Constructive Alignment and the SOLO Taxonomy: A Comparative Study of University Competences in Computer Science vs. Mathematics

Brabrand, Claus; Søndergaard, Bettina Dahl

Published in:
Conferences in Research and Practice in Information Technology

Publication date:
2008

Document Version
Early version, also known as pre-print

Constructive Alignment and the SOLO Taxonomy: A Comparative Study of University Competences in Computer Science vs. Mathematics

Claus Brabrand
IT University of Copenhagen, Denmark
brabrand@itu.dk

Bettina Dahl[1]
University of Aarhus, Denmark
bdahls@si.au.dk

Abstract

In 2007, all Danish university syllabi were reformulated to explicitly state course objectives to comply with a new Danish national grading scale, which stipulated that grades were to be given based on how well students met explicit course objectives. This paper analyzes 550 syllabi from the science faculties at University of Aarhus, Denmark (AU) and the University of Southern Denmark (SDU) that had been rewritten to explicitly incorporate course objectives, interpreted as intended learning outcomes (ILOs), using the principles of Constructive Alignment and the SOLO Taxonomy. In this paper we explain and discuss these principles, give examples of how the new syllabi were constructed, and describe the process by which they were formed. We also explain and discuss the results of a comparative study comparing the competences of Computer Science with those of Mathematics (and classical Natural Sciences for a point of reference). In this study, we focus on what specific competences the respective departments primarily use.

Keywords: Constructive Alignment; SOLO Taxonomy; Competences; Intended Learning Outcomes (ILOs); Computer Science; Mathematics; Natural Science.

1 Introduction

This paper is an analysis of a data set consisting of 632 course syllabi from the science faculties at University of Aarhus, Denmark (AU) and the University of Southern Denmark (SDU). Both faculties have been through a process of formulating intended learning outcomes (ILOs) to existing course syllabi. The reason for this process was the adoption of a new nationwide Danish grading scale, which stipulates that grades are to be given based on how well students meet explicit course objectives. ILOs were thus formulated for all courses in the two faculties for one academic year. However, this paper will comparatively investigate only those of Computer Science, Mathematics, and classical Natural Sciences (here restricted to Physics, Chemistry, Biology, and Molecular Biology), using the SOLO Taxonomy as a tool for analyzing ILO competences. In total, this gives us a data set of 550 courses. All ILOs have been formulated according to the principles of Constructive Alignment and using the SOLO Taxonomy. The academic staff at AU received a course on these principles by a group of five people, appointed by the dean, of which both authors were members and which was chaired by Brabrand. SDU had a very similar process for which Brabrand was a consultant. The paper is both a summary of the keynote talk given by Brabrand at Koli Calling 2007 and a further study of the data. The first part of the paper introduces the Theory of Constructive Alignment and the SOLO Taxonomy. The second part explains our comparative study of competences.

2 Constructive Alignment and the SOLO Taxonomy

2.1 The Theory of Constructive Alignment

The Theory of Constructive Alignment (Biggs 2003) is a theory of teaching and learning developed by John Biggs. It is a *systemic theory* in the sense that the entire teaching context is perceived as a 'system' for which we need to understand the individual parts and how they interact in order to understand and make predictions about the entire system. The theory is also based on the principles of *constructivism*, that knowledge is personal and that meaning is actively constructed by the learners themselves through active engagement with the subject matter. This perspective is in sharp contrast to the (once commonly held) idea that knowledge is 'transmitted' *from an active teacher to a passive student*. Finally, it is a *constructive theory* in the sense that it embodies constructive advice for what teachers ought to do in order to make sure their students learn what they intend. The 'solution' to this important challenge of teaching is for the teacher to *constructively align* courses 'ahead of time' (see Figure 1).

Before we can understand and appreciate why this is an

Definition: A course is said to be [*constructively*] *aligned* when:
- intended learning outcomes (ILOs) are explicitly formulated as operational competences; - the ILO competences are explicitly communicated to the students (early in the course); - the exams measure precisely the ILO competences; *and* - the teaching/learning activities (TLAs) match the ILO competences.

Figure 1: Definition of 'constructively aligned course'

[1] Full name: Bettina Dahl Søndergaard (Soendergaard).

Goal:	*The goal of the course is to:* - **introduce** students to general design techniques for the construction of effective algorithmic solutions to combinatoric problems; *and* - **familiarize** the student with effective solutions to important graph and string problems.
Content:	- algorithmic paradigms: *'divide and conquer', 'dynamic programming', 'greedy algorithms'*; - graph algorithms: *'traversal strategies', 'connectivity', 'topological sorting', 'spanning trees', 'shortest path', 'transitive closure'*; *and* - text processing: *'pattern recognition'*.

Figure 2: Content description for 'Algorithms & Datastructures II' (Computer Science, AU)

interesting thing to do, we first (in the spirit of systemic theories) need to address three dependent factors: *teacher's intention*, *student's activity*, and *exam's assessment*. Please note that throughout the paper we will use the word 'exam' in a broad sense for any assessment activity during and/or after the course that counts towards a student's final grade; i.e. not just one 'all-or-nothing test' at the end of a course.

Teacher's intention

The ultimate goal of any teaching situation is that the students learn whatever it is they are supposed to learn. However, 'learn' is an inherently vague concept and may ambiguously refer to both 'learn (about)' and 'learn (to do)'. To 'learn *about* programming' is clearly very different from to 'learn to *do* programming' (i.e., 'to learn to program'). Traditionally, many teachers have created course descriptions formulated and communicated to students in terms of 'content' to be learned (about). Figure 2 is an example of such a course description for the undergraduate Computer Science course 'Algorithms & Datastructures II' from the University of Aarhus before the faculty-wide rewriting of all syllabi to include ILOs using the principles of constructive alignment and the SOLO taxonomy.

Teachers and examiners, being part of the same research-based teaching traditions, will most likely know immediately from the description in Figure 2 what it is the students are expected to be able to do when they are assessing. They will interpret (and most likely agree) that what is really meant by *'algorithmic paradigms'* is that a student is expected to (be able to), for example, '**construct** algorithms' and '**analyze** algorithms' using standard algorithmic paradigms. However, this is tacit knowledge, usually not known by the students. Students, not part of the same research-based educational tradition, might interpret *'algorithmic paradigms'* in an altogether different sense; for instance, that they should be able to ***name*** standard algorithmic paradigms and ***recite*** running-times of textbook algorithms from each of the algorithmic paradigms, on command. Clearly, to 'construct and analyze' is qualitatively different from, and somehow at an entirely different level from, to 'name and recite'. The nature of this difference is precisely what is accounted for by the SOLO Taxonomy, which we will explain in some detail below.

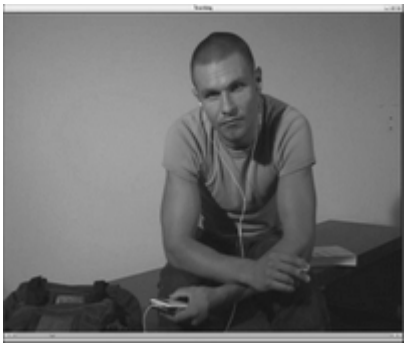
Furthermore, without quantum leap advances in brain scanning technology, we simply cannot *measure* 'understanding'; how much 'knowledge' a student has, how well a student has been 'introduced' to, or how 'familiar' a student is with, a given concept such as 'algorithmic paradigms'. These are internal cognitive structures inside the brain, biochemical high-level structuring of which we know very little and on which we can currently only speculate. What we can do, however, is to have a student *do* something, and then measure the *product* and/or *process*. The keyword here is 'operationality' which is the other aspect captured by the SOLO Taxonomy in that the competences it taxonomizes are operational and measurable. 'Understanding', 'knowledge', and 'familiarity' are inherently non-operational and non-measurable goals. Before we turn to how the above course description would have been authored if adhering to the principles of constructive alignment, we need to consider the students; in particular, a student's activity (before, during, and after teaching).

Student's activity

The Susan and Robert dichotomy, conceived by John Biggs (Biggs 2003), fits students to models (also known as personas) according to their motivation for studying at university. The personas should be thought of as prototypical student strategies rather than actual individuals. Figure 3 depicts Susan and Robert as *personified* in the 19-minute award-winning short film about constructive alignment, entitled 'Teaching Teaching & Understanding Understanding' (Brabrand & Andersen 2006).

Susan (Figure 3a) is intrinsically motivated and at university to learn: "Susan likes to get to the bottom of things; to reach understanding. She often reflects on

a) *Susan*
(intrinsically motivated to learn).



b) *Robert*
(extrinsically motivated to learn).

Figure 3: The Susan and Robert dichotomy introduced by John Biggs

What is the asymptotic complexity of 'topological sorting' on a directed graph $G = (V, E)$*?*		
☐ a)	$\Theta(\log(\lvert V\rvert+\lvert E\rvert))$	i.e., *"logarithmic time"* in the size of the input
☐ b)	$\Theta(\lvert V\rvert+\lvert E\rvert)$	i.e., *"linear time"* in the size of the input
☐ c)	$\Theta((\lvert V\rvert+\lvert E\rvert)*\log(\lvert V\rvert+\lvert E\rvert))$	i.e., *"n-log-n time"* in the size of the input
☐ d)	$\Theta((\lvert V\rvert+\lvert E\rvert)^2)$	i.e., *"quadratic time"* in the size of the input

Figure 4: Hypothetical algorithmic MCQ leading to 'content memorization'

possibilities, implications, applications, and consequences of what she is learning. Susan is characterized by a preference for deep learning. She spontaneously uses higher cognitive processes. Faced with a curriculum, she basically teaches herself. In fact, we almost cannot prevent her from learning" (Brabrand & Andersen 2006).

Robert (Figure 3b) is extrinsically motivated and not at university to learn: "In fact, Robert doesn't really care about the learning in itself. His goal at university is different; his goal is […] to pass exams, get a degree, and get a (decent) job. Robert is characterized by a preference for surface learning. He will only use higher cognitive processes if he really really really has to. He will cut any corner in achieving his goal with minimum effort" (Brabrand & Andersen 2006). Robert will stick with lower-level activities such as identification and memorization as long as they suffice.

As teachers, we do not need to worry about Susan; she will do fine. But what about Robert; what can we do to have him start acting more like Susan? Before we show how constructive alignment can be used to do just that, we need to look at the exam's assessment.

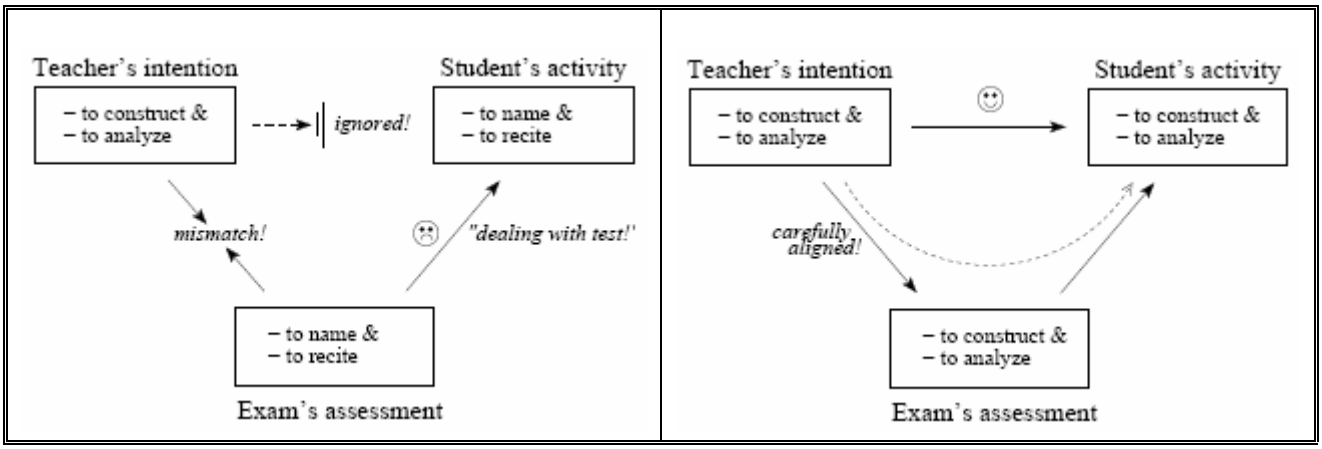
Exam's assessment

For many teachers and students, exams are a 'necessary evil'. However, the exam is perhaps the single most powerful pedagogical motivational tool available to teachers (and students) in that the exam has a constitutional effect on learning; the so-called 'backwash effect'. The exam has ramifications on how the students approach learning; in particular, on how willing they are to engage in learning activities. This includes any learning activity, whether in formally situated learning contexts planed by a teacher or autonomous informal learning spontaneously initiated by the student. "To the teacher, assessment is at the end of the teaching-learning sequence of events, but to the student it is at the beginning" (Biggs 2003, p. 141). As an illustration of this, let us consider the (hypothetical) algorithmic multiple-choice question shown in Figure 4, which seems perfectly innocent.

When featured on an exam, however, the question will in essence reward students for using 'content memorization' strategies. The Roberts will do what is appropriately known as 'dealing with the test'; i.e., they will disregard the teacher's intentions of deep learning, and instead direct their learning towards strategic memorization of running times of textbook algorithms. They will thus stick with their low-level surface learning techniques (e.g., identification and memorization). In essence, we have what is known as an *unaligned* course.

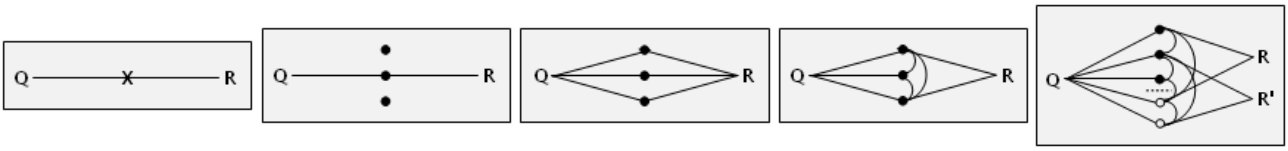
Figure 5 illustrates the essential difference between an *unaligned* and an *aligned course*. (The figure abstracts away the issues pertaining to the teaching/learning activities, which we will address later.) In an *unaligned course*, we have a mismatch between the teacher's intention and the exam's assessment. The teacher intends for students to learn to 'construct and analyze', but the exam measures the competences to 'name and recite'. As outlined above, Robert will focus only on the skills required for the test, and disregard the teacher's intentions. In such courses the Roberts will seem disinclined to participate and engage in higher-level learning activities.

The solution to this problem proposed by John Biggs with his Theory of Constructive Alignment is to *constructively align* courses. The teacher is to formulate ILOs as operational competences from the SOLO Taxonomy (which we will explain shortly), communicate these



a) An *unaligned* course b) An *aligned* course

Figure 5: An unaligned vs. aligned course (reproduced from Brabrand, 2007)



a) SOLO 1 b) SOLO 2 c) SOLO 3 d) SOLO 4 e) SOLO 5

Figure 6: Visualization of the SOLO-levels 1-5. (Based on Biggs and Collis (1982, pp. 24-25).)

explicitly to the students early in the course, and meticulously design the exam such that it measures precisely those ILOs (and convince the students that this is indeed the case). The result is a 'commuting diagram' (Figure 5b) and teaching system where Robert's desire to pass the course invariably leads him through learning the ILO chosen by the teacher.

Teaching/Learning Activities (TLAs)
Now Robert has the necessary *incentive* to learn, but we also need to provide appropriate *support* for him to learn effectively. This is where the teaching/learning activities (TLAs) fit in constructive alignment. The challenge is to choose TLAs that are likely to bring about acquisition of the ILO competences. Framing teaching activities as 'training towards the exam' will help engage and motivate Robert to participate actively.

2.2 The SOLO Taxonomy

The SOLO Taxonomy (short for *Structure of the Observed Learning Outcome*) originates from the study of student learning *outcomes* in university teaching carried out by John Biggs and Kevin F. Collis in the early 1980s. The taxonomy distinguishes five different levels according to the cognitive processes required by students in order to obtain them. "SOLO describes a hierarchy where each partial construction [level] becomes a foundation on which further learning is built" (Biggs 2003, p. 41). As described above, the taxonomy can be appropriately used to define ILOs in implementing Constructive Alignment. It is constructed particularly for research-based university teaching and converges on production of new knowledge (at its fifth and highest level) which is also the purpose and product of research itself. The five levels are visualized in Figure 6 and explained in the following, in increasing order of structural complexity (Biggs & Collis 1982, pp. 17-31; Biggs 2003, pp. 34-53). The figure depicts which elements are involved when a student at a given SOLO-level, given a *problem/question/cue* "Q", produces an *outcome/response* "R" using the kinds of *data* either provided or not. The symbol "x" stands for irrelevant data; "●" stands for known related data that has been given to the student; and "○" stands for hypothetical related data that has not been given to the student.

SOLO 1: The Pre-Structural Level
Figure 6a depicts the outcome formation at SOLO level 1; a student is given a question or cue "Q" and uses irrelevant data "x" in producing a response, "R". At this level, a student does not have any kind of understanding but uses irrelevant information and/or misses the point altogether. Scattered pieces of information (i.e., "x") may have been acquired, but at this level they will be unorganized, unstructured, and essentially void of actual content or relation to a relevant issue or problem, "Q". Of course, students such as Robert may attempt to camouflage their lack of knowledge, by using 'tautological responses' i.e., reusing and rearranging *cues* of a question to produce an answer with essentially the same information that was embedded in the question, giving the illusion of understanding.

SOLO 2: The Uni-Structural Level
Level two is depicted in Figure 6b; a student is now capable of dealing with *one* relevant known aspect or issue "●" and use it in producing a valid but simple response "R". Thus, from level one to two, we see improvements as the student becomes able to discern relevant issues and deal with *one* of these in relation to a problem "Q". At this level, a student is capable of making obvious relevant connections and can, for instance, use correct terminology, remember things, recite, carry out simple instructions, identify, name, count, paraphrase on a sentence level, etc.

SOLO 3: The Multi-Structural Level
From level two to three we see *quantitative* improvements as the student becomes able to deal with a multiplicity of relevant known issues "●". As illustrated in Figure 6c, a student is now capable of dealing with several aspects, but these are considered independently and not in connection to one another; e.g., how they may interrelate to form a whole. Metaphorically speaking, the student sees the many trees, but not the wood. He is able to enumerate, describe, classify, combine, apply methods, structure, execute procedures, etc.

SOLO 4: The Relational Level
At level four (Figure 6d), we begin to see *qualitative* improvements as the details integrate to form a structure. A student may now perceive relations between several aspects and how they might fit together to form a whole and structured response "R". The student now sees how the many trees together form a wood. A student may thus have the competence to compare, relate, analyze, apply theory, explain in terms of cause and effect, etc.

SOLO 5: The Extended Abstract Level
From level four to five, we see further *qualitative* improvements as the structure is generalized and the student becomes capable of dealing with hypothetical information that was not given, "○" (Figure 6e). At this fifth and highest level, a student may now perceive the knowledge structure from many different perspectives and produce multiple responses ("R" and "R'"), depending on the perspective and hypothetical information included. Here, a student may have the competence to generalize, hypothesize, criticize, theorize, or transfer a theory to a new domain, etc.

- Quantitative -		- Qualitative -	
SOLO 2 *uni-structural:*	SOLO 3 *multi-structural:*	SOLO 4 *relational:*	SOLO 5 *extended abstract:*
- paraphrase - define - identify - count - name - recite - follow (simple) instructions - …	- combine - classify - structure - describe - enumerate - list - do algorithm - apply method - …	- analyze - compare - contrast - integrate - relate - explain causes - apply theory (to its domain) - …	- theorize - generalize - hypothesize - predict - judge - reflect - transfer theory (to new domain) - …
a) SOLO 2 competences	b) SOLO 3 competences	c) SOLO 4 competences	d) SOLO 5 competences

Figure 7: Prototypical verbs according to the SOLO Taxonomy; based on Biggs (2003, p. 48)

The terms *'surface understanding'* and *'deep understanding'* (also known as 'surface learning' and 'deep learning') are often used and, in fact, easy to define in conjunction with the SOLO Taxonomy. Surface learning implies that the student is confined to action at the lower SOLO levels (2-3); whereas deep learning implies that the student can act at any SOLO level (2-5), including the higher levels (4-5). Hence a student producing a high-level response (at SOLO 4-5) is thus often deemed to have a deep understanding of the matter. On the other hand, a student producing a lower level response (at SOLO 2-3) does not necessarily have a surface understanding. Finally, levels 2 and 3 are sometimes referred to as *quantitative* levels, and levels 4 and 5 as *qualitative*. Figure 7 lists prototypical competences from the SOLO Taxonomy, many of which were mentioned above.

2.3 Alignment implementation process

Figure 8 below shows the alignment implementation process as it was recommended to the teachers at the faculty courses at AU and SDU. It is explained in the following.

1. Determine overall goals
The first step in designing an aligned course is to consider what are the overall things that the students are to get out of attending the course. Here, it is important to think in terms of competences *in addition to* content (the latter being what teachers are used to); i.e., what is it the students are supposed to learn to be able to *do* with the content once the course is over?

2. Operationalize goals as intended learning outcomes (ILOs)
The next step is to operationalize these goals and express them as ILOs in terms of the SOLO Taxonomy. This step can initially be a challenge for teachers.

3. Choose forms of assessment (relative to ILOs)
Once the ILOs are chosen, the teacher needs to provide adequate *incentive* in order for students to learn the relevant competences (ILOs). Here, the teacher chooses / designs one or more forms of exams to cover all ILOs so that the competences are measured as precisely as possible. Certain combinations of ILOs and forms of exam are obvious mismatches (e.g., have an MCQ test to assess the competence 'to explain'), but is in most cases the teacher needs to carefully judge what exam form best fits the ILOs. This can also sometimes be a challenge due to practical issues and external constraints at the university, e.g., space, time, and/or economic issues.

4. Choose forms of teaching (relative to ILOs)
With the ILOs in place, the teacher needs to provide adequate *support* for students to learn the relevant competences (ILOs). Here, the teacher may choose several teaching activities to cover all ILOs. Again, certain combinations of ILOs and teaching activities obviously do not go together; e.g., 'lecture on (about) programming' vs. ILOs stating 'learning to program' (as in 'learning to do programming'). Again, this calls for careful judgements on behalf of the teacher. Steps 3 and 4 could be carried out in either order, or in parallel, but

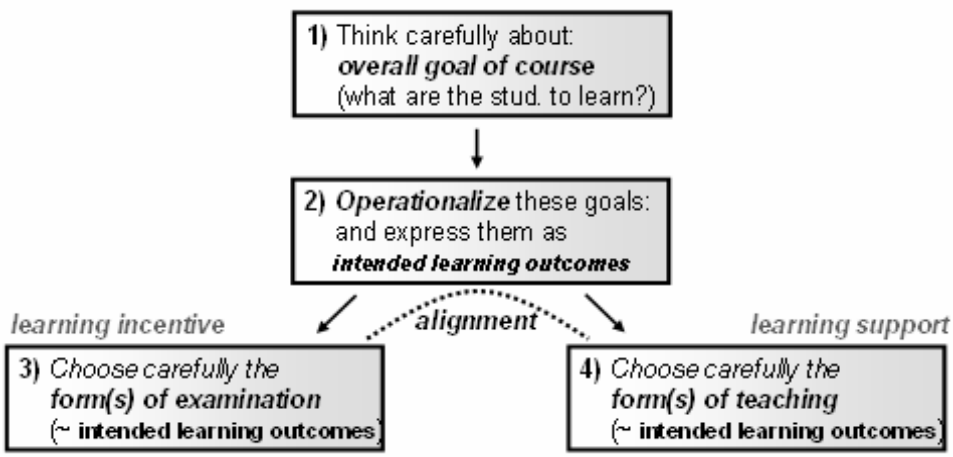


Figure 8: Alignment implementation process

At the end of the course, the student is expected to be able to…:
- **construct** (SOLO 4) and **analyze** (SOLO 4) algorithms using standard algorithm paradigms;
- **identify** (SOLO 2) and **formulate** (SOLO 3) algorithmic problems as graph and string problems;
- **identify** (SOLO 2) and **compare** (SOLO 4) graph and string algorithms for solving algorithmic problems;
- **construct** (SOLO 4) algorithms for simple graph and string problems.

Figure 9: Example ILOs from 'Algorithms & Datastructures II' (undergrad Computer Science)

Competence	Level	Frequency
- identify	SOLO 2	(2x)
- formulate	SOLO 3	(1x)
- construct	SOLO 4	(2x)
- analyze	SOLO 4	(1x)
- compare	SOLO 4	(1x)

Figure 11: Frequency counts for 'Algorithms & Datastructures II'

aligning the teaching/learning activities towards the exam, 'training towards the exam', is likely to make the courses seem relevant to all students, including Robert; hence, it is often instructive to settle the form of exam prior to designing relevant TLAs. Alignment is then a product of how well steps 2, 3, and 4 correspond to one another. For more information on how to carry out this process and implement alignment for a specific science course, we refer to Brabrand (2007).

3 A Comparative Study of Competences in Computer Science vs. Mathematics

We now turn to a presentation and discussion of the specific competences used in Computer Science and Mathematics. Our data material for comparing these two subjects in depth according to their competences consists of 550 course syllabi. For an analysis of i) competence progression, ii) overall differences between science subjects, and iii) differences between similar departments at different universities, using the full 632 course data set, we refer to Brabrand & Dahl (2008).

Each of the 550 courses has a number of goals, each with a number of ILO competences. Figure 9 illustrates the competence description for the undergraduate Computer Science course 'Algorithms & Datastructures II' at AU.

The syllabi were created using the alignment implementation process (Figure 8). We have chosen to focus on the *formulated* ILOs because these have a strong impact on the grading since grades are to be given based on how well students meet the ILOs. The formulated outcomes are not necessarily always the same as the *formal*, *realized* (*operationalized*), or *learnt* outcomes (Bauersfeld 1979; Goodlad 1986) but owing to the constitutional effect of examination on learning ('teaching to the test'), the ILOs have an impact on the learning and in the event of students complaining about grades, it is legally the formulated outcomes that matter. Thus, teachers are forced to take the formulated outcomes very seriously.

There are three ways in which we analyzed the data:

1. SOLO average
We calculated a *'SOLO average'* using a 'double-weight averaging scheme' in which each of the ILOs weigh the same and each of the verb competences within an ILO also weigh the same:

$$[\ (4+4)/2 + (2+3)/2 + (2+4)/2 + 4.0\ ] / 4 = \underline{3.38}$$

One might question whether in practice these ILOs do weigh the same. However, under the guidance of the group appointed by the dean that included the two authors, each department has formulated ILOs from a number of 'standard good examples' provided in advance. Hence if there is a variation, it is consistent throughout the syllabi. The idea of a SOLO average also rests on the assumptions that the 'competence distance' from, say, SOLO 2 to 3, is the same as between SOLO 3 and 4 etc. Such an approach of quantifying qualitative data is in fact often seen in educational research, for instance in the use of Likert scales that quantify degrees of agreement or disagreement using numbers, usually 1-5 (Oppenheim 1992; Robson 2002). Oliver et al. (2004) have carried out an analysis similar to ours using the six levels of the Bloom Taxonomy, but for only a handful of courses.

2. SOLO distribution The SOLO average will be complemented with comparisons of the relative distributions of SOLO levels for both individual and collective courses. One such example is seen in Figure 10, which shows the SOLO distribution of the above-mentioned course.

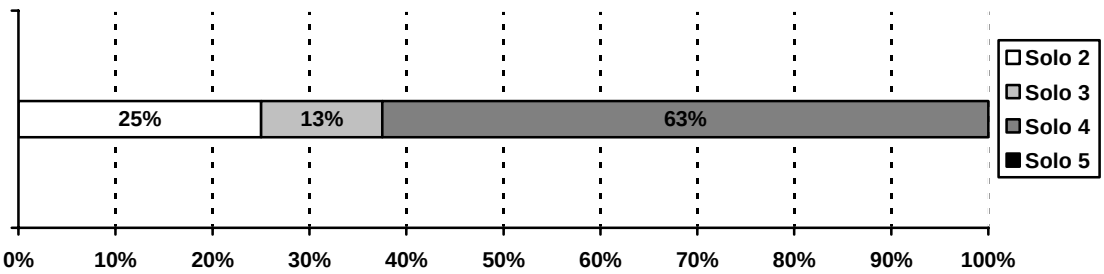


Figure 10: Relative distribution of SOLO levels for 'Algorithms & Datastructures II'

- Quantitative -				- Qualitative -			
SOLO 2 *uni-structural:*		SOLO 3 *multi-structural:*		SOLO 4 *relational:*		SOLO 5 *extended abstract:*	
- identify	(168x)	- describe	(677x)	- explain	(382x)	- discuss	(212x)
- calculate	(80x)	- account for	(593x)	- analyze	(281x)	- assess	(125x)
- reproduce	(64x)	- apply method	(485x)	- compare	(103x)	- evaluate	(58x)
- arrange	(56x)	- execute proc.	(154x)	- argue	(75x)	- interpret	(51x)
- decide	(32x)	- formulate	(85x)	- relate	(70x)	- reflect	(39x)
- define	(25x)	- use method	(75x)	- implement	(55x)	- perspectivate	(37x)
- recognize	(20x)	- solve	(68x)	- plan	(44x)	- predict	(28x)
		- conduct	(61x)	- summarize	(35x)		
		- prove	(57x)	- construct	(31x)		
		- classify	(36x)	- design	(21x)		
		- complete	(34x)				
		- combine	(25x)				

Figure 12: Examples of verbs at SOLO levels
(verbs occurring at least 20 times in course descriptions at AU & SDU)

3. SOLO frequencies

A third way is to see the frequencies, either in raw numbers or in percentages of how often various competences within each SOLO level occur. Figure 11 shows the list for the same example course.

Looking at the whole dataset, the list becomes as shown in Figure 12, which lists the verbs / SOLO competences occurring at least 20 times in the syllabi of all 632 courses. We chose to cut at 20 to give an overview of the main competences used more than just a few times.

The validity of our analysis depends first on SOLO being an appropriate description of competences and second on our SOLO classification being appropriate. In relation to the former, we built our work on the SOLO model, which is the result of extensive research done by Biggs and Collis (1982, 2003); with regard to the latter, through an iterative process of three stages we consulted several other educational researchers (including Biggs) to get feedback on the classification, which resulted in Figure 12. The three approaches to data analysis complement each other and are mixed to illustrate key, but different, characteristics of the data.

3.1 Differences in SOLO competences between Computer Science and Mathematics

We compare the departments of Computer Science, Natural Science, and Mathematics. Our data in Natural Science consist of a compilation of the data of the departments of Physics, Biology, Chemistry, and Molecular Biology. We excluded Geology since it is a department only at AU. These four departments seem to be quite similar with respect to their average SOLO levels (Brabrand & Dahl, 2008). Even though we basically want to compare Mathematics and Computer Science, we felt we needed a third partner in the comparison to shed light on some issues related to the two subjects, hence the choice of Natural Science as a comparison partner. In the following we use the terminology of calling the combined data set of the four natural science departments 'the Natural Science department' even though it is in fact several departments.

Looking at the SOLO averages at the two universities (Figure 13) we see that although there are some differences between the departments, they appear in the same order of SOLO hierarchy at each institution, and so does the average across both institutions. Furthermore the difference between 'sister-departments' at different universities (from 0.1 to 0.3) is generally smaller than the intra-university difference between departments (from 0.1 to 0.6). This indicates that it does make sense to pool the data from, for example, the two mathematics departments, and expect some meaningful data and conclusions.

Subject	AU	SDU	avg.	diff.
Computer Science	3.7	3.4	3.6	0.3
Natural Sciences	3.4	3.3	3.4	0.1
Mathematics	3.1	2.8	3.0	0.2
Intra-university diff.	0.3-0.6	0.1-0.6	0.2-0.6	

Figure 13: SOLO averages by department and university

Turning to the relative distribution of SOLO competences (Figure 14), we get a more detailed view of the

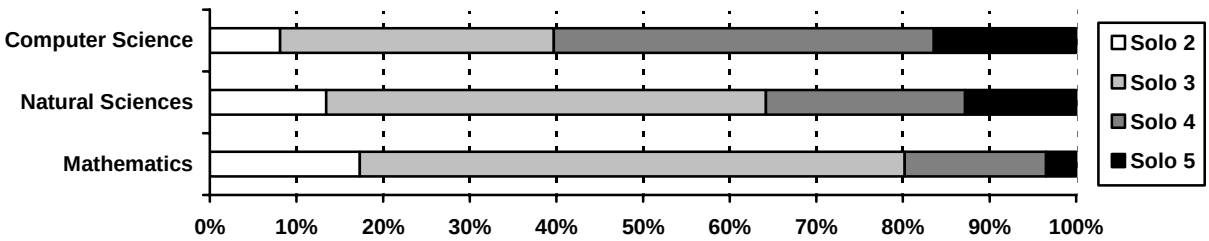


Figure 14: Distribution of competences by SOLO levels for the different departments at AU

TOP 10 COMPETENCES									
	Computer Science			**Natural Sciences**			**Mathematics**		
	Competence	*SOLO*	*Freq.*	*Competence*	*SOLO*	*Freq.*	*Competence*	*SOLO*	*Freq.*
1.	describe	3	13 %	describe	3	15 %	apply method	3	20 %
2.	explain	4	10 %	account for	3	13 %	reproduce	2	13 %
3.	apply method	3	9 %	apply method	3	9 %	solve	3	8 %
4.	implement	4	7 %	explain	4	8 %	formulate	3	8 %
5.	analyze	4	6 %	analyze	4	5 %	prove	3	7 %
6.	discuss	5	5 %	discuss	5	4 %	argue	4	5 %
7.	design	4	4 %	execute proc.	3	3 %	compare	4	5 %
8.	compare	4	3 %	identify	2	3 %	account for	3	5 %
9.	evaluate	5	3 %	assess	5	3 %	illustrate	3	3 %
10.	identify	2	3 %	formulate	3	2 %	combine	4	3 %
"Σ"			62 %			67 %			75 %

Figure 15: Top 10 competences for Computer Science vs. Mathematics vs. Natural Sciences

distribution of SOLO competences and what is behind the differences between the SOLO averages of the three departments. We see that the relative number of SOLO 2 and 3 competences *increases* and that the number of SOLO 4 and 5 competences *decreases* as we move down the departments. In other words, Computer Science uses more higher-level competences than Natural Science, which in turn uses more than Mathematics. Also, the majority of Computer Science competencies are qualitative competences (60% are at SOLO levels 4 and 5), while Mathematics and Natural Science both seem to use mainly quantitative competences (at SOLO levels 2 and 3).

To go even deeper into the differences between the departments, we investigated whether or not there is also a difference in the specific competences employed at the different departments. Figure 15 lists the 10 most-used competences for each of the three departments, an indication of the SOLO level to which they belong, and their frequency.

In Figure 15, we see that for Mathematics, the top 10 competences account for 75% of the competences employed, while the figure is substantially lower for Natural Science (67%) and Computer Science (62%). Mathematics therefore appears to be different from the other two departments since it seems that it uses a slimmer span of competences. Furthermore, SOLO 5 level competences are not part of the top 10 at the Mathematics department whereas two of the competences in the two other departments are at SOLO level 5. Hence Mathematics seems generally to be more careful with using SOLO 5 competences.

Taking the top six of each department we get a list of 12 competences: 'describe', 'explain', 'apply method', 'implement', 'analyze', 'discuss', 'account for', 'reproduce', 'solve', 'formulate', 'prove', and 'argue'. If we compare their frequencies (in percent), we get the picture seen in Figure 16.

We see here that Mathematics again seems to separate out from the two other departments, which seem more alike than different. Particularly regarding the competences 'reproduce', 'formulate', 'prove', 'solve', 'apply method', and 'argue', Mathematics has at least twice as high a frequency as the two other departments. In relation to the competences 'describe', 'explain', 'analyze', and 'discuss', Mathematics has a frequency less than half that of the other departments. Only in the competences 'account for' and 'implement' does Mathematics seem less distinct from, and indeed to lie between the other two departments. It is only in 'implement' that Computer Science seems to be clearly separated from the other two departments. It seems that certain competences are much more common in some departments than others. We therefore grouped clusters of the competences that had separated out to see how much of the total amount of competences within a department they jointly accounted

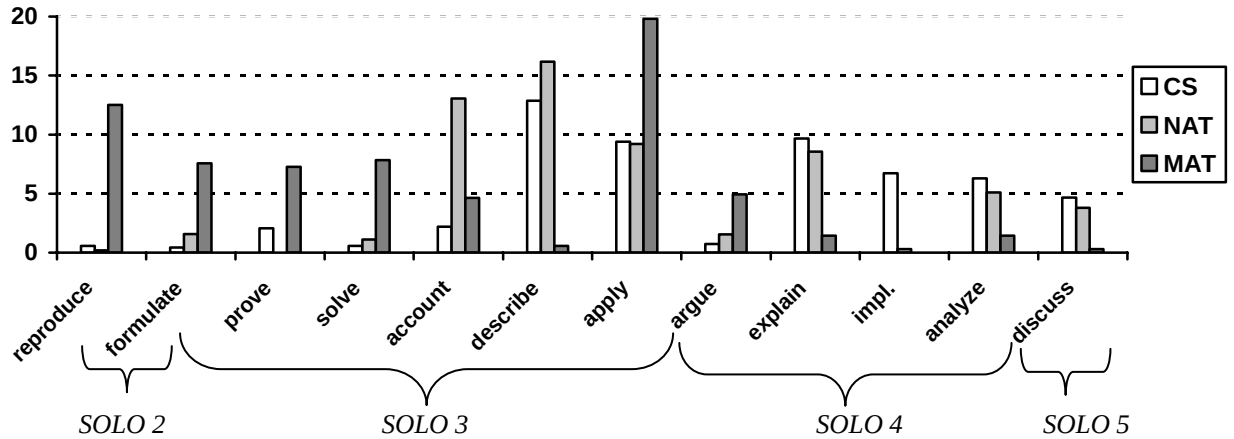


Figure 16: Freq. of common Computer Science competences compared to that of Mathematics and Natural Sciences.

Competence group:	Programming-related competences:	Competences where Mathematics is at 2x frequency (or more):	...and without 'apply':
Competences:	- implement SOLO 4 - program SOLO 4 - design SOLO 4 - construct SOLO 4 - structure SOLO 4	- reproduce SOLO 2 - formulate SOLO 3 - prove SOLO 3 - solve SOLO 3 - apply SOLO 3 - argue SOLO 4	- reproduce SOLO 2 - formulate SOLO 3 - prove SOLO 3 - solve SOLO 3 - argue SOLO 4
Computer Science:	**15%**	14%	4.5%
Natural Sciences:	1.0%	14%	4.4%
Mathematics:	0.3%	**60%**	**40%**

Figure 17: Competence clusters ('programming' and 'math-manipulation') by departments

for. This was to determine if some competences, regardless of SOLO level, were department-specific.

Grouping sets of competences related according to certain criteria provides interesting information, as shown in Figure 17.

Computer Science
Many people equate Computer Science with 'programming computers', but programming-related competences (e.g., 'program' 'implement', 'design', 'construct', and 'structure') occupy only 15% of the Computer Science curriculum (Figure 17). The same set of competences are negligible on the Natural Science curriculum, at 1.0%, and virtually non-existent on the Mathematics curriculum, at 0.3%. As expected, we thus see Computer Science standing out in this respect: we see that although programming is definitely an essential part of Computer Science – hence its reputation – it is by no means the main part.

Mathematics
Combining the competences where Mathematics distinctively separated itself out (i.e., 'reproduce', 'formulate', 'prove', 'solve', 'apply (method)', and 'argue') we see that together they account for 60% of all competences on the Mathematics curriculum; yet the same competences are remarkably less dominant in the other departments, with 14% for both Computer Science and Natural Science. In this respect, Mathematics seems to be distinctively different and one might argue that mathematics (at least the teaching of university mathematics) is to a large extent about reproducing, formulating, proving, solving, applying methods, and arguing. It so happens that according to the SOLO Terminology, these competences are 'lower' level, which is not to indicate that they are easy, nor that mathematics is in some sense 'lower' than the other subjects. If we remove 'apply', which often occurs in the other departments without directly involving mathematics, the difference ratio increases from about `1:4` (i.e., `60:14%`) to about `1:9`.

Natural Science
For Natural Science we tried to identify 'laboratory skills', but were unable at present to clearly isolate them based on competences alone. The competences 'execute (procedure)' and 'carry out (instructions)', which may be somewhat related to laboratory work, come to 4.2% for Natural Science, 2.5% for Computer Science, and 1.7% for Mathematics. We were not able to come up with a logically related set of competences that strictly isolated Natural Science from Mathematics and Computer Science.

4 Conclusion

Different departments
All three methods of analyzing the data suggest the same conclusion, namely that the three departments are different in their use of SOLO competences. The overall SOLO averages seem quite different for Mathematics and Computer Science, with Natural Science as a comparison partner also being different from the other two. Most of the time, Mathematics stands distinct from Computer Science and Natural Science, which, although different, seem to be closer to each other than to Mathematics. One might wonder what the reason is for the difference between Mathematics and the other Natural Science subjects, including Computer Science. One reason might be that Mathematics is usually considered to be a *vertical* discipline, with a hierarchy of theories and methods building upon each other, whereas many other disciplines are *horizontal*, with theories and methods living side by side (Madsen & Winsløw 2007). Furthermore, one can argue that we are researching the *SOLO* competences and the SOLO model might not be as appropriate a tool for describing *mathematical* competences as for other subjects. The SOLO Taxonomy was created using all higher education subjects, not *only* mathematics. In fact extensive work has been done within mathematics education to describe what distinguishes *mathematical* competences; for instance, problem-solving, reasoning and proving, communicating, connecting, representing etc. (NCTM 2000; Niss 2002; PISA 1999).

Different competences (and SOLO distributions):
The three disciplines show differences not only at the level of averages but also in the usage and distribution of verbs. Mathematics seems to use lower SOLO levels, Computer Science more higher-level verbs, and Natural Science in the middle. Also, in terms of the distribution of the specific verbs, Mathematics seems quite different

from the two other departments. Computer Science stands out with its inclusion of programming-related competences and the fact that the majority of the competences are qualitative, while the majority of the Mathematics and Natural Science competences are quantitative. All this suggests that Mathematics and Computer Science are in fact the most 'different' departments of the three. This might seem remarkable given the mathematical nature and basis of Computer Science and that Computer Science 'grew out of' and is part of the Mathematics department at many universities. What might the reason be? It needs further study to answer, but as stated above, the characteristic competences at each of the three departments are placed by the SOLO model at different levels. Does this then *finally and ultimately* explain the core of the subjects? No, it is to some extent also a reflection of the different teaching (and cultural) traditions in the three departments. Furthermore, as stated above, one could also argue that the SOLO taxonomy might not fit each department equally well.

SOLO and Constructive Alignment

Although the SOLO taxonomy might not fit each department equally well, it is good tool to help create a discussion about the purposes of a course and the formulation of ILOs. It is also a helpful tool to point to areas of interest when analysing various syllabi and departments. Also, the process at the two university faculties of implementing ILOs based on the principles of constructive alignment and the SOLO Taxonomy have given us a big standardized dataset and the opportunity to investigate the syllabi. Something that was not possible before, partly due to the 'private' manner in which syllabi were written – usually each teacher with his own personal style. A note of caution is needed, however. Sometimes there are practical reasons, such as time or fiscal constraint, room availability etc., that constrain a teacher to hold exams in less than ideal circumstances. Taking the example from above, it could be that the teacher is forced to use the cheaper MCQ test instead of an oral exam, which he would have preferred owing to his ideal ILOs that mention competences such as 'construct' and 'formulate'. In that case, aligning his course to the actual MCQ test would force him to rewrite his ILOs to a much lower SOLO level, i.e. less ambitious and relevant. Should he actually do that – in the name of alignment? The question is difficult, but in these circumstances it might be better if he keeps his higher SOLO level ILOs, align his teaching to these ILOs even though Susan's behaviour is then not fully rewarded at the exam. She will get her reward later.

'Creating' a language for competences (over time)

One of the purposes in implementing the same model at both faculties is that it can help create a common language to be used among university teachers, since often any "speech community is likely to be composed of different groups, groups which may operate with differing versions of the same language or even with discrete and separate language" (Montgomery 1992, p. 101). Also, the syllabi prior to adopting the SOLO Taxonomy were for the most part not formulated using measurable operational competences. Finally, all these things might ultimately synergize to improve learning since students would also, over time, be used to reading the operational syllabi, hence be more aware of what is expected, and hence be more inclined to act like Susan rather than Robert.

5 Acknowledgements

The authors would very much like to thank Anne Mette Mørcke, Berit Eika, Gitte Wichmann-Hansen, John Biggs, and Catherine Tang for providing extensive and valuable feedback on our SOLO classification of verbs. Also thanks to Torben K. Jensen and Gunnar Handal for interesting discussions about our analysis and the background for it.

6 References

Bauersfeld, H. (1979). Research related to the mathematical learning process. (In International Commission on Mathematics Instruction, ICMI (Ed.), *New Trends in Mathematics Teaching, Vol. IV* pp. 199-213. Paris: UNESCO)

Biggs, J. B. & Collis, K. F. (1982). *Evaluating the Quality of Learning: The SOLO Taxonomy, Structure of the Observed Learning Outcome*. (London: Academic Press)

Biggs, J. B. (2003). *Teaching for Quality Learning at University*. (Maidenhead: Open University Press)

Boag, S., Chamberlin, D., Fernández, M. F., Florescu, D., Robie, J., and Siméon, J. (2003). *XQuery 1.0: An XML Query Language*. [Working Draft: http://www.w3.org/TR/xquery/]

Brabrand, C. (2007). *Constructive Alignment for Teaching Model-Based Design for Concurrency*. Proc. 2nd Workshop on Teaching Concurrency (TeaConc'07), Siedlce, Poland

Brabrand, C. & Andersen, J. (2007). *Teaching Teaching & Understanding Understanding.* 20 minutes award-winning short-film on Constructive Alignment (available on DVD from www.unipress.dk), [homepage: www.daimi.au.dk/~brabrand/short-film/], Aarhus University Press

Brabrand, C. & Dahl, B. (2008). Using the SOLO-Taxonomy to Analyze Competence Progression of University Science Curricula. Paper submitted to international journal on 15 January 2008.

Goodlad, J. I., Klein, M. F., Tye, K. A. (1986). The Domains of Curriculum and Their Study. (In B. B. Gundem (Ed.), *Kompendium 3 Om læreplanpraksis og læreplanteori* (pp. 31-64). Pedagogisk Forskningsinstitutt, University of Oslo) (Reprint of parts of Goodlad et al. (1979) book: Curriculum Inquiry. New York: McGraw-Hill Book Company)

Madsen, L. M. & Winsløw, C. (2007). Relations between Teaching and Research in Physical Geography and Mathematics at Research Intensive Universities. *Manuscript, accepted modulo revisions for*

International Journal of Science and Mathematics Education

Montgomery, M. (1992). *An Introduction to Language and Society*. (London: Routledge)

NCTM (2000). *Principles and Standards for School Mathematics*. National Council of Teachers of Mathematics: VA, USA. http://www.nctm.org/standards/default.aspx?id=58

Niss, M (2002). Kompetencer og matematiklæring – Ideer og inspiration til udvikling af matematikundervisningen i Danmark. *Temahæfteserie nr. 18.* Copenhagen: Ministry of Education http://pub.uvm.dk/2002/kom/

Oliver, D., Dobele, T., Greber, M., Roberts, T. (2004). This Course Has A Bloom Rating Of 3.9. *Australian Computer Society, Inc. Proceedings of the Sixth Australasian Computing Education Conference (ACE2004), Dunedin, NZ. January 2004 Conferences in Research and Practice in Information Technology, Vol. 30, 227-31. Editors Raymond Lister and Alison Young.*

Oppenheim, A. N. (1992). *Questionnaire Design, Interviewing and Attitude Measurement*. (London: Pinter)

PISA (1999). *Measuring Student Knowledge and Skills*. OECD Programme for International Students Assessment, Paris: OECD. pp. 41-57

Robson, C. (2002). *Real World Research*. (Oxford: Blackwell)

Appendix A: The data material

The data for the entire analysis is available online (in browsable XML format) here, in Danish:

Online location (URL):	Data	Description
www.itu.dk/people/brabrand/solo.xml	SOLO data	SOLO attribution of all competences occurring in both data sets (cf. below).
www.itu.dk/people/brabrand/data-au.xml	AU data	All competences for all ILOs for all courses at all NAT/AU departments.
www.itu.dk/people/brabrand/data-sdu.xml	SDU data	All competences for all ILOs for all courses at all NAT/SDU departments.

The data is represented in the following XML format:

```
<!ELEMENT xml (institute*)>
<!ELEMENT institute (group*)>
<!ELEMENT group (course*)>
<!ELEMENT course (goal*)>
<!ELEMENT goal (competence*)>

<!ATTLIST institute name CDATA #REQUIRED>
<!ATTLIST group name CDATA #REQUIRED
                season CDATA #REQUIRED
                year CDATA #REQUIRED>
<!ATTLIST course name CDATA #REQUIRED
                 ects CDATA #REQUIRED
                 id CDATA #REQUIRED>
<!ATTLIST goal value CDATA #REQUIRED>
<!ATTLIST competence value CDATA #REQUIRED>
```

Data representation (in DTD format)

```
<xml>
<institute name="Computer Science">
<group level="Undergrad" season="Fall" year="2007">
  <course name="Algorithms and Datastructures II"
          ects="5" id="7819">
    <goal value="construct and analyze algorithms
                 using standard algorithm paradigms">
      <competence value="construct" />
      <competence value="analyze" />
    </goal>
    <goal value="identify and formulate algorithmic
                 problems as graph and string
                 problems">
      <competence value="identify" />
      <competence value="formulate" />
    </goal>
...
```

Sample data fragment (translated from Danish)

The data can then be queried by, for instance, XQuery (Boag et al. 2003) programs such as the following which calculates the frequencies of all *SOLO level 4* competences from *Computer Science* courses at AU and outputs them in *descending* order of their frequency counts:

```
<result>
{
let $competences := fn:doc("data-au.xml")//institute[@name eq "Computer Science"]//competence
  let $verbs := $competences/@value
  for $verb in fn:distinct-values($verbs)
  let $solo := fn:doc("solo.xml")//competence[@value = $verb]/@solo
  let $frequency := fn:count($all_competences[@value = $verb])
  where $solo eq 4
  order by $frequency descending
  return <competence value="{$verb}" solo="{$solo}" freq="{$frequency}"/> }
</result>
```

Appendix B: Sticky yellow notes

At the beginning of his keynote presentation, Claus Brabrand posed the question, *"What is good teaching?"*. The audience members wrote the following answers …

... that helps students reach the learning goals that they/we have set.

When you get the pupils in a way that they want to learn it.

Good knowledge on your subject. Clearly present that to others.

Understanding that teaching doesn't mean that students actually learn ...

Something that reaches the student and makes him think.

Activation, provocation; leads to new insights and motivation.

- Student-centred
- Teacher-student interaction
- Giving the student responsibility for the learning
- Active teaching-learning environment

To share knowledge and to have contact with learners.

Inspiring a student to learn beyond what they are "taught".

... is getting learners to have the skills and concepts and attitude.

To not to teach but tutor or guide the learner through learning.

Seeing relevance to facts
- Student outcome

Clear
Organised
Interactive

Creating possibilities

Good teaching is teaching that inspires good learning

Teaching that motivates students to want to learn

Good teaching supports students' learning process so that it is faster, more complete

Clear
Inspiring

Creating an understanding in students with an appreciation of its value.

Good teaching happens when the teacher jumps into the head of students, understands their preconceptions.

Good teaching facilitates "good" learning outcomes for as wide a range of learners as practicable.

Is what helps others to learn; does more good than harm ☺

- Teaching that engages
- Teaching that takes the perspective of the student and focuses on the subject

Teaching that informs effectively by entertaining.

What advances good learning

Good teaching is teaching that warrants good learning.

- interactive
- motivating

Good teaching is making students want to learn for themselves, not for the exam.

Support students to learn

Good results in a good atmosphere

When the students don't notice that they are being taught

Good teaching is the science of helping students to move information in working memory to long term memory ("Cognitive Load Theory")

Teaching that helps students form a correct and useful understanding

Good teaching activates the student to learn and think by himself

Good teaching leads to good understanding and a desire to learn more

Teaching that facilitates student learning

Teaching that
* motivates
* starts from the knowledge the student has
* has a clear goal

listening

inspiration

Claus Brabrand gives his own answer to the question in his 19-minute award-winning short film (DVD) about Constructive Alignment, "*Teaching Teaching and Understanding Understanding*". You can watch and order the film at:

http://www.daimi.au.dk/~brabrand/short-film/