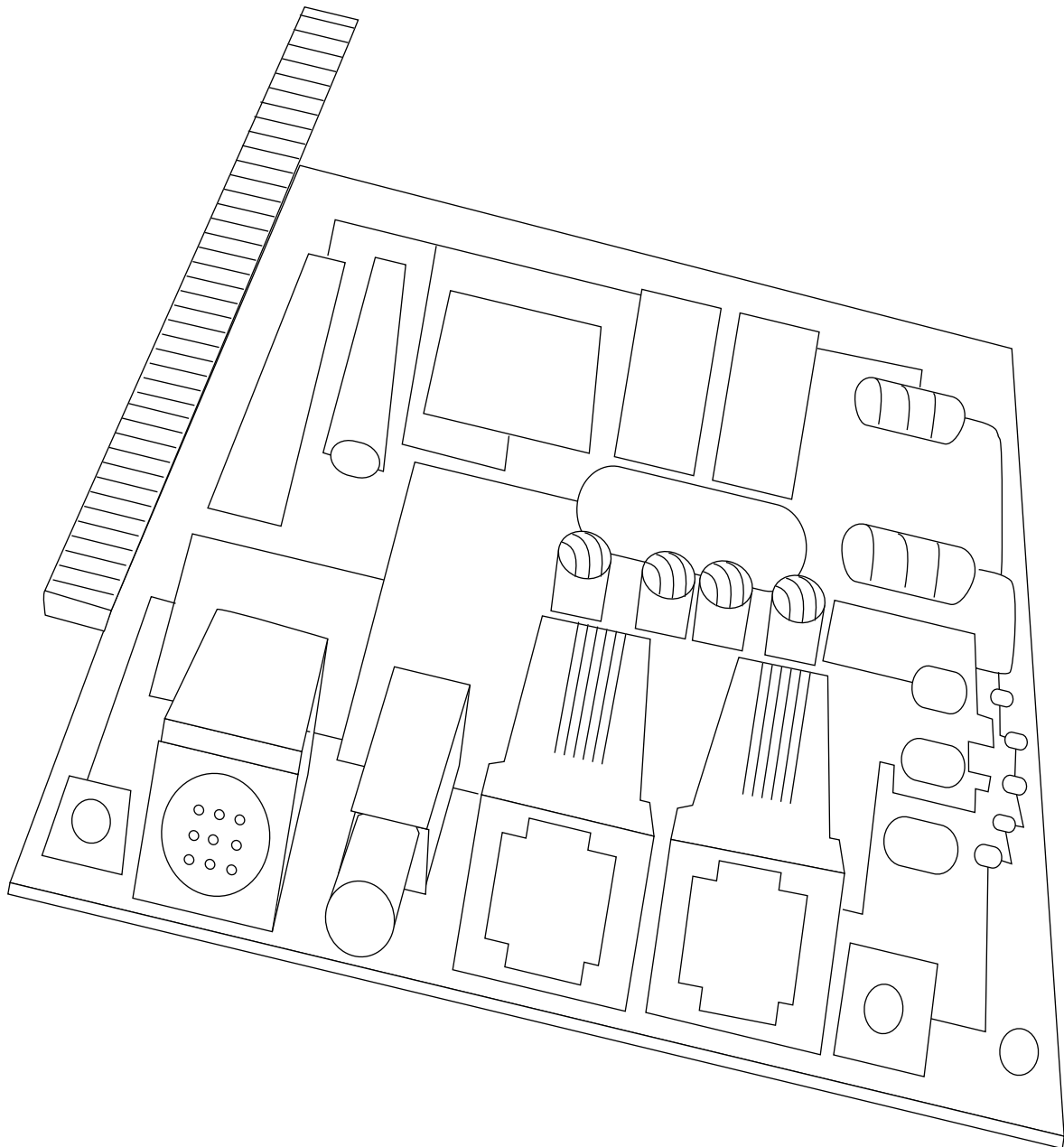


CLUSTER II

Science

Computer Science

Prepared by Don Warner, Lead Discipline Faculty, CSU Sacramento



Summary of Identified Issues

Areas of concern:

1. The articulation of the lower-division program in Computer Science would be best served by adopting the ACM/IEEE Curriculum 2001 (see www.acm.org/sigcse/cc2001/cc2001.pdf) document approach. This document describes a Computer Science curriculum based on knowledge units. The core knowledge units can be taught in any number of course sequences. There are eight (8) different course sequences described in the ACM/IEEE document, each of which covers these core knowledge units. By adopting this approach, we would allow students to transfer from one institution to another without having to articulate on a course-by-course basis, assuming students complete the sequence of courses at the institution in which they began the program.
2. There is a need for the development of core topics to be covered in a discrete mathematics course.
3. The question of including a linear algebra requirement in the major remains.
4. The current Computer Science CAN descriptors are inadequate and need revision.
5. Faculty are concerned with how to further promote the articulation program for lower-division Computer Science curriculum without creating a comprehensive course-by-course articulation system.

Identified Trends/Future Directions:

1. Further develop the knowledge unit approach to articulate the lower-division program of Computer Science.
2. Include a Computer Science service course in the proposed science curriculum (See Biology Appendix 2 for information on the SciGETC Proposal).
3. Provide more meaningful CAN descriptors for Computer Science.
4. Recruit a group of participants to meet with faculty and staff at other institutions to further explore the articulation of the lower-division program based on the IEEE/ACM CC 2001 approach.

Comments from Statewide Meetings and the General Field

Participants at all meetings provided significant ideas and insights regarding the articulation problem as well as the inadequacy of the CAN system.

1. CAN descriptors are not adequate to fully articulate Computer Science courses. If CAN descriptors are more specific, faculty and staff in the UC system will be more willing to use CAN as a course numbering system. Four-year institutions need course outlines/syllabi/detailed descriptions in order to make articulation judgments in the best interest of the students. These descriptions and documents would also need to include expected learning outcomes, core understandings, or common experiences.
2. If UC/CSU faculty were to provide perceived content/skills required for success in the upper-division work on their campuses (couching these suggestions in the nomenclature of the IEEE/ACM CC2001), the community colleges could design courses to fully meet the requirements of four-year programs at neighboring campuses and elsewhere. Content articulation seems preferable to course articulation. A preparatory program of up to two

years is more important than any specific course. The description of requirements for transfer should probably include preparation in math and verbal skills (these might be addressed by course descriptions since the content is better understood).

3. Course-by-course articulation, as it exists between some institutions already, would continue between community colleges and nearby four-year institutions.
4. The IEEE/ACM course content and outcome expectations are more useful than course descriptions as predictors of the success of transfer students. However, no one-page course description can provide all the necessary information for judging the preparation of a transfer student.
5. Participants proposed a strategy for developing CAN descriptors. First, the strategy suggests that faculty should look at the ACM curriculum and develop the threshold of knowledge that must be gained; this is the common goal of the differing set of approaches outlined in the ACM document. Then, a CAN number and descriptor that represents the total sequence should be developed. A key point: don't be concerned about the order of the sequence, as long as the total complex of courses covers all the material and all of the learning objectives are met. The CAN "CS number" would be a sequence number (similar to physics) and the course outlines collectively would be reviewed to verify content. Editorial Note: This is in distinction to the present concept of having 1 course = 1 CAN number.

Recommendations for the Discipline

1. The Computer Science IMPAC working group recommends that the Computer Science faculty of CC, CSU, and the UC Systems address articulation agreements by adopting their agreements based on the knowledge units and topics of the 2001 ACM/IEEE document.
2. At future meetings, participants should develop a method to implement the recommendation stated in suggestions 1 above.
3. Four-year institutions should make a more concerted effort to keep in close contact with community colleges in their regions and inform them of any proposed changes in curriculum as early as possible in order to keep articulation agreements up-to-date.
4. Conduct regional articulation meetings with local community colleges.
5. Conduct additional regional meetings with four-year institutions to further promote the articulation "program" based upon the IEEE/ACM CC 2001 approach.
6. Develop a detailed program CAN descriptor as well as course-by-course descriptors of Computer Science course sequences based on the knowledge unit approach of the IEEE/ACM CC2001 document. (See Computer Science Appendix 1 for a sample proposed CAN descriptor for Computer Science).
7. Develop a website to display course outlines for all institutions.
8. Prepare a database couched in the nomenclature and language of the ACM/IEEE CC2001 listing those items (content and skills) needed by transfer students that would ensure they could compete successfully with native students on four-year campuses. Ideally, this would include each of the topics/concepts in the core curriculum as well as the learning objectives under each topic. At a minimum, it should also be updated annually.

Recommendation for the Support Courses

Develop linear algebra requirements for Computer Science.

Topics for Further Discussion

1. Continue to obtain additional information from the four-year institutions regarding their lower-division program requirements, and data on how they relate to the IEEE/ACM CC2001 document. (See results of the survey conducted in 2002-2003 in Computer Science Appendix 2).
2. Define the CAN descriptor outline to be used to describe a Computer Science lower-division program and courses.
3. Discuss the use of additional science disciplines in meeting the science requirement for Computer Science.
4. Discuss the potential problem of a student failing to complete a course sequence at one institution and transferring to another institution that offers a different sequence of courses for the lower-division program.

Recommendations Forwarded/ to be Forwarded to:

CAN

1. Develop more detailed descriptors for Computer Science courses.
2. Develop a descriptor for the lower-division program of Computer Science

ASSIST

Present a “program” articulation course that reflects the requirements of the lower-division requirements for computer science.

Outreach Presentations made by members of this group:

Organization	Date/Place	Presenters Names	Number Present
CIS/CS Chairs of the CSU	05/02/2003 Sonoma State	Steven Stepanek Don Warner Keith Jolley	12

SUGGESTED CAN DESCRIPTOR OUTLINE

Proposed CAN Descriptors for Computer Science

Course Number:

Course Title:

Course Description: Written in catalog style, highlighting the major topics and general expectations of the course.

Prerequisites: Required courses, units or background

Syllabus: Bulleted list providing an outline of the topics covered.

Units Covered: List of units covered as defined in the CS body of knowledge

Outcomes:

Basic Understanding of:

List of Topics

Through Understanding of:

List of Topics

Exposure To:

List of Topics

Notes: Optional narrative section offering additional explanatory notes about the course. These may include goals, pedagogical suggestions, and assessment strategies.

SURVEY RESULTS OF FOUR-YEAR INSTITUTIONS

Survey of Computer Science Programs at Four-Year Public Institutions in the State of California: Administered during the 2002-2003 IMPAC Year

This survey is an effort to determine an alternative for transfer students majoring in Computer Science from community colleges and four-year institutions to the CSU and UC systems. This is an effort organized by participants in the Intersegmental Major Preparation Articulated Program (IMPAC), which is supported by the Academic Senates of the California Community College, University of California, and California State University systems.

Participants in the regional and statewide meetings have expressed the desire to articulate the lower-division portion of the CS programs using the ACM/IEEE Curriculum 2001 approach of core topics of the defined Computer Science body of knowledge. The topics common to all of these course sequences are the Computer Science programming, data structures, and discrete structures courses.

The ACM/IEEE curriculum document presents eight approaches to organizing the lower-division curriculum, including: the introductory programming sequence, elementary data structures, and discrete structures. The approaches are: Imperative First Sequence 1, Imperative First Sequence 2, Objects First Sequence 1, Objects First Sequence 2, Functional First, Breadth First, Algorithms First, and Hardware First. These sequences are described in the ACM/IEEE curriculum 2001 document in Chapters 5-8. Also, notice that there are columns for the two sequences of discrete structures for Computer Science on the attached spreadsheet.

The attached Excel document summarizes the number of classroom or instructional hours that should be associated with each of the core topics covered in the course sequences identified above.

Please place a 'Y' in the last column of the spreadsheet if you feel that the topic is covered in your lower-division major program. If you can provide an estimate of the number of hours of instruction devoted to each topic, please provide that information as well.

In addition, please indicate the lower-division mathematics and science requirements of your program.

Please send the results to me so that I can compile them. You can email your completed survey to me at warnerd@ecs.csus.edu. You may also send it to:

Don Warner, Chair
 Department of Computer Science
 California State University
 6000 J Street, Campus Zip 6021, Sacramento, CA 95819

Thank you in advance

Survey Results as of 03/21/2003

CSU, Northridge	Implemented
CSU, Sacramento	Reviewing
CSU, San Bernardino	Reviewing
Humboldt	Implemented
CSU, Bakersfield	Implemented
UC Irvine	Implemented

The results here means that some four year institutions are using or are considering aligning their program with the recommendations contained in the ACM/IEEE Curriculum 2001 as a template for their lower division program

Next year a document encompassing the basic requirements of computer science programs of the four-year institutions as defined in the IMPAC and other subsequent meetings and/or reviews will be developed. This document will provide the basic knowledge units expected of all lower division programs as well as other desirable units. This document will be distributed to all computer science programs of the four-year institutions as well as all community colleges for their comments and further input.