

Using a Computing Ontology as a Foundation for Curriculum Development

Lillian N. Cassel¹, Gordon Davies², Richard LeBlanc³, Lawrence Snyder⁴, and Heikki Topi⁵

¹Department of Computing Sciences, Villanova University, Villanova PA 19085 USA
lillian.cassel@villanova.edu

²ACM Education Board gordon.davies7@virgin.net,

³Georgia Tech, Atlanta GA USA rich@cc.gatech.edu,

⁴University of Washington, Seattle, WA USA snyder@cs.washington.edu,

⁵Bentley College, Bentley MA USA htopi@bentley.edu

Abstract. An ontology of computing has been attempted with the intention of supporting, among other things, the development of curriculum. Computing is unusual in having 40 years of curriculum recommendations that have been very influential in determining what is taught in a large number of programs. The ontology offers an objective base on which to build a curriculum recommendation. The ontology is still evolving and the use in curriculum development is entering its first experiments. This paper describes the challenges of developing the ontology and the plans for its application in curriculum development.

1 Introduction

The goal of the Computing Ontology project has been to establish a common language for discussions and explorations across the breadth and depth of computing related domains. This presents opportunities within and among these disciplines for shared and collaborative efforts, and also presents a more cohesive and meaningful view of computing to other disciplines. Of particular interest is the possibility for an ontology of the computing disciplines to contribute to the development of programs of study.

Computing has a long history of providing curriculum recommendations, not as requirements, but as guidance for departments developing programs of study. From Curriculum 68 [1] through the multi-volume set that comprise Computing Curricula 2001 through 2005 [2-4] and through many comments in between reports, for example [5], the field of computing has grown and has split into subfields. The diversification is valuable and does lead to new developments, but also carries the risk of some difficulties if a coherent view of the combined disciplines.

With support from the US National Science Foundation, the ACM Education Board, and the IEEE-CS Educational Activities Board, an initial ontology of all of the computing disciplines has been developed. [6-9] The challenges have been substantial as each subdiscipline uses the same terms with sometimes subtle

differences in meaning. Groups also refer to the same concept with different terms. The intention has always been that the ontology would serve many purposes, including the classification of research works. In this paper, we focus on the applications of the ontology to education.

2 The Computing Ontology

In the most recent computing curriculum recommendations, five distinct fields earned the right to separate descriptions: Computer Engineering, Computer Science, Information Systems, Information Technology, and Software Engineering. A separate volume provided an overview and ways to distinguish among them. [3] At the same time, a benchmarking effort in the United Kingdom has identified more than 6000 different names for computing related programs of study. [10] It is clearly impractical to produce a separate curriculum recommendation for each possible variation on the study of computing.

Our approach has been to combine the bodies of knowledge from all of the recommendations along with the ACM Computing Classification System and other similar lists of computing topics and to organize them into a single ontology of the computing field. The result is accessible in hierarchical form for public view and comment at <http://what.csc.villanova.edu/twiki/bin/view/Main/OntologyProject>. The results of a preliminary visualization effort appear at <http://what.csc.villanova.edu/~harsha/flash/feb%2008/febversion6.swf>.

Table 1. Top Level Computing Ontology Concepts

Algorithms and Theory	Intelligent Systems
Computer and Network Systems	Mathematical Connections
Computing Education Research	Programming Fundamentals
Computer Graphics	Programming Languages
Computer Hardware Organization	Security
Discrete Structures	System and Project Management
Ethical and Social Concepts	Systems Development
History of Computing	User Interface
Information Topics	

Early discussions attempted to compress the breadth of the field into a tightly restricted set of top level concepts. These appear to be the following: Hardware, Software, Information, Human Interactions. After some considerable effort, we identified items from the topic lists that fit into every possible combination of those concepts – each one, each pair, each triple, and all four at once. We also realized that these were only part of the picture. In addition to finding a place among these high level concepts, each topic of interest could be viewed narrowly, for its own sake, or as part of a system. Each could be viewed from the perspective of a researcher, a designer, an applier, or a user of the system or the particular entity. Considerations of social impact and ethical use pervaded all. Because so few things could be classified exclusively in the four high level concepts, we abandoned that model for

the initial project and looked to existing classifications for guidance on the way to organize topics and subtopics. The resulting 17 top level concepts appear in Table 1.

2.1 Visualization of the Ontology

An early effort at visualization yields the image shown in Figure 1. Each red ball is an active link to the next level in the ontology. The goal is to support exploration of the entire space of computing and information related topics, showing their interconnections and relations. The Search box allows direct connection to an entry with a specified word, while clicking links allows browsing.



Fig. 1. Visualization of Top Level Concepts

This work is advancing well, but is far from complete. To illustrate the potential use of the ontology in computing curriculum, we rely on some hand drawn examples. Consider, for example, the concept of *testing* as shown in Figure 2.

Testing is an important part of the development of any software artifact and must be taught as students learn to write programs. Years of research and experience has led to knowledge about testing that is not familiar to every person who teaches programming. Observation of the related concepts allows proper planning for appropriate tools and exercises as well as sufficient time for this important topic. This comprehensive display of the topics related to testing can serve as a learning experience for a faculty member and as a method of evaluating the completeness of coverage of this topic.

The Computing Ontology does not specify any curriculum requirements. Rather, it provides a tool for faculty and students to use in determining how much of a given topic is appropriate in a particular context. The first experience with writing a small

program will not include such an extensive treatment of testing. However, a computer scientist or software engineer will expect to be comfortable with a substantial portion of the *testing* topic before graduation. The role of the ontology is to show the concept and its related concepts so that an informed choice can be made in any given situation. The faculty member will choose the parts of the topic that are meaningful to a given set of students and will make a conscious choice to include some parts of this domain and to exclude other parts.

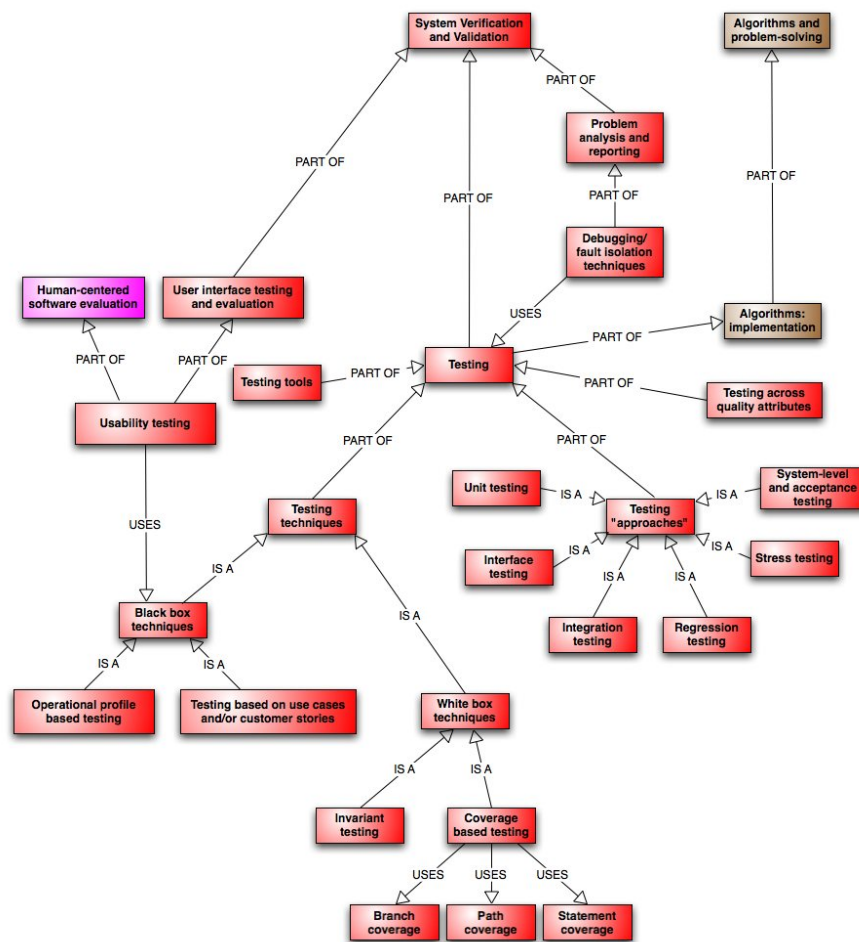


Fig. 2. Graphical representation of the concept of Testing in the Computing Ontology

2.2 Beyond a Single Topic

While there is value in seeing how a particular topic is related to others in the ontology and to be empowered to make a conscious choice of what to include or exclude from a class or course, there is a larger role for the computing ontology in computing education. The following section describes the complications involved in developing a computing related discipline and distinguishing it from others.

3 The Computing Ontology as a Tool for Curriculum Development

The number of separate variations on the theme of a computing curriculum continues to grow. There are two major influences on the content of these programs: Curriculum Recommendations made by the leading professional and scientific societies (ACM, IEEE-CS, AIS) and accreditation requirements from organizations such as the ABET Computing Accreditation Commission and, in the case of Software Engineering, the Engineering Accreditation Commission. In recent years, the approach to accreditation has moved away from set requirements to a specification of characteristics of graduates of programs. Individual departments are left to devise a way to develop a student to meet those expectations. There is a close interrelationship between the curriculum recommendations of the societies and the expectations of the accrediting agency, with the curriculum recommendations generally leading the accreditation expectations.

3.1 A More Responsive Curriculum Recommendation Cycle

In the past, computing curriculum recommendations have been produced by an expensive and cumbersome system involving very large committees and a great deal of time. In a field that grows and changes as fast as computing, it is not surprising that the recommendations are somewhat dated by the time they are published. An important role proposed for the computing ontology is to alleviate the current bottleneck in updating curriculum recommendations so that the process can be more responsive to the needs of a rapidly changing discipline. The new approach is currently under development and is expected to be a part of the next Computer Science recommendation. The general approach is represented in Figure 3.

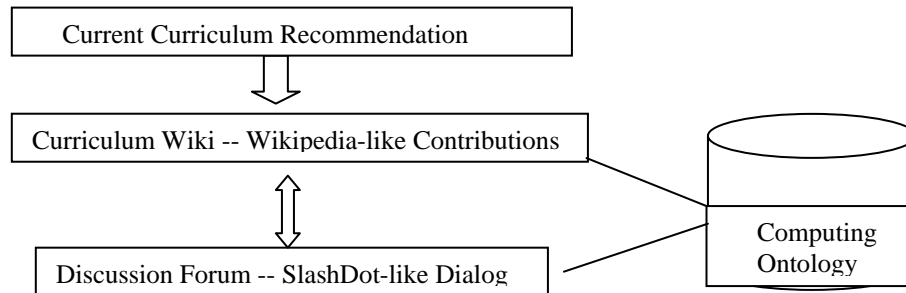


Fig. 3. An Ontology-based curriculum development scheme

There is a recognized need for a stable recommendation that forms the basis of department level curriculum development work. That appears in the scheme as the *current curriculum recommendation*. The wiki provides a way for the community to comment on the current recommendation, pointing out advances in the field that need to be addressed in the curriculum or areas that no longer are as critical as they were. These are informed by the ontology, which sits as an objective categorization of the entire domain of discourse. The ontology is separately edited and kept up to date and provides a continuously improving reference point for the curriculum discussions. Proposed changes appearing in the wiki will generate discussion, which is supported by the forum. A committee will oversee the wiki and the discussion and will note when the proposed changes or indications of problems have become sufficiently significant to trigger formal review and the production of an updated recommendation. Rather than the traditional ten-year cycles, the time frame for review and revision will be generated by community input and changes will occur as needed. The need for stability in the current recommendation will control the pace of major changes, but the community will always have access to the current thinking expressed in the wiki and discussion forum. In that way, those needing the most up-to-date ideas about likely curriculum changes will be well informed.

3.2 Ontology Connections to Outcomes-Based Curriculum

Figure 4 suggests another way in which the ontology can serve education. Imagine a system in which a drop down list of learning outcomes appears. When a particular outcome is selected, its related concepts in the ontology are highlighted. More than that, however; concepts related to those that are directly connected to the outcome are also displayed. The course or module developer now has a clear view of all that is involved in the topic at hand. The relationships are explored and a conscious choice is made about how far to follow the connections. Perhaps this is a minor topic in the module and only the most relevant concepts will be presented. Perhaps this is a key component of the program and the related concepts must be explored in depth. The developer sees what is involved from the beginning and makes choices and

appropriate plans for including as much of the related material as is appropriate for the needs in this specific case.

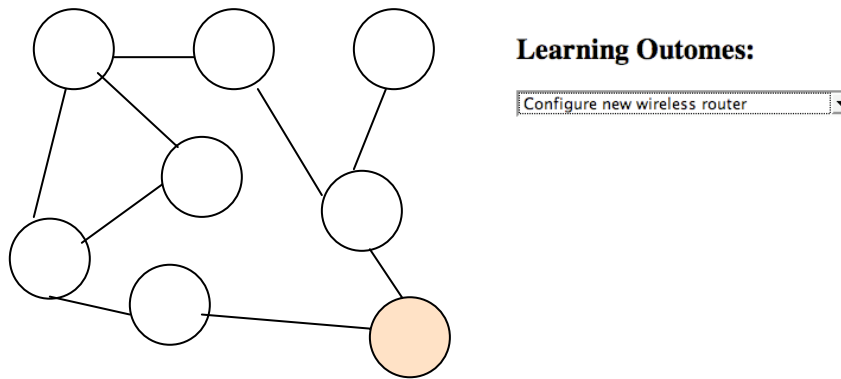


Fig. 4. Ontology-based learning outcomes

3.3 Distinguishing Programs of Study

With the large and growing number of computing programs, how can a student or employer know what each is about? There are several potential uses of the ontology for the purpose of describing and distinguishing among computing programs. One way is to look at the learning outcomes associated with a program. By selecting all the learning outcomes at the program level (as opposed to the module level illustrated in Figure 4), a person can see sections of the ontology highlighted and know that those sections characterize the goals of that program. By comparing the sections highlighted by several programs, the user can see the overlaps and the distinctions between programs.

Alternatively, a person might look at the curriculum recommendations and the sections of the ontology that are associated with each of those. Hopefully, the curriculum recommendations will also be based on expected outcomes. Now, the user can see the characteristics of those types of programs that have documented recommended curricula. This would allow a student or employer to see the overlaps and the differences between programs in computer science and software engineering, for example.

3.4 The Ontology in Digital Library Reference

Finally, a complete reference ontology for all of computing provides an ideal way to categorize the entries in a digital library. When one categorization scheme is used to

design curriculum and to classify resources, the connections between the two are immediate and obvious. Work is currently under way to import the computing ontology into the Computing and Information Technology Interactive Digital Education Library (CITIDEL – www.citidel.org) as the classification system for all entries. CITIDEL is implemented in DSpace and is part of the NSF's National Science Digital Library (NSDL – www.nsd.org).

4 Conclusions

The Computing Ontology project has a number of goals and the diversity of the intended uses complicates its development. However, a significant factor in its design and implementation has been the application to education. Implementations are at various stages of development, but a great deal of thought has been given to the use of a computing ontology for purposes as varied as indexing the content of a digital library to serving as a reference point for curriculum development. Effective visualization of the ontology is key to a number of its proposed uses and that remains a challenge.

References

1. W. F. Atchison, E. J. Schweppe, W. Viavant, D. M. Young, Jr., S. D. Conte, J. W. Hamblen, t. E. Hull, T. A. Keenan, W. B. Kehl, E. J. McCluskey, S. O. Navarro, and W. C. Rheinboldt: Curriculum 68: Recommendations for academic programs in computer science: a report of the ACM curriculum committee on computer science, Communications of the ACM, vol. 11, pp. 151-197 (1968)
2. Joint Task Force on Computing Curricula: Computing Curricula 2001, Journal of Educational Resources in Computing (JERIC), vol. 1 (2001)
3. R. Shackelford, L. Cassel, J. Cross, J. Impagliazzo, E. Lawson, R. LeBlanc, A. McGettrick, R. Sloan, and H. Topi: Computing curricula 2004: the overview project, in Proceedings of the 35th SIGCSE technical symposium on Computer science education. Norfolk, Virginia, USA: ACM (2004)
4. J. Gorgone, G. B. Davis, J. S. Valacich, H. Topi, D. L. Feinstein, and H. E. Longenecker: IS 2002 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems, vol. 2003: ACM AIS AITP (2002)
5. R. J. Dumas and A. E. Nieberding: An alternative to curriculum 68, presented at 6th SIGCSE Symposium on Computer Science Education (1976)
6. L. Cassel, R. Shackelford, and R. H. Sloan: A synthesis and ontology of all of computing, in Proceedings of the 36th SIGCSE technical symposium on Computer science education. St. Louis, Missouri, USA: ACM (2005)

7. L. Cassel, A. McGettrick, and R. H. Sloan: A comprehensive representation of the computing and information disciplines, in Proceedings of the 37th SIGCSE technical symposium on Computer science education. Houston, Texas, USA: ACM (2006)
8. G. Davies, L. N. Cassel, and H. Topi: Using a computing ontology for educational purposes, in Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education. Bologna, Italy: ACM (2006)
9. L. N. Cassel, R. H. Sloan, G. Davies, H. Topi, and A. McGettrick: The computing ontology project: the computing education application, in Proceedings of the 38th SIGCSE technical symposium on Computer science education. Covington, Kentucky, USA: ACM (2007)
10. Computing Benchmark Standard, vol. 2003: UK Quality Assurance Agency for Higher Education (2000)