Summary of Identified Issues

Several key issues emerged. It is interesting to not that an almost identical list of issues came up at all four regional meetings. They are discussed below.

Topics to be Covered in the Calculus Series

Faculty were fairly united in the belief that most important outcome from taking the calculus series would be mental discipline and mathematical/logical reasoning. Performance in calculus is one of the best predictors of performance as a Computer Science major. Almost all four-year faculty agreed that computer science majors should take the series designed for engineering majors.

Topics to be Covered in Discrete Mathematics

There was less congruence in terms of topical coverage in the discrete math course. Boolean algebra and proofs were mentioned the most frequently.

Topics to be Covered in the Physics Series

There was relatively little agreement on topics. The faculty agreed that computer science majors should take the same physics taken by science and engineering majors (possible to substitute chemistry). Four-year faculty agreed that it should be calculus-based. Two-year faculty were less concerned about a calculus prerequisite. It was felt that exposure to rigorous scientific thinking was more important that specific topics, although electricity, etc. may be more relevant topics for computer science majors.

Skills Required to do Well in a Computer Science Program

As mentioned above, the mental discipline required to succeed in a rigorous calculus series and a rigorous science series is considered a good predictor of success in a computer science program. Study skills, English fluency, logical reasoning and abstraction were frequently mentioned as other requirements for success.

Prerequisites/Course Sequencing

There was a great deal of variation between the three systems as to course sequencing and prerequisites. Part of the problem arises because many of the computer-related coursework taught at the community colleges is taught to an audience much broader than computer science majors.

Baseline Curriculum

The "typical" lower-division course pattern for computer science majors was treated at every meeting. It was somewhat surprising to the facilitator how much overlap there was at the lower-division level, despite concerns about rigor and coverage in those courses. Faculty at each of the regional meetings independently came up with a very similar "baseline" lower-division curriculum to be recommended to computer science majors.

Identified Trends/Future Directions

The faculty identified several trends and future directions in computer science.

The Language Problem

There was a good deal of pessimism that the problem of different programming languages could ever be "solved." It would be very difficult to get different programs with different emphases to standardize on a language for Computer Science 1 and Computer Science 2 at one point in time, much less to get programs to change that language in unison.

Evolving Nature of Computer Science

It was agreed that since computer science is such a dynamic field, whatever comes out of these efforts will require that future monitoring, discussion and revision would constantly be needed.

CAN

No computer science courses are currently CANned, including Discrete mathematics. Future groups may wish to address this issue.

Comments from Statewide Meetings and the General Field

Curricular Changes without Adequate Notice to Community Colleges

Many community college faculty indicated that insufficient articulation was taking place with four-year campuses. Computer science faculty initiate needed changes but often do not fully appreciate the amount of lead time necessary to implement changes at the community college level.

Importance of Rigor in Lower-Division Coursework

A fairly common concern is that transfer students have not been exposed to the same level of rigor in their lower-division coursework that native four-year students have been. Many faculty felt that the best indicator of success in the computer science major was performance in a rigorous physics and calculus sequence. Most felt that these disciplines helped develop the critical thinking and reasoning skills essential for computer science students.

Difficulty Community Colleges have Offering Several Levels of a Similar Course, Even Though the Objectives of the Students May Vary Widely

Most community colleges are capable of offering rigorous sections of Computer Science 1, Computer Science 2, etc. However, most of the students taking computer science courses at this level are not intending to transfer to four-year computer science programs. They may be seeking a certificate, an A.S., or to transfer to a Computer Information System program. Only a minority ever transfer to a computer science or a computer engineering program. It is likely economically infeasible to offer sections just geared to computer science or computer engineering majors.

Confusion of CIS, CS, CE Curricula by Community College Students

Many students begin at community colleges with an interest in computing, with no firm idea of what type of program they eventually will seek to transfer to, if indeed they do transfer. This makes the problem of generalist computer science courses that may be less than optimal for computer science transfers even more difficult to overcome. Faculty are put in a very difficult position trying to gear coursework for students while only being able to offer a limited number of sections.

Shortage of Instructors

Given that the starting salaries of some students who have only a certificate are sometimes above those of instructors, it is becoming increasingly difficult to attract and retain computer science faculty at the community colleges. The problem is mirrored at the CSU campuses and to a lesser extent at UC campuses.

Need for "Remediation/Bridge Courses" upon Transfer to a UC Campus

Experience has shown at many four-years that even students who have done relatively well in articulated lower-division coursework are not sufficiently prepared upon transfer to be successful in upper-division coursework. Much of this situation can be attributed to use of different programming languages at different campuses, but clearly not all of it. Computer Science 1, Computer Science 2, etc. taught at the community colleges often do not cover as many topics as those at the four-years. Several campuses have addressed this problem by developing "bridge courses" for transfer students. Most of these students "catch up" fairly well by taking these courses even though, or perhaps because, many topics may be repeated.

Use of Different Programming Languages Across Campuses

Everyone recognized this as a problem. No one had a proposal to overcome it.

Need for Continuous Intersegmental Meetings

The meetings were seen as very valuable by faculty from all segments. Given the issues identified and the dynamic nature of computer science, it was felt that these meetings should somehow become institutionalized.

Students should take Calculus for Engineers and Scientists and Majors' Physics

These sequences are good training for the mind. They should be rigorous. The mental discipline, critical thinking, etc. required are essential for success upon transfer.

Different Programming Language Standards Cause Serious Problems for Transfer Students

This came up again and again. No approaches to this problem would seem too forthcoming given the diverse nature of programs and faculty.

CSAB/IBET Accreditation Standards and the ACM/IEEE Curriculum Guidelines

These standards can serve as the departure point for discussion of curriculum, skills sets, etc. They can be very helpful in efforts such as IMPAC.

CS Program Impaction

Many computer science programs are impacted. Not all four-year or two-year programs have an incentive to increase the number of majors, making facilitating transfer seem perhaps less important to faculty.

Computer Science is Not a Four-year Program

Native student routinely require five years or more. Community college students with weak math skills and job commitments may be looking at 7 or 8 years.

Some Students at Community Colleges May Need three to four-years of Coursework (especially mathematics) Prior to Transfer

Confusion of Computer-Related Disciplines and Programs

There is confusion among many new students as to their ultimate goal within the field of computing. Some may desire certification, some belong in CIS programs, a minority of those without clear direction will end up as transfer computer science majors.

Lower-Division/Upper-Division Coursework

There is some variation in the designation of lower/upper-division coursework. Several courses and topics were treated differently across the four-year campuses.

Recommendations for the Discipline

- Faculty at four-year institutions should post current course syllabuses on the web to allow for timely notification of changes in content/approach/texts.
- Faculty at two-year institutions should work to make sure the courses recommended to transfer students are appropriate to prepare them for the appropriate four-year program (CIS, CS, etc.). Computer Information Systems and Computer Science programs differ so widely in approach and required coursework that students not taking the "appropriate" courses will suffer from unnecessary and/or insufficient preparation. For example, a computer science transfer student who took computer information system foundation coursework would likely be delayed more than a year due to math and science coursework.
- An intersegmental computer science curriculum group should be established to keep the dialogue on lower-division curriculum going after IMPAC runs its course. Curriculum and associated issues will continue to change and evolve.
- Communication between community colleges and UC campuses should be increased. There are currently very few formal or informal lines of communication between faculty. IMPAC has provided a much needed avenue for discussion. Perhaps some agency could provide funding for service-area community colleges and UC campuses to get their faculty together for transfer and articulation discussions on a periodic basis.
- Future groups should determine which courses should apply for CAN status and should review and recommend any existing descriptors as well.

Recommendations for Support Courses

- The one-year physics series should be calculus-based and have a laboratory (the same applies to chemistry, if selected as an option). Computer

science majors should take the same sequence as science and engineering majors. This point was stressed by almost all four-year computer science faculty and most community college computer science faculty. The physics faculty were all in agreement.

- Computer science majors should take the same calculus series as science and engineering majors. There was a fairly strong consensus on this point across computer science and math faculty.
- The discrete mathematics course should contain: functions, relations, and sets; basic logic (including Boolean algebra and 1st order predicate calculus); proof techniques (including proof by construction, proof by induction and proof by contradiction); the basics of counting; graphs and trees; and discrete probability. The math faculty were very solicitous in asking for topics and approaches that would be desirable. It is unclear how much of the input from the computer science faculty will actually be implemented in new/revised sections of discrete math courses taught through math departments.

Topics for Further Discussion

We recommend that next year's group take up the following tasks:

1. Circulate the recommended baseline curriculum as broadly as possible and solicit feedback.
2. Come to closure on the baseline curriculum.
3. Work with ICAS to form an Intersegmental computer science Curriculum Advisory Committee to keep this process going and any recommendations up-to-date.
4. Explore the possibility of developing CAN descriptors for Computer Science 1, Computer Science 2, Machine Architecture/Assembly Language, and Discrete Mathematics (or whatever the final configuration of baseline transfer courses looks like).
5. Keep the dialogue open between systems and try to formalize contacts on a regional basis.

Recommendations Forwarded/to be Forwarded to:

CAN: Work with the discipline to explore possible CAN descriptors for Computer Science 1, Computer Science 2, Machine Architecture/Assembly Language, and Discrete Mathematics.

ASSIST: Develop a report of the current state of articulation in computer science across four-year institutions and distribute it to next year's group.

CSU CSIS Chairs Council: Review the baseline curriculum and get comments to the lead discipline faculty member as a means to keep the council informed and ensure system-wide feedback.

Outreach presentations made by members of this group:

Organization	Date/Place	Presenter's Name	Number Present
CSU Business Assessment Meeting, Cal Poly, Pomona	4/20/01	Tarjan	25

The purpose/progress of IMPAC was reviewed. These faculty/administration assessment leaders were informed of the CIS and business clusters beginning next year and were asked to encourage participation from their respective faculties.