Bloom's Taxonomy Revisited: Specifying Assessable Learning Objectives in Computer Science

Christopher W. Starr, Bill Manaris, and RoxAnn H. Stalvey
Computer Science Department
College of Charleston
66 George Street
Charleston, SC 29424, USA
{starrc, manarisb, stalveyr}@cofc.edu

ABSTRACT

Traditionally, Bloom's Taxonomy has been used for creating exams and other student assessment instruments. In this paper, we advocate its use for specifying learning outcomes in computer science prior to assessment. We have found, over a period of three years, that this facilitates programmatic assessment and related accreditation activities; it benefits instructors selecting pedagogical tools and assignments; and it enhances communication among faculty engaged in curricular development. We describe Bloom's Taxonomy, illustrate a simple process for applying it in computer science (and other disciplines), and present a case study of how it may be applied in a CS1 course. We believe this process has considerably strengthened our department's assessment program and its ability to maintain its ABET accreditation.

Categories and Subject Descriptors

K.3 **[Computers & Education]:** Computer and Information Science Education – *accreditation, computer science education, curriculum, self-assessment.*

General Terms

Management, Measurement, Documentation, Design.

Keywords: Accreditation, Assessment, Bloom's Taxonomy, CS1, Curriculum Development, Learning Objectives

1. INTRODUCTION

One of the problems faced by the average computer science (CS) faculty member is bridging the two gaps between (a) a course description and what to teach in the classroom, and (b) what has been taught in the classroom and how to assess what the students have learned. Additionally, when others examine the assessment instruments used (e.g., tests and quizzes), they should be able to see how these relate, in a substantive way, back to the course description, thus completing the cycle of planning, teaching, learning and assessing.

Bloom's Taxonomy (BT) has traditionally been used to help bridge the second gap, i.e., for educational assessment. In this paper, we advocate its use to help bridge the first gap, i.e., to specify assessable learning objectives. The approach described herein strengthened our department's assessment program and had a positive and significant impact on the results of our most recent ABET program evaluation.

We hope this paper will help advance the on-going discussion among CS educators on how to best integrate BT into CS curricular activities (e.g., [7, 11, 12, 13]).

Our work is motivated by:

(a) the 2009-2010 ABET-CAC accreditation criteria, which will require departments to state and evaluate educational program outcomes;

(b) the realization that BT may considerably improve specification of CS course materials, including topic outlines, lectures, in-class collaborative activities, and assignments; and

(c) the observation that this application of BT may enhance communication among faculty engaged in curricular development and course deployment activities.

The 2009-2010 ABET Criteria for Accreditation of Computing Programs in the US require hosting institutions to document educational objectives and regularly assess learning outcomes [5]. Assessment of learning objectives stated in measurable terms is helpful for ABET-accredited programs, as well as programs in other institutions with a strong emphasis in CS/IS/IT learning outcomes. Use of BT facilitates establishing a connection from the degree program objectives down to the exam question level through an explicit dependency chain.

Secondly, BT improves course preparation and delivery through better specification of course materials. For example, what does it mean to cover loops in a CS1 course? Some faculty may interpret this as a directive to explain the syntax of for/while loops and trace a few examples on the board. Then, they may ask students to develop their own loops on an assignment, and be surprised to see, semester after semester, that many of the students are lost.

Finally, communication among faculty is enhanced through the specification of learning outcomes for courses and programs. A Bloom-level specification of each learning objective enables more concise and informative communication during course development and deployment.

The next section presents a historical overview of how BT has been applied in (a) assessment (as it was originally envisioned), and (b) the specification of learning objectives within CS and software engineering curricula. Section 3 defines the six levels of BT and provides examples. Section 4 explains how to apply the taxonomy in specifying assessable learning objectives in CS. Section 5 is a case study presenting the specification of CS1

SIGCSE'08, March 12–15, 2008, Portland, Oregon, USA.

assessable learning objectives. Finally, Section 6 discusses implications and provides closing remarks.

2. BACKGROUND

In 1956, Benjamin Bloom, University Examiner at University of Chicago, together with other educational psychologists, proposed a classification of the various levels of knowledge mastery that may be achieved by a learner [2]. This taxonomy was motivated in part by the observation that most exam questions require only rote memorization and regurgitation of knowledge; as a result, such questions cannot truly assess how well a student has mastered the concepts. Bloom identifies six levels of learning mastery, namely **knowledge** (the term **recall** is used herein, similarly to [13]), **comprehension**, **application**, **analysis**, **synthesis**, and **evaluation**.

The software engineering education community was first, within the broader computing education community, to embrace BT for specification of learning objectives. In particular, the *Software Engineering Body of Knowledge* (SWEBOK) employs the taxonomy to specify the level of mastery for each topical area that a software engineer (generalist) should have achieved after four years of experience [1]. Topics in each knowledge area are assigned a Bloom level of learning mastery. The *Software Engineering Education body of Knowledge* (SEEK), based on SWEBOK, was developed as a minimum curriculum for instructors in undergraduate SE programs; it also employs BT to specify learning outcomes [14]. A preliminary mapping between SWEBOK and SEEK was made but did not exploit the learning levels [3].

Within CS, Lister and Leaney [12] used BT in an introductory programming course to assign grades based on Bloom-level mastery of tiered curricular components. Rather than grading on a curve, students earn a grade based on subject mastery. Students mastering at the Recall[2] and Comprehension levels earn a "C" grade; students at the Application and Analysis level earn a "B" grade; and, students mastering at the levels of Synthesis and Evaluation earn an "A" grade. .

Based on the above work, Burgess [4] demonstrated that a Bloom-based course assessment tool could be constructed and deployed in a second-level programming course. The result is the assignment of a grade based on objective measurements of learning outcomes. The paper describes the cognitive tasks required at each of the three grade tiers.

Scott [13] states that exams usually do not test students' knowledge across all levels of the Taxonomy; therefore, the instructor does not discover the level of mastery of a given topic for each student. The solution is to offer exam questions related to each level or to each tier for every topic covered on the exam. This technique ensures that students have the opportunity to demonstrate their achieved level of mastery. Of course, this is difficult to do, due to the combinatorial explosion of specifying six different questions (one per Bloom level), for every topic of interest.

More recently, Oliver et al. [9] used BT to assess the cognitive difficulty of computing courses in an IT program by formulating and calculating a *Bloom Rating*. A Bloom level was assigned to each assessment/test question according to the level of cognitive behavior required to properly answer it. A Bloom Rating for each course was determined using a weighted average of the Bloom levels of all the assessment materials for that course.

Finally, Manaris et al. [10, 11] applied BT within CS to specify learning objectives of human-computer interaction courses. They present a collection of courses for various target audiences, including freshman non-majors, junior/senior majors, and graduate students. For each course, they provide an outline containing learning objectives using BT, the amount of time to be spent on each topic, and related in-class activities.

This paper extends the above work by illustrating how BT may be used to specify assessable learning objectives throughout the CS curriculum; and provides pointers on how it may be used for programmatic assessment.

3. BLOOM'S TAXONOMY

Bloom describes levels of student learning within the cognitive dimension [2]. Given *a concept* to be learned by students, there are six levels of learning mastery (or competence): these are **recall**, **comprehension**, **application**, **analysis**, **synthesis**, and **evaluation** (see Table 1). Mastery at a particular level for a particular concept usually implies mastery at all prior levels.[3]

Table 1. Brief explanation of Bloom's levels, and sample questions for the concept "computer program".

Level	Explanation	Sample Question
Recall (RE)	The student is expected to recite memorized information about the concept.	"What is a *program*?"
Comprehension (CO)	The student is expected to explain the concept in his or her own words.	"How is a *program* similar to a recipe?"
Application (AP)	The student is expected to apply the concept to a particular situation.	"What is the output of this *program*?"
Analysis (AN)	The student is expected to separate materials or concepts into component parts so that their organizational structure may be understood.	"Create a top-down design for a *program* to perform a given task."
Synthesis (SY)	The student is expected to put parts together to form a whole, with emphasis on creating a new meaning or structure.	"Write a *program* to perform a given task."
Evaluation (EV)	The student is expected to make judgments about the value of ideas or materials.	"Given two *programs* that perform the same task, which one is better and why?"

[2] Lister and Leaney use the term Knowledge [12].

[3] Nevertheless, in some situations, it is possible for certain levels to be attained in parallel, e.g., synthesis and evaluation; or even out of order, e.g., evaluation prior to synthesis, as is the case with many established art critics (evaluation) who are not accomplished artists themselves (synthesis).

BT was not intended as a model of how humans learn. However, in attempting to better comprehend and apply it, the following observations may be of use:

- BT can be viewed as consisting of three meta-levels: (a) memorization and basic understanding (beginner level); (b) use or competent application (intermediate level); and (c) design or creation and critique (expert level).
- Within each of these meta-levels, we encounter the same two phases: (a) a *production* of an artifact of learning and (b) an *explanation* or analysis of that production (see Table 2).

Table 2. A Meta-level structure for Bloom's Taxonomy.

	Phase within Meta-Level	
Meta-Level	***Produce***	***Explain***
Beginner	Recall	Comprehension
Intermediate	Application	Analysis
Expert	Synthesis	Evaluation

CS students proceed through each of these levels when introduced to a new concept. For example, as a beginner, memorization of the concept "if statement" is exhibited by simply regurgitating (reproducing) a definition, and then by explaining what an "if statement" is in the learner's own words. At the intermediate level, a learner can apply (reproduce) the comprehension of the concept to specific instances (e.g., trace this "if statement"); this is followed by the ability to analyze (explain) how a particular instance of the concept works or relates to a real world experience in terms of its characteristics/components (e.g., explain how this "if statement" solves this problem). As an expert, synthesis is reached when the learner can produce a new instance of the concept (e.g., write an "if statement" that solves this problem); this is followed by the ability to evaluate (explain) the quality of instances of the concept (e.g., is this the best or reasonable "if statement" for solving this problem?).

4. APPLYING BLOOM'S TAXONOMY

Traditionally, CS courses are described using lists of topics, without providing any information about the level of expertise to be achieved by the students across those topics. This is also the case with the current, and previous, ACM Curricular Guidelines [6], in contrast to SWEBOK, which provides information as to the level of expertise to be reached.

When each topic is presented with a Bloom-level of mastery, the instructor is better informed as to what level of mastery is expected; thus (s)he will be able to determine the required time and necessary instruction to help the student achieve the proper knowledge level for each topic. Also, an instructor new to a course is given a more thorough specification of the course's pedagogical expectations. Minimally, a Bloom-level specification of learning outcomes may facilitate textbook selection, as well as preparation of lecture materials with higher confidence and potential effectiveness.

BT also provides a means for designing in-class activities; homework assignments; projects; and, naturally, assessment instruments, such as pop quizzes and exams. By understanding the level of expertise to be achieved by students, test questions may be proposed which accurately assess the students' expected mastery. This is very important.

Further, for advanced courses, it is most useful for the instructor to know the students' level of mastery of all major concepts taught in prerequisite courses. Otherwise, when provided only with a list of general topics from the prerequisite courses, an instructor can either assume the knowledge level or examine each assignment and test given in the prerequisite course.

BT provides a valuable tool for communication among faculty involved in teaching course sequences and/or involved in curricular development activities. These and other realizations have made the authors' department begin a multi-year project to incorporate BT in all course descriptions of the ABET-accredited BS degree program in CS, starting with the core courses.

4.1 A Process for Applying Bloom's Taxonomy

The process developed to apply BT in our department is straightforward and reproducible:

1. Identify/select the topics to be covered. These may come from a collection of sources, such as ABET, SWEBOK, an earlier syllabus, or available textbooks.
2. For each topic, consider each level of knowledge in turn (RE, CO, AP, AN, SY, EV) and decide which is the highest level of mastery that all students should achieve *upon completion* of the course.

We also had to consider what depth of topic coverage is desired for courses for which this course is a prerequisite. It is reasonable to have some topics covered at a Comprehension level while other topics are covered at an Evaluation level. The distribution of levels across topics depends on the complexity of each topic, and will likely exhibit some degree of variability within a course.

By instituting this process, the instructor creates a useful specification document, which may be used to (a) design targeted lectures that meet the needs of the students and effective test questions, and (b) assignments that engage students at the appropriate level.

4.2 A Topical Example: the "For Loop"

In the context of CS1, consider a simple example, the "for loop" topic. In a traditional CS1 course, an instructor will want students to be able to recite the definition of a "for loop" (Recall level), to be able to explain a "for loop" (Comprehension level), and to be able to trace code that contains a "for loop" (Application level). It is also probable that students should be able to decompose a "for loop" into its constituent parts, i.e., counter initialization, termination condition, statements to be repeated, etc. and explain how these relate to a problem specification (Analysis level). CS1 students may be expected to design a "for loop" from scratch to solve a particular problem (Synthesis level). However, it is less likely that a CS1 student will be expected to evaluate the quality (e.g., efficiency) of a "for loop" (Evaluation level). Therefore, a CS1 course description may list the "for loop" topic at the Synthesis level.

4.3 The Application of BT can be Slippery

Johnson and Fuller [7] report that "there is considerable disagreement" between faculty responsible for teaching and

assessing student performance (conveners), and faculty responsible for analyzing the assessment tasks (assessors).

Over a three-year period of applying BT within our department, a similar phenomenon was observed, which we have named *concept shifting*. Concept shifting occurs when the topic/concept being considered during Step 2 of the algorithm (Section 4.1) is inadvertently switched with a related (usually lower level or less abstract) concept. For example, the concept of "selection" may be switched with the concept of "if statement" while performing step 2; or the concept of "iteration" with the concept of "for loop"; or the concept of "abstraction" with the concept of "method".

Consider the subtle, yet significant distinction between the concept of "iteration" and the concept of "for loop". "Iteration" (a more abstract concept) may be implemented in many ways including a "for loop" (a more concrete concept). Assume that the CS1 instructor has chosen to apply BT on the concept of "iteration". Upon reaching the Synthesis level, the instructor inadvertently switches to "for loop" and carries on. "Iteration" at the Synthesis level would require students to be able to design a new way to perform a loop (most probably a graduate level topic). On the other hand, the concept of "for loop" at the Synthesis level requires students to be able to write for loops that perform a specific task (something clearly within the CS1 domain). Almost subconsciously, the instructor has switched to a lower-level concept that is more appropriate for CS1 at a higher Bloom level.

One possible explanation for this difficulty is that BT was never meant for specification, only for assessment. However, as mentioned in [8], this difficulty exists even within assessment. Another possibility is *task interference*. BT requires considerable effort to remember, especially if one is not well versed in it. This memory load affects one's ability to apply it well onto different topics (i.e., "at what level of the taxonomy should this topic be covered?"). Finally, it has been argued that BT is not a perfect model [7, 8]. In our opinion, although possibly imperfect, BT facilitates a greater level of granularity in course specifications, thus considerably improving course delivery and assessment.

5. LEARNING OBJECTIVES FOR CS1 – A CASE STUDY

This section presents a subset of topics from a traditional CS1 course in Java. These same topics are then presented as assessable learning objectives using descriptions related to the expected Bloom's level to be reached by the students upon course completion. Such objectives have been adopted by our department in the ABET-accredited B.S. degree program.

Table 3 shows a subset of the CS1 topic list in our department, prior to BT. Table 4 shows the same subset converted into assessable learning objectives using BT.

Clearly, this specification provides much more information (precisely, an extra dimension) than traditional CS1 course outlines (topic lists). Similarly to the SWEBOK disclaimer [1, p. D-2], this two-dimensional outline is one of many possibilities. Nevertheless, this has proved invaluable in our department for educational assessment at the course level; it has also improved our ability to discuss learning objectives and course materials; finally, it has provided a mechanism to analyze failures in content delivery (e.g., why did all students failed the test question on topic *x*?), and allow for refinement of teaching materials (e.g., because topic *x* was covered at the *Comprehension level* in class, whereas the test question required *Synthesis level*).

Table 3. Example of traditional CS1 topics (without Bloom's Taxonomy).

Topics to be covered:

- **Object-oriented design**
- **I/O**
- **Edit/Compile/Execute cycle**
- **Selection statements**

Table 4. Example of assessable learning objectives in CS1 using Bloom's Taxonomy.

Upon completion of this course, students should be able to:

- explain **object-oriented design** as a mechanism for handling problem complexity as well as facilitating team programming and software reuse (introduce a subset of UML) **[Comprehension level]**
- apply basic **I/O** operations to different data types **[Application level]**
- analyze the **Edit/Compile/Execute cycle** and apply its components in program development (e.g., executing pre-compiled class files) **[Analysis level]**
- design programs using **selection statements** (including nested if statements) **[Synthesis level]**

6. CONCLUSION

This paper proposes the use of Bloom's Taxonomy as a vehicle for exploration, specification, and refinement of assessable learning objectives in CS courses.

Motivated by the new outcomes-based ABET-CAC accreditation requirements starting in the 2009-2010 cycle, we illustrate how a faculty may express and document program objectives through the specification of learning outcomes at various levels, i.e., lectures, courses, and even degree programs. This, in turn, provides the basis for an improved program assessment process. In our department, the most recent ABET program review was positively and significantly impacted by our use of BT as presented.

Once Bloom levels have been applied to learning objectives, the instructor's job becomes much easier, in that the problem of designing a lecture to cover a particular topic becomes less nebulous, more clearly specified. This also applies to in-class activities, homework assignments, and test questions. In particular, the daunting task of designing exams is transformed from "What question should I include on this test?" to "Given that I taught topic *X* at the *Y* Bloom level, what exam questions are appropriate to assess if the students have reached this level of mastery? Should I also include related questions to assess lower levels of achievement?" Clearly, in most cases it may be impossible to include a complete spectrum of questions for the different levels of each topic on a single test; however, the proposed approach makes the different possibilities concrete, and may also provide inspiration for new test questions.

Finally, the use of BT for specifying learning outcomes facilitates faculty communication. For instance, a new faculty member presented with a course outline using BT is given far more precise instructions as to what needs to be covered in a particular course. Upper level instructors also have a better understanding of the topics covered in prerequisite courses. We hope, this approach, which is already in use by the SE community, will be adapted by the CS community at large as a vehicle for enhancing understanding and collaboration among CS educators, and for enabling more consistency in student learning experiences and topic mastery, thus ultimately improving the professional competency of our graduates.

7. ACKNOWLEDGEMENTS

The authors would like to acknowledge Anthony Leclerc, George Pothering, Renée McCauley, and Walter Pharr for their contribution to this work through extensive brainstorming and discussion. This work was supported in part by the National Science Foundation under Grant No. 0226080.

8. REFERENCES

[1] Abran, A. and Moore, J. M. *Guide to the Software Engineering Body of Knowledge: Version 2004 (SWEBOK),* IEEE, June 2004, viewed 11 January 2007 <http://www.swebok.org/ironman/pdf/Swebok_Ironman_June_23_ 2004.pdf>.

[2] Bloom, B. S. (ed.). *Taxonomy of educational objectives: Handbook 1: Cognitive domain*. Longmans, Green and Company, New York, 1956.

[3] Bourque, P., Robert, F., Lavoie, J-M., Lee, A., Trudel, S. and Lethbridge, T.C. Guide to the Software Engineering Body of Knowledge (SWEBOK) and the Software Engineering Education Knowledge (SEEK) - A Preliminary Mapping. *Proceedings of the 10th International Workshop on Software Technology and Engineering Practice (STEP '02)*. (Canada, October 6 – 8, 2002). IEEE, Piscataway, NJ, 2002, 35-44.

[4] Burgess, G.A. Introduction to Programming: Blooming in America. *Journal of Consortium of Computing Sciences in Colleges*, *12, 1* (October 6-8 2005), 19-28.

[5] *Criteria for Accrediting Computing Programs: Effective for the Evaluation During the 2007-2008 Accreditation Cycle.* ABET-CAC, Baltimore, MD, October, 2006, viewed 31 August 2007 <http://www.abet.org/Linked%20Documents-UPDATE/Criteria%20and%20PP/C001%2007-08%20CAC%20Criteria%208-8-07a.pdf>.

[6] Engel, G., and Roberts, E. (2001). *ACM-IEEE Computing Curricula 2001*, IEEE with ACM Press, December 2001, viewed 11 January 2007 <http://www.sigcse.org/cc2001>.

[7] Fuller, U., Johnson C., et. al. (2007). ITiCSE 2007 WG2 Working Group Report on Developing a Computer Science-Specific Learning Taxonomy. *SIGCSE Bulletin.* (to appear).

[8] Johnson, C.G., Fuller, U. Is Bloom's Taxonomy Appropriate for Computer Science? *Proceedings of the 6th Baltic Sea Conference on Computing Education Research (Koli Calling 2006)* (Koli, Joensuu, Finland, November 9-12, 2006), 120-123.

[9] Oliver D., Dobele, R., Greber, M., Roberts, T. This Course Has A Bloom Rating Of 3.9. *Sixth Australasian Computing Education Conference* (Dunedin, NZ, January 2004). Australian Computer Society, 2004, 227-231.

[10] Manaris, B., and McCauley, R. (2004). Incorporating HCI into the Undergraduate CS Curriculum: Bloom's Taxonomy Meets the CC'01 Curricular Guidelines. In *Proceedings of 34th ASEE/IEEE Frontiers in Education Conference* (Savannah, GA, October 20 – 23, 2004). IEEE, Piscataway, NJ, 2004, T2H10 - T2H15.

[11] Manaris, B., Wainer, M., Kirkpatrick, A.E., Stalvey, R.H., Shannon, C., Leventhal, L., Barnes, J., Wright, J., Schafer, J.B., Sanders, D., Implementations of the CC'01 Human-Computer Interaction Guidelines using Bloom's Taxonomy. *Computer Science Education Journal, 17, 1* (March 2007), in press.

[12] Lister, R., Leaney, J. Introductory Programming, Criterion-Referencing, and Bloom. In *Proceedings of the 34th SIGCSE technical symposium on Computer Science Education* (Reno, Nevada, February 19-23, 2003). ACM, New York, NY, 2003, 143-147.

[13] Scott, T. Bloom's Taxonomy Applied to Testing in Computer Science Classes. *The Journal of Computing in Small Colleges, 19, 1* (October 2003), 267-274.

[14] Sobel, A. E. K. *Computing Curricula -- Software Engineering Volume. Final Draft of the Software Engineering Education Knowledge (SEEK)*, IEEE with ACM Press, April 2003, viewed 11 January 2007 <http://sites.computer.org/ccse/know/FinalDraft.pdf>.