Is Bloom's Taxonomy Appropriate for Computer Science?

Colin G. Johnson
Computing Laboratory
University of Kent
Canterbury, Kent, CT2 7NF
England
C.G.Johnson@kent.ac.uk

Ursula Fuller
Computing Laboratory
University of Kent
Canterbury, Kent, CT2 7NF
England
U.D.Fuller@kent.ac.uk

ABSTRACT
Bloom's taxonomy attempts to provide a set of levels of cognitive engagement with material being learned. It is usually presented as a generic framework. In this paper we outline some studies which examine whether the taxonomy is appropriate for computing, and how its application in computing might differ from its application elsewhere. We place this in the context of ongoing debates concerning graduateness and attempts to 'benchmark' the content of a computing degree.

1. INTRODUCTION
Bloom's taxonomy was devised in the 1950s as a generic instrument for dividing the cognitive aspects of learning into hierarchical levels. It is now widely used in course design in higher education, as a way of ensuring that teaching and assessment strike the right balance between rote learning of content and high level skills such as synthesis and evaluation. The application of these cognitive levels now goes far beyond the design of individual modules [1]. Its influence can also be seen in attempts to define 'graduateness': what a student should be able to do at the end of a Bachelor's or Master's degree. Such specifications underpin the European Higher Education Area (EHEA)'s drive to ensure the international recognition of qualifications and the mobility of labour. The Bologna Declaration [8] has resulted in major higher education curriculum reform across most European countries and generic statements of competence at the end of the first, second and third cycles. There is an ongoing process, known as the Tuning Project [1], which is generating EHEA-wide, subject-specific statements of competencies akin to the UK's subject benchmarks [2].

A departmental attempt to improve assessment led the authors of this paper to apply Bloom's taxonomy to a number of first year modules and to wonder whether the ordering in its hierarchy is appropriate for computer science. This paper outlines our study of practice in a single university, and throws the question of the aptness of Bloom to computer science open to wider debate.

2. LEARNING TAXONOMIES
The learning taxonomy devised by Bloom et al [5] divides the cognitive aspects of learning into six hierarchical levels:

- Knowledge (recall of facts, et cetera)
- Comprehension
- Application
- Analysis
- Synthesis
- Evaluation

Bloom et al were somewhat equivocal about whether evaluation should be above or on the same level as synthesis and they were also not dogmatic about whether evidence of performance at a higher level necessarily demonstrated performance at all the lower levels.

There appear to be many interpretations of this taxonomy. Some teachers see the hierarchy as applying to individual topics. Every topic is capable of being approached at each of the levels, and the more successful the student is the higher the level she or he will reach. An alternative idea is that the hierarchy represents progress through the subject as a whole, for example in a degree programme. Under this interpretation, the lower levels correspond to early years of study, with the final aim of the programme being that all students will be enabled to achieve at the highest level.

Recent re-evaluation of Bloom's taxonomy by Anderson, Krathwohl et al [3] has suggested that the top two or three levels of the hierarchy may be flat (Figure 1). They have also proposed that the taxonomy should be two dimensional, with the (slightly reconfigured) original categories of Remember, Understand, Apply, Analyze, Evaluate and Create forming the cognitive process dimension and Factual, Conceptual, Procedural and Meta-Cognitive forming a knowledge dimension.

Whilst Bloom's taxonomy of the cognitive domain has the widest currency, it is not the only such taxonomy. For example, Bloom and his colleagues produced a much less well known taxonomy of the affective domain, while Biggs' SOLO taxonomy [4] charts increasing structural complexity in student learning outcomes. This identifies that learning first changes quantitatively, as the amount of detail in the students response increases, and then qualitatively, as the detail becomes integrated into a structural pattern.

The computer science education literature contains a small number of examples of the use of a taxonomy as an analytic tool. Bloom's taxonomy has been applied in course design; for example Scott [9] and Lister & Leaney [6] have used it for structuring assessments. Taxonomies have also

[1]In this paper we use the term module to denote a unit of learning that is assessed as a whole and might, typically, constitute a quarter, eighth or tenth of a year's study for a full-time student

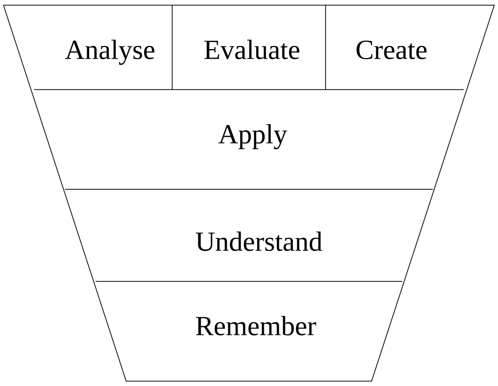

Figure 1: Bloom's Taxonomy 'flattened' [3].

been applied retrospectively, for example Lister et al [7] used the SOLO taxonomy to classify free-form responses to a problem-solving task

3. A STUDY OF ASSESSMENTS

A study was carried out which looked at all 54 assessments that were given to the first year students studying Computer Science in our university during one year. These were examined by a panel of five academics from the department (some of whom had been involved in these parts of the course, some not), who were asked to decide which of the levels in the Bloom taxonomy were being assessed by that particular assessment. The results are presented in Table 1.

4. INTERVIEWS WITH COURSE LECTURERS

A structured interview was held with the lecturer who was responsible for organising (and teaching a large component of) each of the first-year modules. As part of this interview, the lecturer was asked about the use of the various Bloom levels, whether they were relevant both to the material taught in the module, and, more specifically, whether they were evaluated as part of the module.

This part of the interview was introduced with a preamble about how learning can involve different levels of understanding according to the material being learned, and that assessment can emphasize these different levels.

Table 2 contains the questions, a sample of answers, and a summary of how many modules assessed material at this level.

5. COMMENTS AND OBSERVATIONS

A number of observations can be made from our study of assessment in first year computer science modules. The first is that there is considerable disagreement between the academics responsible for the design and delivery of these modules (conveners) and the group who analysed all the assessment tasks (assessors) about the level at which assessment was being carried out. The assessors felt that the vast bulk of assessment was at the *application* level, while conveners considered that they were also assessing *analysis*. One reason for this could be the difficulty of determining the taxonomic level of the assessment without having an intimate knowledge of the way in which the material being assessed was taught. (This difficulty was identified by Bloom et al themselves). This could lead to a task that was taught explicitly to students, and thus should be regarded as testing application, being assessed as involving a higher level skill such as synthesis—or *vice versa*. Another possibility is that the conveners and the assessors had different understandings of the levels in Bloom's taxonomy. All the assessors, but only a minority of the conveners, had been involved in a study group on taxonomies and assessment, so this could be the case.

The other notable finding is that several of the conveners felt that the highest levels of Bloom's taxonomy—synthesis and evaluation—were not appropriate to their module. In some cases it was clear that this was because the convener subscribed to the view that these levels would not be addressed until the final year of the degree programme. In others it seemed that it was because they felt that application was the 'core' of what computing is about and so it is appropriate to concentrate on its development in teaching and assessment.

6. A PERSPECTIVE: APPLICATION AS THE AIM

Let us take forward the idea that *application* is the aim of computer science teaching. In many disciplines, the aim of study is to develop an informed, critical perspective on the subject. For example, a history graduate would be expected not just to know lots of dates but also to be able to make critical and comparative comments on historical events, based on knowledge and theories. On the other hand, this graduate would not be expected to apply their knowledge to producing new history. Thus in such a discipline the long-term aim of study is particularly oriented towards the synthesis and evaluation levels in the taxonomy.

As noted above, a significant feature of our study of assessment in computer science modules was that the focus of assessment appeared to be at the *application* level. We might hypothesise that in disciplines such as computing the aim of study is what we might term 'higher application'. Here we are using the word *higher* in the sense that is used in terms such as 'higher criticism' or 'higher journalism'—i.e. the application informed by a critical approach to the subject, but where the criticism is not, as such, the focus of the work. In such work, the focus is at the application level in Bloom's taxonomy—yet this needs to be informed both by levels that Bloom puts below *and above*. This is illustrated in Figure 2, which contrasts with Figure 1 by adding a *higher application* capstone level.

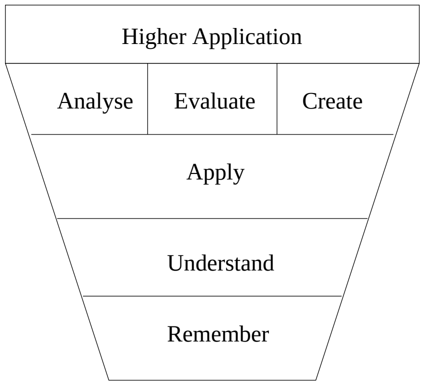

Figure 2: A suggested revised Bloom taxonomy for computing, incorporating *higher application*.

Bloom Level	Assessor 1	Assessor 2	Assessor 3	Assessor 4	Assessor 5	Mean
Knowledge	54	4	54	43	53	**42**
Comprehension	54	13	54	9	52	**37**
Application	51	43	54	5	29	**36**
Analysis	25	17	9	0	3	**11**
Synthesis	0	2	6	1	2	**2**
Evaluation	0	3	2	0	0	**1**

Table 1: Summary of assessment study: five academics rated the various assessments on the course and decided which Bloom level the material was at. The table shows how many of the assessments were rated as being at a particular level by each of the five assessors on the panel.

What other subjects might be said to have this characteristic? Clearly, subjects that are commonly compared with computing, such as engineering subjects, are of this type? Perhaps, though, this might point out similarities to more remote subjects—for example art and design subjects. Are there similarities in the way in which 'synthesis/evaluation used to improve application' is approached in those subjects? For example, peer criticism is a common approach in art and design education—is this because it is good for those subjects as such, or is it more because this is good generally for subjects that have the relationships between Bloom levels that these subjects have?

7. QUESTIONS FOR DISCUSSION

- Can a reformulation of Bloom's taxonomy provide more helpful descriptions for cognitive levels in Computer Science?
- Is the aim of computing education primarily focused on tasks that can be described as 'higher application' rather than evaluation/synthesis being the ultimate end-point of the educational process? If so, what can we learn from this?
- Should a taxonomy of learning inform the process of identifying points of reference for generic and subject-specific competences of first and second cycle graduates in Computer Science across the European Higher Education Area? If so, which taxonomy should be used?

8. REFERENCES

[1] Tuning project, tuning methodology. University of Deusto, 2004, Accessible at `http://tuning.unideusto.org/tuningeu/index.php?option=content&task=view&id=172&Itemid=205`, Accessed on June 28, 2006, 1999.

[2] Honours degree benchmark statement - computing. The Quality Assurance Agency for Higher Education, Gloucester, UK, 2000 `http://www.qaa.ac.uk/academicinfrastructure/benchmark/honours/computing.pdf`, 2000.

[3] Lorin W. Anderson and David A. Krathwohl. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives.* Addison-Wesley, 2001.

[4] John B. Biggs and Kevin F. Collis. *Evaluating the Quality of Learning: The SOLO Taxonomy.* Academic Press, 1982.

[5] Benjamin S. Bloom et al. *Taxonomy of Education Objectives (Volume 1 : Cognitive Domain).* McKay, New York, 1956.

[6] Raymond Lister and John Leaney. Introductory programming, criterion-referencing, and Bloom. In *SIGCSE '03: Proceedings of the 34th SIGCSE technical symposium on Computer science education*, pages 143–147, New York, NY, USA, 2003. ACM Press.

[7] Raymond Lister, Beth Simon, Errol Thompson, Jacqueline L. Whalley, and Christine Prasad. Not seeing the forest for the trees: Novice programmers and the SOLO taxonomy. In *Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education (ITICSE '06)*, pages 118–122. ACM Press, 2006.

[8] European Ministers of Education. The Bologna Declaration of 19 June 1999. Bologna, Italy, 1999. Accessible at `http://www.bologna-bergen2005.no/Docs/00-Main_doc/990719BOLOGNA_DECLARATION.PDF` accessed 4 August 2006, 1999.

[9] Terry Scott. Bloom's taxonomy applied to testing in computer science classes. *Journal of Computing in Small Colleges*, 19(1):267–274, 2003.

Bloom Level	Questions and responses
Knowledge	**Questions:** *Is the direct learning of facts important in first year computer science, and in your module more specifically? Does this impact upon your module? If yes, do you assess this directly in your module?* **Sample comments:** *'Being able to use the right words is helpful; direct learning of a formula so that they can parrot it, no.'; 'it is a language learning course, to an extent, and languages are made of facts and things.'; 'Yes, direct learning of facts is important.'* **Assessment** at this level: 6/7 modules (the other 'marginally' assessed material at this level).
Comprehension	**Questions:** *Is the ability of students to explain the course material important in your module, and in first year computer science more generally? Does this impact upon your module? If yes, do you assess this directly in your module?* **Sample comment:** *'The first goal is to be able to do it, and then the second goal is to be able to explain it. Realistically I'm not sure how many of them can effectively explain what they're doing by the end, and I'm not sure how much I would let that affect my assessment. If the student does it, but does not explain it well, I would probably be reluctant to seriously penalise them for that. A proper, complete solution should include an explanation.'* **Assessment** at this level: 4/7 modules - two others 'partially'.
Application	**Questions:** *Is the application of techniques learned to new situations important in your module, and in first year computer science more generally? Does this impact upon your module? If yes, do you assess this directly in your module?* **Sample comments:** *'It is essential.'; 'Yes, extremely important.' 'The more different examples they encounter the better placed they are to understand that the foundational concepts apply regardless of the context of a particular problem.'* **Assessment** at this level: 7/7 modules
Analysis	**Questions:** *Is the ability to analyse a range of information and decide which aspects of learning to apply important in your module, and in first year computer science more generally? Does this impact upon your module? If yes, do you assess this directly in your module?* **Sample comments:** *'Yes, in a very constrained environment. Clearly assessed in the later assessments and in the later exam questions.'; 'That is essential. Assessed indirectly all the time. It is harder to do that explicitly.'; 'In the spreadsheets there is quite an aspect of that, but not in the more programming oriented sections.'* **Assessed** at this level: 6/7 modules.
Synthesis	**Questions:** *Is the ability to bring together diverse aspects of learning important in your module, and in first year computer science more generally? Does this impact upon your module? If yes, do you assess this directly in your module?* **Sample comments:** *'No. The module sticks to a very constrained domain.'; 'To a degree, e.g. in the section on finite state machines, but it is not central. There is a small attempt to assess it.'* **Assessment** at this level: 2/7 modules (both small components)
Evaluation	**Questions:** *Is the ability to evaluate and come to judgements in the light of material learned important in your module, and in first year computer science more generally? Does this impact upon your module? If yes, do you assess this directly in your module?* **Sample comments:** *'Assessed indirectly, because some of the problems will have easier or harder ways to do them. If they have learned to identify an easier route they will do better on the exam.'; 'No, not explicitly.'; 'Looking at it, but not assessing it directly.'* **Assessment** at this level: 1/7 modules.

Table 2: Interviews with course lecturers