Constructive Alignment in Teaching Modeling

Birgit Demuth

Technische Universität Dresden, Faculty of Computer Science

Abstract. Constructive Alignment devised by John Biggs is a widespread didactic approach and one of the most influential ideas in higher education. The basic assumption is that a course is designed in a way that learning activities and assessment tasks are aligned with intended learning objectives. Additionally Bloom developed a taxonomy which classifies learning objectives. This taxonomy was revised later and became a powerful teaching tool. In our large-scale software engineering course for undergraduate students focused on object-oriented modeling and programming we were unsatisfied in the past with the exam results. Therefore we adapted our course to the Constructive Alignment ideas and the revised Bloom's taxonomy. We started to explicitly specify our intended learning objectives for each topic, and we aligned learning activities and assessment criteria in the software engineering course to make the relationship between learning and assessment for the students transparent. In this paper we formalize the used basic concepts and report on our approach applying Constructive Alignment in a course for undergraduate students.

1 Introduction

A frequent observation is the gap between the teacher's and the student's perspective on a course. The teacher thinks: *What I teach is what topics I have in mind.* In contrast the student thinks in terms of *What you test is what I learn.* John B. Biggs proposed the Constructive Alignment approach [3] to bridge this gap and to provide the student with a clearly specified goal. This approach became one of the most influential ideas in higher education. The basic assumption is that a course is designed so that learning activities and assessment tasks are aligned with learning objectives that are intended in the course. We need to think about learning as what we want the student to do. Bloom developed a taxonomy which classifies learning objectives. This taxonomy was revised later [4] to become a powerful teaching tool. In software engineering education in general, Constructive Alignment was investigated by Armarego [2] in 2009. Armarego showed that a relationship exists between learner and learning model, and that this relationship should be exploited in the development of courses. However, to the best of our knowledge there is no study how to apply constructive alignment in teaching modeling.

In our software engineering courses we tell the students how important a sound requirements elicitation and documentation is. Requirements should be

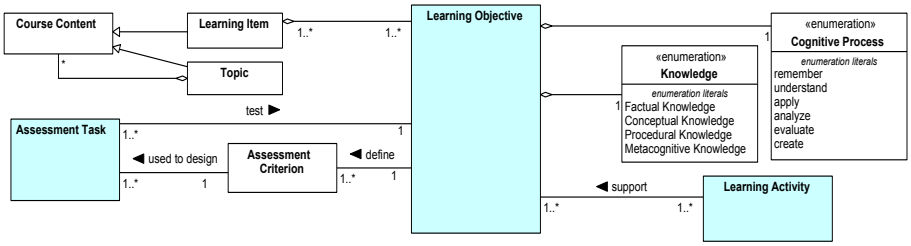

Fig. 1. Domain model of basic concepts

specified in such a manner that test cases are derivable. Additionally, we have a common set of phrase patterns to describe these requirements. Requirements in software development are the learning objectives in teaching. Software tests correspond to assessment tasks. Why do we often not apply this successful technique in teaching to define learning objectives and assessment tasks? Another state-of-the-art software development technique is test-driven development. We tell the students to program the test first. Constructive Alignment argues in a similar manner. After the definition of learning objectives, the teacher should derive the assessment tasks and subsequently choose the adequate learning activities. We should live what we teach in software engineering.

In this paper we report about first experience to apply Constructive Alignment in our courses for undergraduate students. In the following, we introduce basic concepts first in Sect. 2. In Sect. 3 we show how we applied Constructive Alignment to our software engineering course for undergraduate students to didactically improve our teaching approach. To conclude we summarize our approach and give an outlook for further research in Sect. 4.

2 Basic Concepts

The main concepts of Constructive Alignment are learning objectives, learning activities and assessment tasks. All three main concepts must be aligned so that the students are able to achieve the learning objectives by the selected learning activities. The assessment tasks must allow the student to demonstrate that the learning objectives are achieved.

We illustrate the relationships of these and related basic concepts used in this paper by means of a UML class diagram (domain model) (cf. Fig. 1). It is recommended to start the planning of a course with the definition of the *Learning Objectives*. All learning objectives should be assigned to the units of a course and inversely. This helps the teacher to structure his course and to concentrate on the learning objectives. We consider the *Course Content* as hierarchy of *Topics* and *Learning Items*. Learning items represent the leaves in the course content hierarchy.

Since we want the students to do things, it makes sense to specify the learning objectives in terms of verbs. This will also have the added benefit of leading us to design assessment tasks that measure the outcomes. In considering the verbs we refer to Bloom's taxonomy revised by Krathwohl [4]. In this taxonomy, two dimensions are distinguished. One is the *Cognitive Process* dimension representing verb forms to describe the learning objectives, and the other is the category of *Knowledge*. The Cognitive Process categories differ in their complexity from the lowest level *remember* to the highest level *create*. Each of these categories (verb forms) can be instantiated by typical respectively alternative verbs describing the required cognitive level. Additionally, each intended activity specified by a verb refers to a Knowledge category. *Factual Knowledge* is the basis knowledge that students must know like terminology and their specific details. *Conceptual Knowledge* refers to classifications, principles and theories. *Procedural Knowledge* includes knowledge of algorithms, techniques, methods amongst others. *Metacognitive Knowledge* includes knowledge about when, what and how to use strategies for learning or for problem solving. A defined *Learning Activity* should cause the students to engage with learning to attain the learning objectives. We need approaches that require participation of students that are active and encourage high-level learning. It is important for the teacher to carefully choose such learning activities that satisfy the learning objectives. Each learning objective should define one or more *Assessment Criteria*. Hereby, the teacher should combine different levels of learning objectives. This requires consideration of some lower level learning objectives that deal with the basic facts, as well as to incorporate higher levels that require the students to deal with new situations. An assessment criterion helps to design a set of *Assessment Task*s. The assessment tasks together with their classification according to the learning objective taxonomy can be added to a task repository which is then easily manageable by the teacher to perform oral, written or electronic tests. The teacher should pick assessment tasks of every relevant Cognitive Process and Knowledge category combination to balance the complexity of tasks in his test.

3 Application of Constructive Alignment in a Software Engineering Course

In our first year's software engineering course, computer science students are introduced to the software development process. We teach them object-oriented thinking, modeling and programming demonstrated by using UML and Java [1]. In this challenging course, we were not satisfied with the learning results of students. Hence we revised our course in due strict consideration of the Cognitive Alignment ideas.

Learning objectives are usually expressed in terms of a subject matter, and a description of what is to be done (verb) with or to that subject matter [4]. We use the phrase *A student shall be able to* <*verb*> <*subject matter*> to express a learning objective.

Table 1. Representative verbs used for learning objectives in teaching modeling

remember	understand	apply	analyze	evaluate	create
recall	exemplify	undertake	analyze	review	model
list	explain	map	distinguish	discuss	design
		visualize	abstract	decide	
		derive	read		
		reuse	identify		

Table 2. Learning objectives examples

A student shall be able to ...	Knowledge	Cognitive Process
list the basic properties of an object	Factual	remember
explain the difference between a structural and a behavioral model	Factual	understand
undertake a CRC card session	Procedural	apply
distinguish an object from a class	Conceptual	analyze
abstract things	Metacognitive	analyze
review a UML model	Conceptual	evaluate
model an application domain	Conceptual	create

There are multiple resources of typical verbs representing the different Cognitive Process levels[1]. In Table 1, we list a subset of the verbs which we consider adequate for teaching modeling. In Table 2, we list a few examples of learning objectives and classify them accordingly by their Knowledge category and Cognitive Process dimension.

In the recent courses in 2014, 2015 and 2016, we introduced the Cognitive Alignment approach, and we could basically observe improved results as measured by the success rate and the achieved average examination mark[2] in the written exam after completion of the course (cf. Table 3). The preconditions for exam in 2013 and later exams were the same particularly with regard to course content and exam complexity/structure. The examination marks reflect - besides modeling assessment tasks - the results of programming tasks of students. The presented exam results show that an in-depth empirical study considering further influencing factors is required to analyze the effect of a constructively aligned modeling course. In 2016, for example, we introduced gamification by optional quizzes as new kind of learning activities. Unfortunately, we are not allowed in accordance with our module regulations to test learning activities before the written exam is conducted. However, we observed an increased tutorial participation and got feedback of students that they are more motivated to concentrate on learning items and to accept learning activities. In tutorials, students explicitly stressed that the documented learning objectives help them to focus on the course content. A side effect was that the teacher understood

[1] e.g. `http://www.celt.iastate.edu/teaching/effective-teaching-practices/revised-blooms-taxonomy`

[2] scale 5 (means failed) to 1 (best examination mark)

more in depth what problems the students deal with. The most difficult-to-meet requirement for unexperienced students is abstraction. This insight helps us to better align the planned learning activities with the learning objectives.

Table 3. Written exam results without (2013) and with (2014, 2015 and 2016) the Constructive Alignment approach

	2013	**2014**	**2015**	**2016**
Number of students taken part	302	237	264	274
Success rate	43%	79%	59%	84,7%
Average examination mark	3,9	2,8	3,6	2,8

4 Summary and Outlook

We presented the use of Constructive Alignment ideas in conjunction with the revised Bloom taxonomy in a large-scale software engineering course for undergraduate students. We provided a model of the unification of both approaches. We have seen that Constructive Alignment fosters clarity in the course design as well as transparency in the relationship between learning and assessment. Students are more motivated to concentrate on learning items and to accept learning activities.

In future, we will refine our set of aligned learning objectives, learning activities and assessment tasks. We create a respective repository for the teacher considering all levels of the Cognitive Process dimension and the Knowledge category. The repository should also provide a reviewing system for quality assurance of assessment tasks. Furthermore, we plan to identify by an empirical study which factors improve the modeling skills of students categorized by the Cognitive Process dimension and Knowledge categories.

References

1. Akayama, S., Demuth, B., Lethbridge, T.C., Scholz, M., Stevens, P., Stikkolorum, D.R.: Tool use in software modelling education. In: Proceedings of the Educators' Symposium co-located with ACM/IEEE 16th International Conference on Model Driven Engineering Languages and Systems (MODELS 2013), Miami, USA, September 30th (2013), `http://ceur-ws.org/Vol-1134/paper6.pdf`
2. Armarego, J.: Software Engineering: Effective Teaching and Learning Approaches and Practices, chap. Constructive Alignment in SE Education: Aligning to What?, pp. 15–37. Information Science Reference (an imprint of IGI Global) (2009)
3. Biggs, J., Tang, C.: Teaching for Quality Learning at University. McGraw Hill Education, England, third edn. (2007)
4. Krathwohl, D.R.: A revision of blooms taxonomy: An overview. Theory into Practice 41(4) (2002)