voor algoritmes en datastructuren – ja, ik zit, misschien is het voor leerlingen wel heel erg interessant om robotjes en zo te maken, daar heb ik gewoon, met die producten, die niet-informatica producten, met de niet-programmeer-gerelateerde onderwerpen heb ik gewoon veel minder – ja, dat kan ik gewoon heel bot zeggen – daar heb ik gewoon geen interesse in, daar heb ik ook gewoon geen ervaring in en affiniteit mee, dus dan wordt het automatisch óf minder leuk óf moeilijker, en daar zit tussen minder leuk en moeilijker natuurlijk een verband.

R: Ja, dat snap ik.

I: Dus ik weet niet of jij nog vakken hebt die je als laatste vak moet doen, maar dat soort vakken..

R: Ja, ik snap het wel. Ik loop aardig op schema, maar ik snap wel wat u bedoelt. Dat komt aardig overeen met wat u hebt aangegeven in de vragenlijst, dus dat is mooi.

I: Gelukkig, dan ben ik in ieder geval weer een keer consequent geweest.

[14:03]

R: Dan heb ik nog een andere vraag. U heeft in de vragenlijst aangegeven dat u het maken van een toetsopdracht over het vertalen van een probleem naar een algoritme dat dat u vrij veel tijd kost, maar ook dat u het vrij eenvoudig vindt om een toetsopdracht hiervoor te maken en dat u het ook vrij eenvoudig vindt om voor dit leerdoel inzicht te toetsen. Kan u verklaren waarom het maken van toetsopdrachten, u dat dus blijkbaar vrij eenvoudig vindt, maar dat het u toch vrij veel tijd kost of zou kosten?

I: Je hebt vrij snel een idee te pakken, maar je moet heel goed nadenken over een aantal dingen: je moet heel goed nadenken over de manieren waarop het – precies wat ik nu aan het doen ben trouwens – je hebt een idee voor een opgave en dat is op zich vrij eenvoudig; ik noemde er net bij die datastructuren, over die gebalanceerde bomen en bij die eindige automaten noemde ik zo een aantal voorbeelden, en dan krijg je het punt van oké, hoe schrijf je dit nou op en hoe makkelijk of hoe moeilijk is de vraag interpreteerbaar, maak je het dan moeilijker om extra moeilijkheid aan de vraag toe te voegen door veel tekst (dan moeten ze de tekst uitpakken), formuleer je het zo dat bétas meteen snappen wat er bedoeld wordt, maar alpha's denken van 'Huh? Wat wordt hier gevraagd?', en dat zijn allemaal valkuilen die je bij vragen hebt, en vooral als het om iets gaat als een vertaalslag van een probleem naar een algoritme; het is makkelijker om te vragen van 'hier heb je een algoritme, wat doet dit?', en als je dus vraagt van 'hier heb je een probleem en verzin eens een oplossing', dan krijg je een oplossing en dan moet je ook even terugkijken van 'wacht eens even, mijn eerste reactie is 'de oplossing is fout', maar je tweede reactie zou moeten zijn van 'heb ik de vraag wel goed gesteld?', en zeker als meerdere leerlingen dezelfde fout maken, en 'heb ik de vraagstelling wel goed geformuleerd'.

[16:43] I: Iets wat ook veel tijd kost, maar dat ligt aan mijn werkwijze, is – en

daar ben ik nou met Renske ook al tegenaan gelopen – is als ik een algoritme- of datastructuur-vraag stel, dan implementeer ik het algoritme ook, inclusief het testalgoritme, dus ik wil, kennelijk is onder leraren vrij gebruikelijk om gewoon een algoritme-vraag te stellen zonder zelf de implementatie te maken en dat doe ik niet, omdat ik zeker niet uitsluit dat ik fouten kan maken.

R: Ja, dat snap ik.

I: En dus, je vraag kan vrij eenvoudig lijken of je probleemstelling kan vrij eenvoudig lijken, maar dat de implementatie toch vrij pittig is, en daar kom je pas achter als je het doet. Dat is dus niet moeilijk, maar het kost wel veel tijd.

R: Oké, dan is dat duidelijk.

[17:54] R: U had het net ook al kort even over alpha- en bèta-leerlingen, maar u heeft in de vragenlijst voor ieder leerdoel neutraal ingevuld bij de vraag of het leerdoel beter past bij leerlingen met een NT-profiel of bij leerlingen met een CM-profiel. Kan u dat verklaren?

I: Nou, dat zit er meer in, zeg het nog eens

R: Of het leerdoel beter past bij leerlingen met een NT-profiel of bij leerlingen met een CM-profiel.

I: Kijk, ik had kunnen zeggen van het is evident dat wiskunde makkelijker is voor de bèta's. Ik denk dat je informatica zo kunt geven dat het voor iedereen geschikt is, en ik denk dat je dat ook moet doen, dus je zou het zo moeten maken dat mijn antwoord klopt, dat het eigenlijk niet uitmaakt. Dat is eigenlijk wat ik bedoel met mijn antwoord.

R: En u denkt dat dat ook voor alle leerdoelen mogelijk is?

I: Nee. Ik denk dat je bij sommige leerdoelen, eindige automaten, afhankelijk van de taaltjes, nou je kunt een abstractie-drempel overschrijden, waarmee – je kunt het zo maken dat het voor iedereen te moeilijk wordt overigens – , je kunt een abstractiedrempel overgaan waarmee het voor de niet-bèta's (jij noemt het NT, maar ik noem het bèta's), waarmee het voor de niet-bèta's gewoon te moeilijk wordt, en ik denk dat dat bij eindige automaten vrij gemakkelijk kan, functioneel programmeren gaat dat ook vrij gemakkelijk voor, vooral als je lambda-calculus zou gaan behandelen, dus nee, ik denk niet dat dat voor ieder leerdoel haalbaar is, nee.

R: Maar u heeft toch overal neutraal ingevuld.

I: Ja, omdat ik vind dat het zo moet zijn, oftewel daar waar het niet kan, die leerdoelen moet je wegknikkeren, maar dat wordt waarschijnlijk een pittige discussie met Erik; ik zie heel duidelijk aan het lijstje leerdoel dat Erik – je weet wel wie Erik, hoe heet Erik

R: Ja, Erik, Erik Barendsen, die ken ik.

I: Barendsen, ja. Je ziet heel duidelijk dat Erik uit de lambda-calculus hoek komt. Met name theoretische informatici en wiskundige ook, die hebben iets met elegantie en schoonheid, en dat hebben heel veel meer praktische, niet-theoretische informatici hebben helemaal niets met dat woord en dat zal

ook gewoon blijken. Ga maar straks opletten als je, je ziet het nu eigenlijk al: de keuze voor wat er uitgewerkt wordt, daar wordt elegantie naar achter geschoven en zo worden er nog een aantal dingen naar achter geschoven en dat zijn de dingen waarvan men kennelijk vindt dat de leraar dat moeilijk uit kan leggen en/of dat de gemiddelde leerling daar weinig tot niets mee kan.

R: Ja.

[22:10]

[End of the MemberChecking part of the interview with Michael]

G.3.6 William

00:55

R: Je had een leerdoel en dat was “De student kan van een algoritme een programma maken”. En nou was de vraag, wil je dit praktisch of met een [schriftelijke] toets toetsen? Je hebt toen neutraal aangegeven. Kun je dat toelichten?

I: Volgens mij heeft het te maken... Het gaat om van een algoritme naar een programma?

R: Ja.

I: Nou, ja, in het algemene zin ben ik wel voor om het gewoon in een toets te doen omdat je dan beter zicht krijgt of de leerlingen het echt geleerd hebben. Dus, als het kan zou ik het 't liefst in een gesloten setting willen doen, dus een toets van 60 minuten, weet ik wat. Omdat je dan een beter beeld krijgt.

R: Minder fraude gevoelig?

I: Nou ja, als je het in PO toest bijvoorbeeld, dan heeft zo'n leerling meer hulpmiddelen, en wat je dan ziet denk ik, in de praktijk, en meer tijd en hulp, wat dan kan gebeuren is dat hij het wel voor elkaar krijgt. De vraag is of die weer een soortgelijke opdracht zou krijgen, of die dat dan zou kunnen als die dat dan in zijn eentje zou moeten. Dus of hij echt heeft geleerd. Dus, dat is een beetje de afweging. Maar tegelijkertijd, als je kan programmeren in zo'n setting, hoe noem je dat, lab setting, dan is de vraag, wat toets je precies? Het kan zijn dat een leerling heel veel tijd kwijt is met debuggen en wat fout heeft, bijvoorbeeld een punt-kommatje wat die over het hoofd ziet, en dat hoort op zich wel bij programmeren, maar de vraag is of je dan, hoe zou ik zeggen, of je dan een objectief beeld krijgt van iemand's programmeren. Dat kost tijd

R: Ja, dat lijkt me dat dat beter bij domein D hoort. Het programmeren zelf. En de vraag gaat over of je een algoritme in een programma kan zetten.

I: Oh ja, want dit is, is dit, dit is eigenlijk domein B? Wacht even. Oh ja?

R: Een algoritme naar een programma? Dat is toch B?

I: Mag ik hem even er bij pakken. Welke is het? 14???

R: Kan voor een gegeven algoritme een programma ontwikkelen in een imperatieve programmeertaal. Oh nee, dat is natuurlijk D. Sorry, dom van me.

I: Ja, het is wel D, toch?

R: Ja, sorry, het is D.

I: Ja, oké. Dus ja, dat is een beetje de afweging, daarom heb ik denk ik neutraal gezegd. Omdat...

R: Allebei heeft zijn voordelen en allebei heeft zijn nadelen.

I: Ja, en op zich, zoals debuggen en dat soort dingen - hoe zeg je dat - bij programmeren komen ook vaardigheden om het zo maar te zeggen, en dat is ook prima, dus dat wil je ook wel toetsen, maar dat [de vaardigheden] zou je meer bij een praktische opdracht willen toetsen.

R: Maar ik hoor van je dat ze dus allebei kunnen?

I: Maar ze hebben allebei hun voor en nadelen inderdaad. Kijk, wat mij betreft wil je gewoon een objectief beeld van programmeerkennis en vaardigheden, en de vraag is, hoe kun je dat objectief meten? Je benaderd het een beetje uit de software, semi-wetenschappelijke, het gaat me niet om het afrekenen van de leerlingen, maar het gaat me om een correct beeld te krijgen van het niveau van de leerling. En als je het op die manier benaderd, wat is dan het beste manier?

R: Je wilt ze een kans geven om te laten zien wat ze kunnen?

I: Ook, ja, zeker.

R: Oké.

[05:02]

R: Ik had nog een vraag, “de leerling kan voor een gegeven situatie, een digitaal product ontwerpen, implementeren en evalueren.” Je had toen aangegeven dat het geschikt is voor CM profiel en NT profiel; je vind het meer geschikt voor CM profiel dan voor NT leerlingen. Het gaat om een digitaal product ontwerpen, implementeren en evalueren.

M: Oh ja, heb ik dat gezegd?

R: Ja, maar je mag ook zeggen, ik heb de vraag toen niet zo goed begrepen.

M: Hm, sorry. Misschien kan het naar aanleiding van een andere vraag komen dat ik daartoe ben gekomen? Dat het meer geschikt is?

R: Ja, welke doelen vind je meer geschikt voor CM dan voor NT profiel?

M: Oh, zo.

[06:10]

M: Nou, ik kan me voorstellen dat voor de maatschappij profielen dat die leerlingen het ook wel interessant vinden in ieder geval, nou, dat in ieder geval voor die leerlingen de techniek minder op de voorgrond staat en moet staan ook. Dat is dan de techniek van, nou, ik vind wel dat ze allemaal moeten leren programmeren hoor, maar dat is dan techniek in meer detail of verdieping voor hun minder van belang is, hoe zeg je dat?

R: Meer het ontwerpen van belang is voor CM leerlingen?

M: Wat zeg je?

R: Dat meer het ontwerpen van belang is voor CM profiel?

M: Ja, ja dus. Het klinkt alsof dat voor de N-profielen daardoor niet van belang is, terwijl juist het ontwerpen voor alle leerlingen, het ontwerpen en de ontwerpcyclus is voor alle leerlingen relevant vind ik.

[07:01]

[interruption]

[08:54]

R: Je gaf ook aan dat dit leerdoel beter met een groepswerk dan individueel te toetsen is, dus” voor een gegeven situatie, een digitaal product ontwerpen, implementeren en evalueren.”

I: Ja, dat lijkt me typisch.. ik weet niet of we de vorige vraag al helemaal

hadden afferond, maar.. Ja het lijkt me typisch, ja, want bij het ontwerpen komt dat je dat in samenwerken doet met andere experts, om zo maar te zeggen, dus ja, in diezin lijkt het me wel wat wat je juist in groepswerk doet. Ja dus, maar nu ik er over nadenk, daar zou je misschien ook wel iets individueel doen, maar dat past niet zo goed bij de praktijk, zou ik zeggen, zoals ontwerp processen meestal worden gedaan.

R: Ja. Je gaf ook aan dat het meer praktische werk is dan voor een geschreven toets.

I: Ja

R: Dus meer praktisch, zou je er eerder een PO aan geven?

I: Ja

R: Omdat dat meer de werkelijkheid nabootst, eigenlijk?

I: Ja, want ik vind ook bij, zoals het soms wordt ingericht dan, dat bij zo'n praktische toets, dat leerlingen het eigenlijk ook maar één keer doen, en dat is wel... nou goed, dat is dan misschien meer het didactisch aspect, dat is dan misschien niet zo zeer het toetsen, maar meer het leren. Want ontwerpen is iets wat je echt moet leren, of er zitten allerlei aspecten aan die je kunt leren. En dat gaat niet van zelf. Ten minste, ik vergelijk dat altijd met voetballen, je kunt kinderen leren voetballen maar dan laat je ze niet alleen wedstrijden spelen, je gaat specifiek dingen oefenen, dus

R: Ja, pionnetjes neerzetten en in kleine stapjes dingen los van elkaar inoefenen

I: Ja, precies. En dat geldt volgens mij voor die vaardigheden ook wel, denk ik, ten minste, het is wel belangrijk om het altijd aan een bepaalde context te koppelen, maar wel dat het kleinere opdrachtjes zijn dat je meerdere malen doorloopt.

R: Ja, precies. Nu je het zegt, op de universiteit heb ik dat ook geleerd met kleine oefeningetjes.

I: Het ontwerpen bedoel je?

R: Ja, maar bij physical computing deed je dat ook. Je maakt voor de Tamagochi een modelletje, en die pas je later nog aan, dan breidt je het weer uit, toch?

[11:19]

I: Ja, en dat is volgens mij bij ontwerpen ook, dat soort vaardigheden zoals bij iteratief werken en ook feedback verzamelen, en ook, dingen zoals divergeren, convergeren, en niet in de eerste-de-besten oplossing in meegaan. Maar dat zijn allemaal dingen, hoe zeg je dat, dingen die je moet oefenen, en dat leer je ook niet zo gauw uit een boekje heb ik het idee.. je hebt over het algemeen een expert, en docent, die dat benadrukt, er vragen over stelt en het steeds herhaalt.

R: En dat convergeren of divergeren waar je het over had. Is dat iets wat je in een toets zou kunnen toetsen?

I: Tja, ik weet niet. Tja, nee, dat denk ik eigenlijk niet. Hoe toets je dat nou? Ja dat is lastig. Maar net zo in een PO. Hoe toets je dat in een PO

dan? Ja, hoe maak je dat zichtbaar? Hoe maak je dat meetbaar?

R: Nou ja, je kunt leerlingen vragen of ze hun beredenering documenteren, in een logboek, je kan zien of ze een keuze hebben gemaakt. En of ze hun keuze, of meerdere keuzes, hebben overwogen van te voren.

I: Ja. Ja, precies. Ja, dan kun je wel, ja dat is waar.

[12:37]

R: Een andere ging om de oplossing van een probleem te vertalen naar een algoritme, dus van het probleem naar een algoritme. Je hebt enerzijds aangegeven dat je het makkelijk vond om dit te toetsen, maar anderzijds dat je het wel moeilijk vond om inzicht te toetsen, dat je niet zo goed wist wat de misconcepten en fouten waren, en dat je het moeilijk vindt om een andere context voor dezelfde vraag te verzinnen. Dus de vraag aan te passen dusdanig dat de moeilijkheidsgraad hetzelfde blijft. Kan je dat toelichten?

I: Ja

R: Is mijn vraag helder?

I: Ja, volgens mij is het idee, je krijgt een beschrijving van een oplossingsrichting in taal, in natuurlijke taal en als leerling moet je daar dan een stroomdiagram bij maken die een implementatie, nee, niet een implementatie, die de algoritme beschrijft. En ik kan me voorstellen dat je dat voor, nou ja ik kan me voorstellen dat dat voor de standaardalgoritmen vrij makkelijk is. Je omschrijft gewoon bubbelsort en zijn maken daar, uhm, schrijft het maar in een stroomdiagram. Ja, dus zeg maar, als het gaat om een aantal standaard uhm algoritmen of standaarddoelen, dan lukt dat denk ik wel, maar uhmm... de reden waarom, wat ik dan moeilijk vind is het zoeken naar algoritmen die een soort van gelijk van niveau zijn maar dan met een andere doel of met een andere context. Ik denk dat die best lastig te vinden zijn en dat het moeilijk is om de moeilijkheidsgraad van de algoritmes hetzelfde te houden.

R: Ja, wij hebben dat ook wel eerder gezien, toen kwam ik met mijn opgave en had jij zo iets, maar het is niet duidelijk wat je wil. Ik kan me dat voorstellen, ik herken dat wel.

I: Ja, ik heb dat nog nooit, hebben we dat nu in deze toets zitten? Niet of nauwelijks eigenlijk he? We hebben daar nu geen opdracht in zitten, van dit is een oplossingsrichting, maar hier nou een stroomdiagram van.

R: Hmm... oh ja.

I: Dat is best een complexe, complexe vraag, ook over hoe je dat beoordeeld, denk ik.

R: omdat er meerdere manieren te beoordelen zijn?

I: Nou Ja, ik kan me voorstellen, als de leerling hem helemaal goed heeft dan, oké is het goed. Maar het zit hem juist in de, als die het voor een deel goed heeft, weet je wel, je kunt dan net van die uitwerkingen hebben waarbij die uh, jaaa... de richting is wel goed maar er is toch iets wat, er zit misschien een gekkigheid in waarvan je denkt van wat heeft de leerling

hier nou van begrepen? En dat is dan ook precies de fout.

[15:50]

R: Ja. En bijbehorende misconcepten? En Fouten? Bij het vertalen van een probleem naar een algoritme?

I: Ja, nou waar ik wel, dat is dan het onderzoek van Ebrahim, waarin wel, hij heeft gekeken naar de, nou misconcepten, nou fouten die leerlingen maken in stroomdiagrammen, dus daar valt wel wat van te zeggen. Maar dat zijn dingen zoals, hoe zeg je dat? Dat leerlingen bijvoorbeeld een if, zo'n keuze met een ruitje opnemen in hun stroomdiagram en dat daar dan drie pijlen uit komen, zeg maar. In plaats van, twee, dat soort dingen.

R: En Ebrahim is je enige bron daarbij?

I: Ja, en dan ga je uit van, zeg maar, kun je een opgave verzinnen, een algoritme verzinnen waarbij, of nee, een oplossingsrichting bedenken waarvan je denkt, nou, dat gaat die misconcepten uitlokken. Tenzij je het anders formuleert. Ik heb hier een oplossingsrichting en hier heb ik 4 mogelijke uitwerkingen, wat is nou de juiste? Op zich, zo'n vraag zat er trouwens wel tussen. Die had [David] gemaakt geloof ik.

[17:01]

I: Maar die had nog niet de uitwerking, hij had vier dezelfde algoritmes. Maar het idee van, welk van deze vier is correct?

R: Dat is wel interessant he?

I: Ja. Dat is ook wel lastig. Wat ik het nadeel van vind is dat het niet zo'n efficiënte vraag is. Dat zijn dan vier stroomdiagrammen die je moet opnemen

R: Ja, dat is voor leerlingen inefficient

I: Ja, kost [hun] heel veel tijd, en het is een multiple choice, dus hoeveel punten kun je daaraan toekennen? Weet je wel, dat komt de betrouwbaarheid niet ten goede.

R: Zou je dat zonder stroomdiagram kunnen doen? Ja, dat is lastig he?

I: Ja, dus, ja, ken jij daar misconcepten over?

R: Nee, ook niet. Ebrahim zou voor mijn daarin voor mij ook een van de weinige bronnen. Ik heb dus niet veel meer dan jij, nee.

I: Tenzij je, als je het in pseudocode maakt, dan kom je op, wat je net ook al zei, de if-then-elsjes, en de loopjes, dan zit je meer richting de programmeermisconcepten en daar is natuurlijk al heel veel over bekend, maar ja, daar ga je dan al vanuit. Ja hoe toets je dat, het beschrijven van een algoritme in pseudocode, daar is al wel van alles over bekend, maar om die misconcepten uit te lokken zou ik maar zeggen, en dus expliciet te testen, dat lijkt me nog wel heel moeilijk ja.

[18:24]

R: Ja. Behalve inderdaad natuurlijk als je een voorbeeld uitwerking geeft en zegt "Geef aan welke fouten hierin zijn".

I: Ja, precies.

R: */not clear enough to hear clearly or understand/*

I: Dat je niet in een toets hoeft te vragen, “dit is een oplossingsrichting, maak een stroomdiagram”, maar dat je meer zegt “ok, hier is een eerste aanzet tot, hier ontbreken nog wat delen, en vul die verder aan. Ik weet ook niet of je veel breder moet gaan dan gewoon wat meer de standaard doelen, zoals zoeken en sorteren, graven... of dat je niet een hele nieuwe context bedenkt en een heel nieuw doel, weet je wel.

R: Omdat het voor de leerlingen veel tijd kost om erin te komen?

I: Ja, ik zit nou even hard op te denken hoor, maar is dat nou, bijvoorbeeld, kun je ook zeggen, nou, we gaan, iets van, dat je een string kan omdraaien, of een geheel andere... Ja, dat kan wel ja. Nou, ik zit even hardop te denken. Zijn dat nou de oplossingsrichting waar je aan moet denken? Nou, een toepassingsrichting of een doel wat de leerlingen eigenlijk nog niet hebben gehad, een T2 vraag

R: Ja I: Ja, ik denk wel, het zou wel mooi zijn. Dat is ook wel, ja. Wat ik moeilijk vind is ook, dat kun je natuurlijk oefenen, maar is het ook niet zo dat elke vraag toch weer zijn eigen aspecten heeft? Dus dan wordt he ook alweer gauw een beetje inzicht.

R: Ja, dat kan

I: Ja zeker. Nou, god.. wel interessant

R: Ja

[20:04]

R: Ok, Een laatste [vraag] over wat je had ingevuld. “De student kan een standaardalgoritme herkennen en gebruiken”. Het gaat dus om standaardalgoritmen, en hier staat dat je het makkelijk vind om te toetsen, maar eigenlijk weer hetzelfde, je vond het moeilijk om inzicht te toetsen, en misconcepten, enzo, het ging om de standaardalgoritmen, maar volgens mij heb je daar net ook al een beetje antwoord op gegeven.

I: Standaardalgoritmen herkennen en gebruiken, hè?

R: Ja, die vond je makkelijk om te toetsen, maar moeilijk om inzicht te toetsen

I: Ja, het is makkelijk om te toetsen, ja, want, ja, die standaardalgoritmen die kun je zo, ja, die heb je gewoon en dan, en dan zeg je...

R: Is het goed? Is het niet goed?

I: Ja. Of je kunt ze vragen beschrijf of maak een stroomdiagram voor een standaardalgoritme, of ook, nou ja, of ook dat voorbeeld wat jij volgens mij ook had, van, dat is ook wel een mooie, van je hebt een algoritme en waar herken je nu het standaardalgoritme? Omcirkel het, wat was er nog meer? Wel een efficiënte vraag om het zo maar te zeggen.

R: Ja. En andere contexten hiervoor vinden vond je lastig. Is dat van, er zijn maar zo veel standaardalgoritmen? Zou je daar geen context aan willen hangen? Ik bedoel, dat je een ander opgave verzint die net zo moeilijk is.

I: Ja, dan denk ik meer, stel, bij zoeken, dat je zegt een keer zoek je in een

database, of hoe zeg je nou, niet een database, maar, dan gaat het over die artiesten he, zoals [Michael] had, dan gaat het over een bibliotheek, weet ik veel, ja, dat is natuurlijk makkelijk, maar niet in wezenlijk andere [context].. nou, zoals die opgave die we vandaag bespraken, zo'n situatie van, de leerling wil, de docent wil de leerlingen tellen [refers to an assessment question made by Charles] of kijken of ze allemaal aanwezig zijn, of dat idee, maar dan loopt het gauw spaak. Dat zag je bij deze opgave ook. Het klopt niet helemaal. En dan is het een geforceerde context.

R: Ja.

I: Ik vind dat sowieso, met dat soort contexten, we hadden het er over, toen, die algoritmiek module, en daar wordt heel erg de relatie met de leefwereld van de leerling gelegd, maar bijvoorbeeld, hoe kun je nou de kortste route van huis naar school vinden, maar dat is niet helemaal de context vind ik. Ik bedoel, het is wel een leuk, het kan wel een speelse manier zijn om dat te bespreken, maar, het is voor een leerling toch niet relevant hoe je nou de kortste weg naar school vind, ik bedoel, die vind die wel. Om het zo maar te zeggen, snap je?

R: Ja

I: Dus, het is niet echt de toepassing, het is niet ECHT een context.

R: Ja, dus het is daarom ook makkelijker om het juist contextvrij te laten, gewoon?

I: Ja. Maar dat is ook wel jammer, want ik ben er wel voor, want die toets die wij nu maken, ik moet denken aan Felienne altijd, met die, ik heb toen met haar gesproken over de Beverwedstrijd. Ja, nou goed, mede naar aanleiding daarvan, dat idee van dat je, zo'n beverwedstrijd suggereert een beetje van "dat kun je nu eenmaal, of je kunt het niet", dat is een soort van inzicht. Terwijl we willen juist laten zien, informatica is ook een vak dat kan je leren. Programmeren kun je ook leren. En daarnaast, je zie heel veel informatica, dat de programmeeropgaven, die contextloos zijn, inderdaad, maar voor heel veel leerlingen is dat niet interessant... je wil, leerlingen willen juist zien wat je er mee kunt. En dan wordt het voor hun relevanter en interessanter. Dan maak je ze meer enthousiast voor het vak als je dat doet.

[23:36]

I: Ja, en zo'n toets, dat is daar ook onderdeel van. Weet je wel. Want als je zegt, dit is de toets, en het is allemaal contextvrij, dan geef je daarmee een beeld van, ja informatica is gewoon, is voor der nerds en de wiskundigen zeg maar. Je kunt daar geen *** mee.

R: Ja, het zijn dan gewoon modellen, ja.

I: Ja, terwijl, ja.

R: /sound drops/

[24:04]

R: Oh, je gaf hetzelfde aan bij grammatica's. Geldt daar hetzelfde voor dat je, op zich, dat je via een contextloze droge opgave kun je makkelijk

toetsen, maar als je daar verder in wil gaan...

I: Ja

R: De inzicht, en het aanpassen van zo'n toetsvraag, dat vind je weer lastiger.

I: Ja,

R: Dat is hetzelfde soort verhaal als bij de vorige?

I: Ja precies, dat denk ik wel.

R: Ja, dat idee had ik ook wel.

[23:37]

[End of the MemberChecking part of the interview with William]