ABSTRACT

Every 10-12 years, the professional computing societies, ACM and IEEE-CS, publish new curricular recommendations for undergraduate programs. On December 20, 2013, an ACM/IEEE-CS Task Force published its new recommendation. Each published recommendation challenges schools to respond. Three core questions involve, "To what extent do these recommendations fit with the mission and goals of a local institution?", "How can a school organize the extensive content into courses?", and "Is it feasible for a school to offer a major that meets all of the recommendations?"

This paper describes how the Grinnell College Computer Science faculty utilized the CS 2013 recommendations, as they emerged, to review and update its program. Both the process and the resulting curriculum directly address the three core questions and may provide insights for faculty at other schools as they review their own programs.

1. INTRODUCTION

Every academic program must consider at least three Core Questions:
1. What content should be covered, consistent with the school's mission and goals?
2. How should that content be organized into courses?
3. What combination(s) of courses should be identified for a disciplinary major?

To promote discussion and provide guidance, the major computing societies, the Association for Computing Machinery (ACM) and the Computer Society of the Institute of Electrical and Electronics Engineers (IEEE-CS), publish recommendations for undergraduate programs every 10-12 years. For example, a joint Task Force issued Computing Science Curricula 2001 in December 2001 [5] with a modest update [1] in December 2008. Specification of content was extensive: [5] spanned 240 pages, identified 14 high-level "Knowledge Areas", and specified 132 detailed "Knowledge Units" covering 280 hours of class time. [5] also outlined a few hypothetical courses to suggest ways to organize topics, but the courses had not been offered at actual schools.

The ACM/IEEE-CS recommendations [1, 5] primarily addressed content (Core Question 1), but not Core Questions 2 and 3. Also, questions sometimes arose in relating ACM/IEEE-CS recommendations to a school's mission. For liberal arts colleges, the Liberal Arts Computer Science Consortium (LACS) published separate guidelines [6] in 2007 that addressed all three Core Questions. Although recommendations from ACM/IEEE-CS and LACS share similarities, differences arise in focus and content. LACS guidelines also grouped content into possible courses and outlined a viable major.
As a partial response, ACM and IEEE-CS took several steps in developing its new curricular recommendations for 2013.

- The new Task Force includes substantial representation from liberal arts colleges.
- Content (Core Question 1) is divided into two tiers to allow some flexibility. [3]
  - All schools should cover most or all core Tier 1 topics.
  - Each school should strive to cover about 80% of Tier 2 content.
- Although still encouraging curricular experimentation, sample "exemplar" courses illustrate the integration of topics [2], partially addressing Core Question 2.
- As an aid to organizing topics into an overall curriculum (Core Question 3), the final report provides five "Curricular Examples [4].

This paper reports how these emerging recommendations have helped one liberal arts school, Grinnell College, update its curriculum. Grinnell's 2012-2013 CS program was largely consistent with both the 2001/2008 ACM/IEEE-CS recommendations [1, 5] and the LACS 2007 guidelines [6]. Section 2 outlines this starting point and describes a CS Curriculum Workshop in July 2013. Section 3 then discusses the use of an hours-oriented spreadsheet, developed by the ACM/IEEE-CS Task Force for the Strawman draft [3], to review Grinnell's three-course, introductory, multi-paradigm sequence.

Between the Strawman draft [3] and the Ironman draft [2], the ACM/IEEE-CS Task Force identified about 1052 learning outcomes to clarify content. Section 4 discusses Grinnell's use of a corresponding spreadsheet to review its full curriculum to identify gaps in its curriculum. Section 5 then describes Grinnell's response. With constraints on the size of majors in a liberal arts setting, a revised, 32-credit CS major will likely cover 70%-85% of the 2013 Tier 1 topics and 51%-61% of Tier 2 topics. Through advising, students might take 6 additional CS credits and attend departmental talks to cover about 96% of Tier 1 topics and 78% of Tier 2 topics. The paper concludes with Acknowledgments (Section 6), Conclusions (Section 7), and References (Section 8).

2. CONTEXT

Grinnell College is an undergraduate, liberal arts college of about 1600 students. Due to system administration duties by one person, the CS teaching faculty numbers 4.6 FTE. In recent years, 12-15 majors have graduated each year. As a liberal arts college, a major at Grinnell cannot exceed 32 credits, plus a few supporting courses. To fit within the College's constraints, Grinnell's 2012-2013 CS major had these elements:

- Multi-paradigm, 3-course introductory sequence (all 3 required): Functional problem solving with Scheme, Imperative problem solving and data structures with C, and Object-oriented problem solving and algorithms with Java
- Core Courses: Analysis of algorithms and Theory of computation (both required)
- Systems course (1 required, both strongly recommended): Computer organization, or Operating systems and parallel algorithms
- Languages course (1 required): Programming language concepts, or Compilers
- Project-based, software development course (1 required): Software design, or Databases and Web Application Design
• Supporting courses: (3 required): Calculus I, Combinatorics (Math) or Discrete Structures (CS), and 1 mathematics elective with calculus (or later) as prerequisite. Additional electives provide a breadth of experiences, but further courses cannot be required by college policy. Students taking only one systems course did not meet recommendations of *Computer Science Curricula 2001* [5], but students taking both systems courses cover most recommendations by both ACM/IEEE-CS [5] and LACS [6].

**Preliminary Faculty Discussions; Grinnell's 2013 CS Curriculum Workshop**

Grinnell's CS faculty review their program regularly every 2-4 years, most recently in late July 2013. Data under review included exit surveys of graduating CS majors, 2008-2010 surveys of alumni, and observations from faculty. In this effort, the faculty articulated themes from graduates and alumni, identified areas that seemed to be working well, brainstormed opportunities for improvement, and explored some new approaches — particularly for projects-based courses. Several themes emerged.

• Important strengths include the introductory three-course, multi-paradigm sequence, and courses in algorithms, theory, architecture, and operating systems.

• College parameters do not allow both architecture and operating systems to be required, so coverage of both topics depends upon advising.

• Coverage of programming languages has extended beyond normal undergraduate requirements, and several graduates have pursued this topic in graduate school.

• A course in computer networks had been offered regularly, but not required. Since the course did not fulfill any current major requirements, enrollment was spotty.

• Coverage of computer security topics was thin.

With this initial analysis, the CS faculty wanted to use the emerging ACM/IEEE-CS recommendations [2] to identify areas of solid coverage, uncover topics needing more attention, and consider how Grinnell's curriculum might relate to emerging subjects.

**3. USING THE HOURS-ORIENTED 2013 STRAWMAN SPREADSHEET**

The ACM/IEEE-CS Task Force prepared a spreadsheet to support each draft of its evolving 2013 curricular recommendation. The spreadsheet for the Strawman draft [3] was organized by high-level knowledge area, rows corresponded to knowledge units, columns to individual courses, and individual cells recorded the hours spent. Grinnell's faculty utilized the spreadsheet for each of the three courses in its introductory, multi-paradigm sequence. In practice, completion of the spreadsheet required several steps.

• In an initial pass, each class period in a day-by-day schedule could be mapped to one or more knowledge units, assigning approximate hours to each topic.

• A course topic might relate to several knowledge units. For example, coverage of an insertion sort might involve loops, arrays, simple sorting, loop invariants, algorithmic analysis, etc. An initial pass for assigning hours often reflected a systematic bias; some topics were shown with many hours, and others very few.

• Adjusting hours among topics required an iterative approach, so that hours were accurate, but also clearly represented the emphases and approaches of each course.
Overall, this detailed review of the 171 Knowledge Units in [1, 5] required moderate time—perhaps 3-5 hours per course. When done, the spreadsheet confirmed many expected themes, but also helped identify areas for course refinement. Details are available in the "course exemplars" section of the CS2013 final report [4, pp. 454-462].

4. USING THE OUTCOMES-ORIENTED 2013 IRONMAN SPREADSHEET

The 2013 Ironman draft [2] added roughly six learning outcomes to each knowledge unit, yielding about 1052 learning outcomes among the 171 knowledge units. In July-August, 2013, the Grinnell CS faculty mapped its entire curriculum, based on a new spreadsheet from the ACM/IEEE-CS Task Force. Several observations follow:

- Completing of a cell in an outcomes-based spreadsheet often is simple; either the course meets the outcome or not. However, a spiral approach may require notations for progressive levels of mastery beyond just "yes" or "no".
- The statement of some learning outcomes raised questions regarding meaning or intention, some outcome statements seemed duplicated in several places, and some might not fit with the mission or goals of an individual institution.
- Some topic hours for the 2013 exemplars did not map clearly to outcomes.
- Faculty reactions varied to completing the outcomes-based spreadsheet. Some found the outcomes-based, "yes/no" format simple and fast. Others found the large size—with 1052 cells per course—frustrating and time consuming.
- Tracking of coverage in the areas of social issues and professional practice seemed particularly difficult, since these themes arise in many Grinnell courses, and it may be hard to name a specific course in which students attain desired mastery.
- To complete the outcomes-based spreadsheet, the faculty identified groups of 2-3 related courses (e.g., introductory courses, systems, projects, network/security, etc.). Two faculty collaborated on each group for 2-3 hours to complete the spreadsheet.

Initial analysis of existing Grinnell courses indicated several omissions, especially in new areas such as computer security. In some cases, new topics or courses had been identified from comments from graduating seniors and alumni, but even for these topics, the detail of the CS 2013 materials helped identify specific themes and topics.

Overall, the review with CS 2013 hours and outcomes raised several issues. To clarify choices, faculty brainstormed approaches, mapped 18 existing or evolving courses to the learning-outcome spreadsheet, and reviewed some 18,936 spreadsheet cells.

5. ADJUSTING GRINNELL’S COURSES AND THE MAJOR

At an early stage, faculty discussions identified the need for a new structure for the CS major. The 2012-2013 major was highly prescribed. Feedback from alumni indicated a need for more flexibility. Incorporation of computer security and networking would require new or adjusted coursework and replace parts of the existing major. Separate discussions of software development (beyond the scope of this paper) suggested separation of the study of methodology from work on software development itself.

Clearly, coverage of the breadth and depth of material in CS 2013 would require considerable ingenuity and careful planning. Further, since Grinnell constrains any major
to 32 credits (plus a few supporting courses), all material recommended for CS 2013
would not realistically fit within a minimal, 32-credit major.

Eventually, a two-level framework for the CS major evolved. The required major
provides a basic foundation for students with wide-ranging career goals:

- Multi-paradigm, 3-course introductory sequence (12 credits required)
- Functional, imperative, and object-oriented problem solving with Scheme, C,
  and Java—largely as developed in the existing 2012-2013 curriculum.
- Required Upper-level Courses (10 credits required)
  - Analysis of algorithms (4 credits)
  - Theory of computation (4 credits)
  - Implementation of programming languages (new/revised, 2 credits)
- Systems Elective I (4 credits required; 8 credits strongly recommended)
  - Computer organization and architecture
  - Operating systems and parallel algorithms
- Systems Elective II (2 credits required, 4 credits strongly recommended)
  - Computer security (new, 2 credits)
  - Computer networks (revised from 4 credits to 2 credits)
- Software Engineering Component (4 credits required, 6 credits recommended)
  - Software engineering principles and methodology (new, 2 credits)
  - Team-based community project (new, 2 credits, may be repeated)
- Supporting courses (12 credits required)
  - Calculus I, Combinatorics (Math) or Discrete Structures (CS), and Mathematics
    elective with calculus or later course as prerequisite

Additional discussion may be found in the "Curricular Exemplars" section of [4,
pp. 480-491]. Beyond courses, advising encourages students to take both options for
System Elective I, both options for Systems Elective II, and some additional work in a
team-based project (38 CS credits total). In addition, regular attendance at a weekly, non-
credit, departmental seminar series would consistently provide modest topic coverage of
social and professional issues (about 4% of the 2013 learning outcomes). [7]

6. ACKNOWLEDGMENTS

The process of revising Grinnell's CS program involved extensive collaboration
among all of Grinnell's CS faculty, including Janet Davis, John David Stone, Jerod
Weinman, and the authors. To begin this effort, the faculty met for a three-day
Curriculum Workshop in late July 2013, with encouragement from Grinnell's Dean's
Office, particularly Associate Dean Mark Schneider.

Parts of this paper were presented orally during a session of the Liberal Arts
Computer Science Consortium, meeting 2-3 August 2013 at Grinnell College.
Subsequent discussions provided helpful feedback for Grinnell's review process.

CS 2013 committee members Andrea Danyluk, Williams College, and David
Reed, Creighton University, provided considerable additional feedback on possible
directions for Grinnell's curriculum and curricular mapping.
7. CONCLUSIONS

Curricular revision requires substantial faculty effort. In Grinnell's case, work started with informal discussions among faculty and with a Curriculum Workshop to review feedback from graduating majors, alumni, and faculty. Grinnell's faculty then utilized both the hours-based spreadsheet from the Strawman draft [3] and the outcomes-based spreadsheet from the Ironman draft [2] to systematically review both individual courses and the overall curriculum. Completion of each spreadsheet required interpretation and complexity, but ultimately highlighted both strengths and weaknesses. Such work provided insights that led to developing some new courses, rethinking several existing courses, and restructuring the major.

The CS faculty continue to refine details for a revised curriculum and major. Allowing both a minimum (32-credit) major and a recommended (38-credit) major allows students to shape their courses according to their educational goals. The following table, from a report from the Grinnell faculty to the 2013 ACM/IEEE-CS Task Force [7], illustrates how various alternatives fit with the emerging 2013 recommendations:

<table>
<thead>
<tr>
<th>Possible Revised CS Major and Program</th>
<th>Tier 1</th>
<th>Tier 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimal major – only the basic requirements</td>
<td>70-85%</td>
<td>51-61%</td>
</tr>
<tr>
<td>Minimal major plus attendance at IP talk/discussion</td>
<td>73-89%</td>
<td>51-61%</td>
</tr>
<tr>
<td>expanded (38-credit) major – both 211/213, both networks/security</td>
<td>92%</td>
<td>73%</td>
</tr>
<tr>
<td>38-credit major plus attendance at IP talk/discussion</td>
<td>96%</td>
<td>73%</td>
</tr>
<tr>
<td>38-credit major plus AI course plus attendance at IP talk/discussion</td>
<td>96%</td>
<td>78%</td>
</tr>
</tbody>
</table>

Altogether, Grinnell's evolving CS program shows one way for a liberal arts college to satisfy most identified targets with modest course offerings and staffing.

8. REFERENCES