A Cognitive-Based Approach to the Implementation of the Introductory Computer Science Programming Sequence

David D. Langan, Michael V. Doran, David L. Feinstein, Herbert E. Longenecker
University of South Alabama
School of Computer and Information Sciences
Mobile, Alabama 36688

ABSTRACT

A cognitive-based approach is being used to develop comprehensive materials for the first courses in Computer Science based on Implementation D of Computing Curricula 1991. The distinguishing features are: (1) materials based on a strategic sequencing and the associated Bloom level of mastery of key topics, (2) topical coverage carefully based on a spiral approach to information presentation, (3) integral use of structured labs as a necessary component of the course, (4) an emphasis on frequent feedback to facilitate learning and to evaluate the effectiveness of instruction, (5) an early use of teams, (6) a student surveying tool used to track all students to provide outcome assessment, and (7) review and evaluation by multiple institutions for iterative material refinement and national dissemination. A preliminary on-site evaluation, by a team of five consultants with expertise in the fields of computing, computing education and educational psychology, was conducted at the beginning of the project to critique project planning and initial materials. Materials currently developed will be class tested and evaluated by other faculty during the remainder of this year. These updated materials will be refined and further disseminated. The evaluation of materials will continue with the original five on-site consultants, three off-site consultants and several review/adoption institutions. An Undergraduate Faculty Enhancement workshop has been funded and is being planned for June of 1996. This will allow 20 participants to be exposed to the methodology and materials developed in this project.

INTRODUCTION

The ACM, IEEE and DPMA have over the years presented curricula guidelines[1,7,10]. These guidelines have been used by institutions to assure a quality education for computer and information science students. Compliance with these guidelines is often used by accrediting boards in evaluating the quality of a program.

Although the published curricula state suggested topical coverage and time allotments, the implementation is subject to a wide range of interpretations. The general nature of these suggestions allows each instructor the freedom to make different assumptions regarding student capabilities upon entering the course. Similarly, the targeted course behavioral objectives may differ substantially from one instructor to another. As a consequence of these differences between instructors, students will not be equally prepared by the same course or sequence. This problem is further aggravated by the use of part-time faculty or graduate students teaching the course. These instructors are often totally unfamiliar with the curriculum guidelines and are given vague or limited instructions on course objectives.

In addition to the problems associated with faculty implementation of these guidelines, there is also a problem with student understanding of course/curriculum intent. Few students in these first courses in the curriculum see the “big picture”. Students typically view courses as isolated learning experiences, fail to see topical relationships within a course, and are unsure of course behavioral objectives. Collectively these problems lead to students who are ill-prepared for upper level courses, who miss many fundamental concepts of the discipline, and who are confused and frustrated perhaps even to the point of leaving the discipline.

We assume knowledgeable and qualified instructors and sufficiently intelligent, well prepared students. It is the approach and methods employed to carry out this instruction that are in need of improvement. We believe that, when the process is better understood by both the instructor and the student, a more efficient and effective learning experience can be achieved. In this paper we define an approach to accomplish this improved understanding and we report on initial experiences with the use of that approach in the classroom.

CS CURRICULUM

First courses in any discipline are crucial. In Computer Science, students develop and apply problem-solving skills and learn elementary software engineering principles. These first courses, previously referred to as CS1 and CS2, were topically defined[11,12].

In 1991, the ACM/IEEE-CS Joint Curriculum Task Force released Computing Curricula 1991. Computing Curricula 1991 defined multiple options by proposing a suite of possible curricula for different audiences utilizing differing course content. The diversity they provided was to support a greater variety in types of institutions as well as in the types of computing programs.

The earlier curriculum guideline[1] focused on a course by course approach based on recommended topics. The 1991 guidelines took a different approach to describing a curriculum. This newer version is based on a more holistic view of the curriculum content. These guidelines include a collection of knowledge units. Each knowledge unit listed topics to be covered, total hours to devote to the knowledge unit, and prerequisites to the knowledge unit. The guidelines also include descriptions of several implementations intended for different institutional types (e.g., liberal arts, engineering). The implementations were also organized based on the presentation approach (e.g., breadth first, depth first). Within each implementation, courses were described according to a list of knowledge-unit fractions. Each knowledge-unit fraction identified the percentage of that knowledge-unit which should be covered in the course. For example, AL1 (Basic Data Structures) has 13 total hours of coverage in the curriculum and in an introductory course the knowledge-unit fraction “5 /13 AL1” would be used to indicate that one fourth of the total time spent on data structures should come from this course.

One of the curricula proposed, Implementation D: A Program in Computer Science, begins with two courses (Introduction to Computing I and II) that are quite similar in topical coverage to CS1 and CS2. It is likely, therefore, that many institutions already have in place courses similar to Computing I and II. Implementation D describes a traditional "depth first” approach for the introductory course sequence, as opposed to the "breadth first” approach described in Implementation E. An earlier NSF grant, led by Tucker, Turner and Barker, funded the development of materials for use in the first two years of Implementation E. The work of this project deals with Implementation D and complements the work of that earlier grant. Implementation D, by its closer proximity to many existing programs, is likely to be a popular target for many institutions. Thus, there are potentially many beneficiaries of this project.

APPLICATION OF PEDAGOGICAL TOOLS

A 1990 NSF workshop[13] advocated the application of educational techniques to scientific disciplines. The works of Bloom[3] in cognitive based learning and Gagné[9] in strategic sequencing have been successfully used in scientific disciplines[4]. This project applies these educational techniques to introductory Computer Science courses. A major distinguishing attribute of the proposed approach is its application of: (a) traditional cognitive-based learning objectives, and (b) strategic sequencing and blending to the domain of introductory programming courses.

Bloom's Taxonomy

Bloom characterized the educational process in terms of knowledge levels. The following table briefly describes Bloom's six levels of learning.

Level 1:	*Knowledge*: Students can recite, recognize and differentiate facts on a given subject.
Level 2:	*Comprehension*: Given cues, students can paraphrase, translate, interpret, extrapolate, and otherwise use facts.
Level 3:	*Application*: Without cues, students can appropriately apply facts to solve problems in new situations.
Level 4:	*Analysis*: Students can define the relevant components of new abstractions.
Level 5:	*Synthesis*: Students can synthesize the organization, development, and appropriate usage of new abstractions.
Level 6:	*Judgment*: Students can evaluate the effectiveness and efficiency of alternate syntheses.

Bloom proposes that an expected level of mastery be identified by behavioral learning objectives for each topic and that metrics be created for outcome assessment based on the behavioral learning objectives. This assessment is used to assure that behavioral learning objectives have been achieved.

As a part of this project, we mapped specific topics to target Bloom knowledge levels. Reaching a given knowledge level requires achieving all lower levels of knowledge first. Two issues addressed during the project development to assure success were:

1. what ordered activities are needed to facilitate the student's growth through these levels of knowledge and

2. how will instructors know that required levels of knowledge are being achieved.

Level 1, recognition and differentiation of facts, is achieved by rehearsal of the basic facts and concepts. This level of knowledge can be reached through assigned readings and classroom work. However, recitation sessions are important to ensuring mastery of the factual knowledge. Level 2 relies upon an explicit re-enforcement of the facts and use of the facts in a problem solving setting. Level 3, "application knowledge", relies on the mastery of previous levels. Level 3 requires definition and discussion skills and the kind of trial-and-error experiences gained through extensive practice in a laboratory. Students will be guided in the use of the facts to produce solutions. Both "closed" structured labs and "open" programming labs are extremely critical to the success of reaching targeted Level 3 knowledge. Structured labs promote open discussion of the ideas and provide immediate solutions to problems that otherwise would be put off until later or simply forgotten. These labs allow for the timely comparison of algorithms and code solutions. Thus, labs help to accomplish Level 2 and 3 objectives.

The target levels for students in Computing I and II range from Level 1 to Level 3. For some sub-objectives, it is reasonable to expect students to achieve Level 3 knowledge in these introductory courses.

Strategic Sequencing

Gagné describes a system's view of prerequisite knowledge[9]. In this description, he states that successful completion of each topic lays a necessary foundation for subsequent topics. In the educational context, a "system" may be viewed as those events organized to enhance the knowledge, understanding, or skills of a student. The result of one system serves as input to the following system. In the educational process we see this output/input relationship manifested in course prerequisites. A clear and explicit set of system goals is required if one hopes to design a system that can achieve the desired transformations[5,14].

Learning occurs by the execution of a well-planned sequence (a "strategic sequence") of subsystems. Each subsystem is designed to accomplish an incremental goal, ultimately leading to the attainment of the overall course goals[6]. Research applying the concept of general system theory to the specialized problem of instructional design has been conducted[8,9,15]. In order to control progress through a strategically sequenced collection of subsystems, feedback is required[2]. In educational systems, the feedback is obtained by comparing the actual progress with the expected progress towards the system goals.

Bloom's taxonomy will serve as the basis for the metrics used in evaluation. System theory applied to the introductory sequence dictates that (a) each course must internally meet a topical prerequisite structure and (b) must have a well defined exit expectation which serves as the entry level knowledge for the next course.

PROJECT IMPLEMENTATION

In applying the educational techniques previously identified to address the problems cited, we identified and developed topical modules. Each module is defined in terms of information from parts of several knowledge units (KU's) from the CC91 document. The component parts of each module are:

(a) An Overview**.** This is handed out to the students and used by the instructor as a guide. It consists of a brief description of the module goals and overview. The list of topics will consist of both the high level topics found in the CC91 KU's, as well as the lower level detailed sub-topics generated by our interpretation of the CC91 document. Also included is an identification of appropriate reading materials.

(b) A Rationale**.** This is the underlying discussion of the goal and focus of the module, based on the CC91 document. It describes how we decided to include and organize these topics into the course activities. This is only used by the instructor, but must be communicated to the students via the lecture, lab and other related activities.

(c) A Script. This is a detailed presentation of the items associated with the topics and micro-objectives identified. This is only used by instructor.

(d) A Tentative Schedule. This shows both instructors and students the expected pace of the specified topics.

(e) A Set of Behavioral Objectives. This will list the description of outcomes which students must be able to demonstrate. It will be described in terms of topics and micro-objectives mapped to Bloom levels. Sample questions to measure achievement of the level will be provided. These sample questions could be used by students to practice for quizzes, tests and homework assignments. These materials will be provided to the students as part of a workbook to track their activities and monitor their performance.

(f) Assigned Homework. These assignments are in support of the state behavioral objectives.

(g) A Glossary. To assure consistency of vocabulary and the identification of key terms each module will have an associated glossary. This will help to emphasize which words are important. It will also form the basis for communicating Bloom level one factual knowledge.

(h) A Daily Log. This is an after the fact recording of actual classroom experiences. It serves to document for future instructors what worked and what didn't and to critique the experience.

(i) An Online Module Test. This test helps students to check their understanding (at Bloom level 1) of the basic facts, prior to a module quiz or test.

PROJECT PHASES

Evaluate CC91

The initial phase of this project involved project planning and careful evaluation of the ACM/IEEE 1991 curriculum guidelines. The flexibility inherent in the CC91 guidelines allowed it to be used to meet a variety of program needs, but it also could potentially lead to the problems of discontinuity mentioned earlier. It was also observed that using knowledge-unit fractions for a course description was meaningful when considering an entire curriculum, but was too vague and inconsistent when developing specific courses. Additional problems were encountered with the prerequisite structure provided in the 1991 guidelines. The guidelines listed which knowledge units were prerequisite to other knowledge units. For example, they said "KU_X is prerequisite to KU_Y", and then describe KU_X with 10 topics and KU_Y with 12 topics. This does not identify which of the topics from KU_X were needed before which topics from KU_Y. This also fails to identify when certain topics within a knowledge unit might prove to be prerequisite for others topics within the same knowledge unit. Several members of the team that developed the 1991 curriculum guidelines stated that the document was intended as a broad guideline and that it relied on the proper interpretation by the faculty who used the guidelines. Textbook authors often make similar assumptions about the usage of their materials. These assumptions can be problematic if the faculty assigned to a course are not adequately familiar with course intent.

Design and Evaluation

During the planning phase of this project we identified the appropriate topics found in the knowledge units for each of the courses of our introductory sequence. We selected and grouped the topics for each course and sequenced their presentation into modules. We checked the topics of the various modules against the recommended time coverage as proposed in the curriculum guidelines. Due to the “flexibility” built into the guidelines we acknowledge that ours is only one possible implementation based on those guidelines. Most of our decision making process was driven by the nature of our institution and student population. There was no intent to redefine the course curriculum and efforts were expended to assure completeness of content coverage and consistency with the 1991 guidelines.

To help document the process of relating knowledge units to our presentation modules, we took several of the knowledge units from the guidelines and decomposed the selected topics into micro-objectives. For each micro-objective we identified a targeted Bloom level of knowledge and listed behavioral objectives that would demonstrate mastery of the micro-objective at the targeted Bloom level. Next, we created sample activities that would help to achieve and measure mastery at these targeted Bloom levels. These materials were sent to the consultants for review prior to an on-site group evaluation. This group, consisting of L. Bellamy, H. Takacs, E. Koffman, and J. C. Little, concurred with the major themes of the project and suggested that we focus our development efforts in the area of creating instructional materials.

Implementation and Evaluation

The next phase of project involved the initial implementation of the first course of the introductory sequence. During this phase we generated materials that were consistent with the approved prototype modules.

At the end of the first course, course evaluation took place using several different evaluation instruments. The results of these surveys are being analyzed to identify the degree of satisfaction and benefit gained by the use of the course materials. Further evaluation of this initial implementation is also underway in the form of external reviewers. The course materials are being externally critiqued by both the project consultants and a team of reviewers at other universities. Within our university, the materials for that course are now in use by other faculty members who are providing insights regarding their use. The feedback from all sources will be used to direct future efforts to revise the existing materials and to create similar materials for subsequent courses.

Dissemination

A UFE summer workshop to disseminate the materials and the approach used is scheduled for June of 1996. The materials developed in this project will serve as an example of the methodology being proposed to the workshop participants. It is the goal of the workshop that the participants will apply the methodology to courses at their institution.

CONCLUSIONS

During the Fall quarter of 1995 the methodology proposed here was applied to generate materials for the first course in the introductory sequence. This effort was substantially aided by a new networked PC lab dedicated to this course sequence. This new lab facilitated an integration of teaching and experimentation (which is integral to the approach advocated).

The faculty working on this project have been involved in the introductory sequence for several years. This past experience forms the basis for comparison to the methodology currently being evaluated. Our major observations from this experience follow:

- With explicit expectations, students who worked diligently were very successful.

- Most students in the survey expressed appreciation for course materials, though a few students (generally the weaker ones) found the copious notes and handouts to be overwhelming.

- Few instruction-related excuses were offered for any lack of individual success.
- Students credited explicit expectations for a greater degree of confidence in their knowledge.

- With an explicit set of behavioral objectives for the course, the skill levels of students from various sections entering the second course were less disparate.

- Students found less need for the textbook. We speculate that this may have been related to our explicit statement of expectations. With such a statement, there was no need for them to figure out from the text what was important.

- Retention rate remained stable.

- Developing and implementing a course using the proposed methodology requires a substantial effort, but then the materials are re-usable and maintenance is minimal..

ACKNOWLEDGEMENT

This project is partially supported by: NSF-CCD grant DUE-9455522, June 1, 1995 - May 31, 1996

REFERENCES

(1) ACM Curriculum Committee on Computer Science, "Curriculum '78 - Recommendations for the Undergraduate Program in Computer Science", *Communications of the ACM*, 22(3):147-166, March 1979.

(2) Niv Atihuv and Seev Neumann, Principles of Information Systems for Management, 3rd edition, William C. Brown, Dubuque Iowa, 1989.

(3) Benjamin S. Bloom et al, The Taxonomy of Educational Objectives: The Classification of the Educational Goals. Handbook I: The Cognitive Domain. McKay Press, New York 1956.

(4) Paul Chance, "Master of Mastery", *Psychology Today*, April 1987, pp 43-46.

(5) C. West Churchman, Systems, Dell Press, New York, 1968.

(6) M. Doran, H. E. Longenecker Jr. and J. H. Pardue , "A Systems Approach to a Data Structures Course for

Information Systems Students Consistent With DPMA IS'90, ISECON'94, October 28-30, 1994, Louisville, KY.

(7) Herbert Longenecker and David Feinstein (eds.), Information Systems: The DPMA Model Curriculum for a Four Year Undergraduate Degree, IS '90, DPMA, Park Ridge, Illinois.

(8) Robert M. Gagné, The Conditions of Learning, Holt, Rinehart and Winston, New York, 1985.

(9) Robert M. Gagné, Leslie J. Briggs and Walter W. Wager, Principles of Instructional Design, Holt Rinehart and Winston, New York 1988.
(10) Computing Curricula 1991 - Report of the ACM/IEEE-CS Joint Curriculum Task Force.

(11) Elliot P. Koffman, Phillip L. Miller, and Caroline E. Wardle, "Recommended Curriculum for CS1: 1984 a Report of the ACM Curriculum Task Force for CS1", *Communications of the ACM*, 27(10):998-1001, October 1984.

(12) Elliot P. Koffman, David Stemple, and Caroline E. Wardle, "Recommended Curriculum for CS2: 1984 a Report of the ACM Curriculum Task Force for CS2", *Communications of the ACM*, 28(8):815-818, August 1985.

(13) Report of the National Science Foundation Workshop on the Dissemination and Transfer of Innovation in Science, Mathematics, and Engineering Education, Dr. Richard D. Breslin (chair), Directorate for Education and Human Resources, NSF, May 1990.

(14) Rita Richey, The Theoretical and Conceptual Bases of Instructional Design, Nichols Publishing, New York 1986.

(15) David F. Salisbury, "What Should Instructional Designers Know About General Systems Theory?", *Educational Technology*, August 1989, pp 42-45.

BIOGRAPHICAL INFORMATION

DAVID D. LANGAN is an associate professor of Computer Science at the University of South Alabama in Mobile, Alabama. He earned a Ph.D. from the University of Southwestern Louisiana in Computer Science in 1988. Areas of interest include: programming languages, parallel processing, and computer science education.

MICHAEL V. DORAN is an associate professor of Computer Science at the University of South Alabama in Mobile, Alabama. He earned a Ph.D. from Tulane University in Computer Science in 1989. Areas of interest include: programming languages, artificial intelligence, and computer science education.

DAVID L. FEINSTEIN is professor and chair of Computer Science at the University of South Alabama in Mobile, Alabama. He earned a Ph.D. from Stanford University. He has been active in curriculum development efforts for DPMA, ACM, IEEE, and AIS. He is also a chair of a CSAB review team.

HERBERT E. LONGENECKER is an associate professor of Computer Science at the University of South Alabama in Mobile, Alabama. He earned a Ph.D. from Rockefeller University. Areas of interest include: information systems, system theory, and computer science/information science education. He has been active in curriculum development efforts for DPMA, ACM, IEEE, and AIS.