

Scholarship Expectations *

Department of Computer Science, Grinnell College

September 2, 2011

Executive summary

In this document, we describe the forms of scholarship that we consider appropriate for faculty members in computer science at Grinnell. Grinnell College's policies on promotion and tenure, expressed in the Faculty Handbook [13] and in various other documents, prioritize peer-reviewed work in a form appropriate to the discipline. We intend this document to help clarify the forms appropriate to the discipline.

Computer science differs from many other disciplines in three respects. First, many subdisciplines of computer science prioritize conference publication and treat it as a final and archival form of publication. Second, many departments (particularly at liberal arts colleges) deem research on computer science education an appropriate area of scholarship. Third, there is a growing trend in the computer science community to treat the products of our research, particularly software, as an appropriate form of scholarship. We affirm these three perspectives. That is, we consider conference publications a high-level form of publication, on par with journal articles, we encourage our faculty to pursue peer-reviewed scholarship in computer science education, and we value appropriately assessed forms of software and other products.

Just as scholarship in computer science differs from that in other disciplines, so do the expectations of a scholar at Grinnell differ from those of scholars at other institutions. In particular, Grinnell encourages faculty, particularly faculty in the sciences, to engage in in-depth research projects with students, often through the Mentored Advanced Project (MAP) program. Grinnell also increasingly encourages faculty to include interdisciplinary projects in their work. The Department of Computer Science affirms these two perspectives, with the caution that student-faculty research and interdisciplinarity, while important, may not be appropriate for every faculty member in the discipline.

*This working document presents a consensus that has emerged among computer science faculty at Grinnell College. Although this document has been communicated to the administration, the college has not provided feedback. In particular, this policy has not been endorsed by Grinnell College. Copyright © 2011-2013 by Janet Davis, Samuel Rebelsky, John David Stone, Henry M. Walker, and Jerod Weinman.

This work is licensed under the Creative Commons Attribution 3.0 United States License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/us/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

1 Introduction

Like all departments at Grinnell, the Department of Computer Science expects its faculty to be active scholars in the discipline. Of course, each discipline has its own norms for sharing scholarly work. In this document, we set out our expectations for scholarly work. We expect that this document will serve our tenure-track faculty as they chart their scholarly careers before tenure and our tenured faculty as they continue their careers, and that it will inform our colleagues on the Personnel Committee and other College committees about the norms of the discipline.

1.1 Goals of this document

Our primary goal in writing this document is to make the faculty review process as transparent as possible. We hope that current and prospective faculty will find it a useful guide. In writing these guidelines, we are following an institutional trend toward transparency and clear departmental expectations for scholarly work.

In writing this document, Grinnell's computer science faculty have tried to synthesize principles, expectations, and guidelines for at least three audiences: our departmental colleges, the Dean and college, and the greater community of computer scientists. Within our department, the faculty want to clarify expectations as much as possible to aid junior colleagues, to guide discussions related to faculty performance, and over the long term to provide insights for prospective faculty.

The computer science faculty also realize that some circumstances within the discipline parallel other disciplines reasonably well, but other circumstances can be somewhat different. Thus, this document provides perspectives and background for the Dean and college committees that consider cases of re-contracting, promotion, tenure, and faculty review.

Finally, the subject of expectations for scholarship is a lively topic of discussion nationally among computer science faculty. Each college and university has specific priorities and perspectives, but many embrace the importance of scholarship in some form. We expect that our own discussions will be informed, at least in part, by the Carnegie Foundation's *Scholarship Assessed*. [11]

This document is designed to describe in some detail an approach used by this department, and we hope that this careful documentation may help other departments and schools as they deliberate about their expectations for scholarship.

1.2 Institutional faculty review processes

Scholarship is only one of three aspects of the work of a Grinnell faculty that are considered in the review process (the others are teaching and service). Furthermore, the department shares the responsibility for conducting reviews with other individuals and committees. We understand, therefore, that this document plays only a small role in the overall faculty evaluation process. Part III of the Grinnell College Faculty Handbook [13] describes the review process in detail. The Office of the Vice President for Academic Affairs provides additional information.

In brief: After gathering and reviewing appropriate data, the department makes a recommendation. The Divisional Personnel Committee reviews the department's recommendation and adds its own. The College Personnel Committee, an elected body of the faculty, then reviews both recommendations and the accompanying documents and submits them, with its own response, to the College's Board of Trustees (or, more typically, to the President as the Board's representative) for a final decision.

1.3 Institutional perspective on scholarship

Scholars in different disciplines present their scholarly work in different ways. As the faculty handbook notes, faculty are expected to carry out "scholarship of high quality in the form of publication, performance, exhibition, or other final form usual to the discipline." [13, IV.C] The College gives special weight to two aspects of scholarship: It esteems work that is peer-reviewed, and, as the Faculty Budget Committee has noted, it particularly values work that has had a clear impact.

1.4 The Department's understanding of scholarship

The Department of Computer Science affirms the significance of peer review and impact. At the same time, the Department understands scholarship broadly and embraces Boyer's four models of scholarship [5]. In this influential report from the Carnegie Foundation, Boyer indicates that, in addition to the traditional *discovery* model of scholarship, three other models significantly benefit society and the learned disciplines: scholarships of *integration*, *application*, and *teaching*. These additional models seem particularly appropriate for scholarship at a small liberal arts college.

In the scholarship of integration, one integrates existing results, typically across disciplines. Such integration might take the form of a literature review, an interdisciplinary analysis, or even certain kinds of textbooks.

In the scholarship of application, one applies knowledge gained through the scholarships of discovery and integration to problems in society or in support of the discipline. While

we all strive to service society and our professions, the scholarship of application represents such service that draws upon the special knowledge and skills of the discipline. In the discipline of computer science, we consider certain kinds of software development as scholarship of application. We expect the resulting software to undergo some form of assessment. While this assessment may incorporate peer review, it may also take forms like those suggested by the Carnegie Foundation [11]. For example, it may be the end users of the software, rather than fellow computer scientists, who provide the most useful assessments of it. Boyer also notes forms of application that include serving as a consultant or assuming leadership in professional organizations.

Finally, Boyer notes the importance of a scholarship of teaching. This form of scholarship seems particularly appropriate for a school like Grinnell that places high emphasis on teaching. Again, peer review is primary. Many of our faculty participate actively in the computer science education community, undertaking a form of scholarship that combines discovery (the creation of new knowledge) with teaching.

As a small department, we do not limit our faculty to teaching and scholarship in particular subdisciplines. We understand computer science broadly, in accordance with our view of the role of the discipline in the liberal arts. The Department supports faculty in choosing the areas of scholarship that best fit his or her talents and interests.

2 Forms of scholarship

Computer science scholarship extends over a wide range projects and activities. Some work fits neatly within a specific subdiscipline of computer science, but other projects are inherently interdisciplinary in nature. The type of scholarship also may widely over what Boyer has classified as the scholarship of discovery, the scholarship of application, the scholarship of integration, and the scholarship of teaching [5].

With such diversity, scholarship can lead to many different types of products, such as presentations, articles in conferences or journals, technical documents, entire books, chapters within books, software, patents, and grants.

In assessing this scholarship, the Department of Computer Science reaffirms the importance of peer review. Not only should projects run to completion with an appropriate product; but others should be able to review the product, provide feedback, and offer perspectives for the future. In some venues, the nature of peer review is quite similar to what might be found in many other academic disciplines. In other circumstances, peer review is strongly present, but the context of that review may be somewhat different than is commonly found within academia. In yet other circumstances, assessment may take other forms, such as those suggested by the Carnegie Foundation [11]. When the nature of assessment or review differs from that used for traditional scholarly products,

faculty are encouraged to describe the review process when reporting on publications in faculty activity reports and elsewhere.

2.1 Conference proceedings

The role of conference proceedings in computer science differs significantly from their role in other disciplines. Computer science conferences typically require authors to submit full-length papers—not just abstracts—which then undergo a thorough and competitive review process. Accepted papers are publicly presented at the conference, published before the conference in an archival conference proceeding, and usually appear in an online repository such as the ACM Digital Library [4].

Conference proceedings may be preferred to journal publications for several reasons. In many subdisciplines, considerable emphasis is placed on the quick dissemination of new research; journal publication may simply take too long. Conference proceedings' shorter time-to-print is significant. Furthermore, conferences provide the opportunity for authors to present and discuss their work before a community of peers. Papers presented at prestigious conferences enjoy high status and visibility; many computer scientists turn to both conference proceedings and journals when seeking important, current, and relevant work.

For those outside the discipline, Appendix A provides further analysis of the role of conference proceedings in computer science.

In summary, peer-reviewed conference proceedings are a “final form usual to the discipline” [13, IV.C] of computer science and the Department values them as such.

2.2 Journal articles

A computer scientist may choose to submit an article for journal publication in order to reach a particular audience or to explore ideas in greater depth than the page limitations of conference proceedings permit.

Longer journal articles provide the opportunity to develop more elaborate theory, describe a system or method in more detail, expand upon experimental results, or synthesize several years of experience with an algorithm, system, or method. Journal publications can be extremely valuable when they fill one of these roles. However, length is an awkward measure of completeness or quality. Indeed, in many sciences, the top journals (e.g., *Science* or *Nature*) have significant page restrictions. Conference papers are long enough to show how new work builds on prior work, so one need not always consult journal articles to follow the full development of a line of research.

Thus the Department values journal articles as publication in a “final form usual to

the discipline,” [13, IV.C] but reiterates that they are not the only or best such form. Conference papers are also a final form of publication in computer science and can have impact equal to that of journal articles. Whether conference or journal publication is preferable for a particular work depends greatly on the nature of the work and the context in which it is developed.

2.3 Magazine articles

The ACM and the IEEE Computer Society publish a number of magazines for scholarly and professional audiences, such as *Communications of the ACM*, *ACM Inroads*, *interactions*, *IEEE Computer*, and *IEEE Intelligent Systems*. Magazine articles enjoy very high visibility. They challenge the author to motivate and explain their work for a broader audience. Yet, they typically have stringent word or page limits, and may even have limits on the number of citations. Peer review of magazine articles can vary dramatically, ranging from no peer review at all, to invitation by a column editor, to review by an editorial board, to double-blind reviewing. Moreover, different types of contributions to the same magazine may receive different levels of peer review. Thus, magazine articles should be assessed on a case by case basis.

2.4 Technical documents

Various groups publish technical documents that describe projects and their results or that propose new mechanisms for accomplishing common or important tasks. For example, one form of technical document are the reports from the working groups that meet at the annual June conference for Innovation and Technology in Computer Science Education (ITiCSE) [1]. These reports are typically published by the Special Interest Group on Computer Science Education (SIGCSE) of the Association for Computing Machinery (ACM) publishes reports of working groups. Another prominent form of technical document is the Request for Comments (RFC) published by the Internet Engineering Task Force (IETF) [16]. RFCs, despite their name, document important Internet standards, including such things as e-mail and domain name protocols. A third form of technical document are reports from professional bodies, such as the joint IEEE/ACM Computing Curricula [3].

Many technical reports represent a form of Boyer’s scholarship of application or scholarship of integration [5] in that they provide useful information for the broader community and draw upon the domain-specific expertise of the authors.

Although technical documents typically present a significant body of work, the process for acceptance and publication varies substantially from one series to another. Many undergo a form of peer review, in which feedback from various audiences are used to

improve or assess the document and to determine its value to the community.

When technical documents receive appropriate external assessment, they serve as a form of peer-reviewed scholarship.

2.5 Textbooks

Textbooks provide a venue for extended integration and development of a subject. Following Boyer's forms of scholarship [5], well-constructed textbooks involve the scholarship of integration and the scholarship of teaching. Textbooks require authors to investigate a subject well beyond the level of normal teaching responsibilities. Thus, the Department of Computer Science embraces textbook publishing as a form of scholarship.

However, the Department also recognizes that peer review is typically a rather different process for textbooks than for other types of publication. Commercial textbooks often undergo an extensive review at multiple stages, from identification of subject area, through organization, to technical details in the final version. In this context, a book is finally published only after the reviewers, publisher, developmental editors, and author are satisfied with the technical content, structure, and exposition of a final manuscript.

However, only textbooks with significant market appeal receive this close attention. Textbooks for a limited market are seldom of interest to commercial publishers, regardless of their quality. For this reason, and also because of the increasing cost of textbooks, many faculty are turning toward Web-based or open source textbooks for their writing. Assessing the value of these textbooks is more difficult because there is a less formal peer-review process. Still, there are some implicit processes. For example, one might argue that adoption of a text constitutes a type of review; the faculty would not use the book if it were not of sufficient quality.

It remains to be seen whether a peer-review process for these new kinds of textbooks will emerge and establish a useful standard of quality. The Department encourages its faculty who choose to write textbooks to identify the forms of dissemination (commercial or noncommercial, copyrighted or open source, Web or printed, etc.) that they expect to be most valuable to their community. In making choices, faculty should consider what forms of peer review are available.

2.6 Other books

Books of other kinds provide opportunities for extensive exploration of a subject. Typically, books allow authors to go well beyond the scope a journal article—even an extended journal article.

As with textbooks, however, the peer-review process can vary widely. Some publishers will accept almost anything, while others require an extensive review and editing process before accepting and publishing a manuscript.

2.7 Book chapters

Book chapters provide focused contributions on a designated theme. Typically an individual or small group serves as project leader and editor, and this leader formulates a plan, including topics and book structure. The leader then invites potential authors to write individual chapters.

As with textbooks and other books, reviewing of book chapters is sometimes strong and careful, sometimes almost non-existent. However, in most cases, a leader will invite contributions only from authors who are known to have appropriate expertise and writing ability. From this standpoint, the invitation to write a book chapter represents a professional achievement, although the level of peer review must be clarified for each project.

2.8 Software

Scholarship arises in many disciplines when the scholar identifies problem and then explores it creatively. In computer science, one common method of addressing and resolving problems involves the development of software. Software may demonstrate a new approach and/or the creative integration and modification of existing techniques. Altogether, the development of software to address a problem often illustrates Boyer's scholarship of discovery, scholarship of integration, and scholarship of application. When the problem is related to teaching or pedagogy, the development of software also may include elements of the scholarship of teaching.

However, peer review of software is a difficult and complex issue. One approach is to describe the software in a peer-reviewed publication, which works well when the underlying problem comes from a regular academic discipline and supports an intellectual problem.

In other cases, software might be reviewed by technical reviewers. For example, one might submit a contribution to the Linux kernel by sending e-mail to the appropriate mailing list for kernel developers. This contribution would go through several stages of expert review [9], analyzing its technical quality and appropriateness in detail. If the other developers find the software to have sufficient merit, the head of the project could decide to incorporate it into the next release. This process reliably distinguishes valuable contributions from worthless ones and is sufficiently similar to traditional academic review to count as a peer-review process.

Unfortunately, not all development of technically significant software fits well into either of these two frameworks. Nevertheless, in many instances, a successful software package has a greater impact than other forms of publication, even though the nature of this impact is different. In particular, end users experience the effectiveness of software from an outside perspective, and these users often provide the feedback that is the intended result of a more traditional peer review.

Overall, the development of software can engage a faculty member in many aspects of significant scholarship, but review of software presents challenging issues. Sometimes peer review can occur in a traditional context, but sometimes the quality and impact of the work may require inferences about or explicit statements from the user community. In those cases, the rubrics and approaches of the Carnegie Foundation's *Scholarship Assessed* [11] are likely to provide an appropriate starting point.

For those outside the discipline, Appendix B provides further analysis of the role of software in computer science.

2.9 Patents

Patents are valued in many scientific disciplines, and patents may apply within some areas of computer science—particularly in areas related to computing hardware. When applicable, a patent is granted after formal review of a careful application (a paper or product). Thus, the Department of Computer Science accepts patents as a form of peer-reviewed publication.

We note, however, that in recent years applications for patents on software, algorithms, and programming techniques have often received inadequate reviews. Many such patents have been granted despite extensive prior art, even for techniques that are obvious to experienced practitioners. We advocate maintaining a high standard in such cases and suggest that it would be appropriate to provide additional evidence of originality and significance when adducing a software patent as a peer-reviewed publication.

2.10 Grants

In many settings, the grant of funding to a faculty member includes all elements of a typical peer-reviewed paper in another discipline, so the winning of a competitive grant should count the same as a peer-reviewed paper.

In a normal grant process, a scholar compiles a careful proposal, based on past scholarship as well as a vision and plan for the future. The proposal requires synthesis of much background material, creativity, development of a careful plan, and evidence that the proposed plan can be successful. For many grant programs, proposals are subject to

a careful peer-review process, and acceptances are highly competitive.

3 Computer science at a liberal arts college

As a liberal-arts college, Grinnell provides an excellent home for Boyer’s broader perspective on scholarship [5]. As active teacher-scholars, we have the opportunity, and indeed the responsibility, to carry out scholarship, not just by creating new knowledge, but by disseminating it. We teach our students to think and communicate clearly and can draw upon those skills to synthesize summary documents. We care about teaching and can make teaching a core part of our scholarship. And, particularly through the Expanding Knowledge Initiative [12], we can think about ways to apply knowledge from one discipline to another.

3.1 Computer science: an open discipline

Like most disciplines, computer science has many subdisciplines and many connections to other disciplines. Grinnell’s Department of Computer Science is firmly committed to supporting a broad range of areas with a clear computer-science component, including core areas that range from the theoretical to the experimental, interdisciplinary areas that include a clear algorithmic or computational component, and forms of scholarship that have a clear impact on computer science.

While our discipline includes the term “computer”, not all scholarship need include computers or computation. For example, studies on the limits of computability need not use computers, and studies on the use and usability of technologies need not solve problems computationally.

Our department also highly values computer science education as an important scholarly domain. While, in many cases, computer science education serves as a faculty member’s secondary area of scholarship, we expect that some faculty will choose computer science education as their primary research domain. This perspective was affirmed by external reviewers [10].

3.2 Including students in research

The ability to emphasize student-faculty research is one of the strengths of a liberal-arts college like Grinnell, and Grinnell’s Science Division has a long-standing tradition of including students in faculty research. Such student–faculty research is valued by the College, the Division, and the Department. Opportunities for such research take a

variety of forms, including in-class projects, independent study projects, and summer Mentored Advanced Projects (MAPs).

We are fortunate that Grinnell students are capable of contributing to many kinds of research in computer science, particularly in experimental and applied branches of the subject. We encourage our faculty to consider projects that are likely to support student involvement.

At the same time, some areas of computer science can require such significant preparation that it is unlikely that undergraduates can make real contributions. For example, successful research in complexity theory typically requires deep understanding of a large body of knowledge, beyond even the initial graduate-level course in the subject. Also, in some areas to which our undergraduates might be able to contribute, the effort of preparing and supporting students to carry out the research can be so burdensome that it hinders, rather than helps, faculty work.

Hence, while we encourage our faculty to select research projects that can involve undergraduates, we do not require them to do so. In the end, it is most important that faculty choose areas and projects that interest them and to which they expect to make contributions.

3.3 Mentored Advanced Projects

For the reasons discussed above, we encourage, but do not require, members of the Department of Computer Science to supervise Mentored Advanced Projects (MAPs) [14]. While other forms of student-faculty research also benefit students, the in-depth collaboration available through MAPs is a particularly valuable opportunity for our students.

In our experience, MAPs should not be reserved for our best students. Often, our mid-level students derive greater benefit from MAPs, as they discover that they have real skills and that they can be surprisingly successful in the focused environment that a MAP provides. We have often heard from such students how much difference a MAP, or some similar major experience, has made in their careers.

These benefits to students do not come without costs. Certainly, supervising a MAP in computer science requires a great deal of faculty input, not just in selecting and designing a project, but also in teaching students about a new domain of knowledge. In that sense, MAPs are as much a form of teaching as they are a form of scholarship. Furthermore, while MAPs may benefit a faculty member's research program, the time required to train and supervise the students equals, and often exceeds, the value of their contributions to a faculty research project.

Because of the costs associated with MAPs, we encourage faculty, particularly pre-tenure faculty, to consider carefully which MAP opportunities to provide and to focus

the MAPs that they offer on projects that advance their respective scholarly programs.

4 Other issues

The Department of Computer Science encourages its faculty to engage in a wide range of scholarly endeavors. Previous sections have commented upon many common forms of scholarship and have related scholarship in computer science to the liberal arts environment. This section examines additional issues related to scholarship.

4.1 Other kinds of scholarly activities

The Department encourages its faculty to be actively engaged within the broader computing community. Sometimes this work leads directly to a paper that can be peer reviewed, and sometimes it takes the form of service (e.g., serving as Treasurer for a professional organization). However, in many settings, professional work requires a faculty member to draw extensively on scholarly insights, expertise, and creativity, even though the product is not available for normal peer review.

For example, experienced faculty may be called on to review papers submitted to conferences or journals, to participate in external reviews of other departments, to conduct personnel reviews for faculty seeking tenure or promotion at another institution, or to join a panel that evaluates grant proposals. Invitations for such work recognize faculty member's stature and experience in the field. Further, many of these activities lead to written reports that may entail a considerable synthesis of technical material applied to new environments. (For example, several of Henry Walker's recent external review reports have exceeded forty pages and covered many aspects of a school's program in computer science.) However, while these reports often provide invaluable feedback to the client, the reports are private. Although the clients examine them carefully and evaluate them critically, the reviewer normally receives no feedback, so that there is no useful peer review in such cases.

Similarly, faculty are sometimes asked to consult with schools, departments, or individuals on projects and grants. For example, a faculty member may be asked to assess another institution's proposal for a grant from the National Science Foundation. Again, this work may yield a written report, but the document(s) likely will be private and not available for peer review.

Editing a journal or newsletter is yet another form of scholarship. In some cases, this type of work includes writing articles, and in principle these articles could be subject to peer review. However, the work of an editor also involves gathering, assessing, and improving the writing of others. Often these tasks require the editor to engage tech-

nical material at a significant level, but peer reviewers would often be hard pressed to distinguish the specific contributions of the editor as distinct from other writers and contributors.

Work on a program committee creates similar problems regarding assessment and peer review. Of course, some roles on a conference committee, such as making local arrangements, clearly emphasize service rather than scholarly insight. However, in many cases, the development of the technical program for a conference requires substantial scholarly involvement, insight, and creativity. In this case, scholarship leads to a technical program, but peer review of the program does not seem productive.

Altogether, the Department of Computer Science and Grinnell College encourage faculty to be active in a broad professional community. Sometimes this work fits nicely within the traditional context of peer review, sometimes the products of the work are private and not available for peer review, and sometimes the scholarly activities result in work to which peer review does not seem to apply.

4.2 Interdisciplinary work

Although interdisciplinary scholarship is not required, we value it from two different perspectives.

From an institutional perspective, Grinnell College's commitment to interdisciplinary scholarship is expressed through the Expanding Knowledge Initiative (EKI) [12]. A faculty member whose scholarship integrates different forms of knowledge or ways of thinking serves as an example of the integrative nature of the liberal arts. Moreover, interdisciplinary scholarship can feed into interdisciplinary teaching—the main focus of the EKI.

From a disciplinary perspective, research in computer science is often driven not by questions but by problems: for example, to identify likely inhibitors of a particular gene's expression, or to design cell phones that can be used by deaf people to converse in sign language. Such problems can arise from other fields of research, such as biology in the first example, or from people's needs and desires in everyday life. The computer scientist brings expert knowledge of computational principles and system design, but solving the problem effectively also requires knowledge of disciplines relevant to the application. Biological knowledge is a prerequisite to modeling biological problems; building effective tools for the deaf requires understanding of not only sign language but Deaf culture. As Fred Brooks argues, working on a concrete problem forces the computer scientist to face the whole problem in all its complexity, often leading to new approaches that enrich computer science as a discipline [6].

Futhermore, while computer science was itself an emerging discipline not so long ago, computer science now contributes to new disciplines such as bioinformatics and human-

computer interaction. Computer scientists who work across disciplines help build such new fields.

5 Conclusions

Section IV.C. of the Grinnell College Faculty Handbook [13] discusses “Scholarship,” beginning with this paragraph:

Grinnell College, while holding excellence in teaching to be its primary mission, recognizes the importance of scholarship as a complement to effective teaching, as a source of intellectual vitality for the institution, and as a contribution to the fund of human knowledge. The quality of a faculty members scholarship is consequently an important criterion in evaluation by colleagues for promotion, the granting of tenure and merit-based salary increases.

The Department of Computer Science takes a broad view of faculty scholarship, including the scholarship of discovery, the scholarship of synthesis, the scholarship of integration, and the scholarship of teaching, as described by Boyer [5]. The Department also embraces the College’s belief that scholarship should be subject to appropriate review beyond an individual author. In some cases, however, a significant challenge for the discipline of computer science can be to identify an appropriate framework for this external review.

For many projects, within computer science or extending through an interdisciplinary environment, traditional peer review works well. Reviews provide fine feedback to authors, and the quality of a work can be determined by the venue in which it is accepted. For example, the traditional peer-review process is clearly applicable in the following settings:

- publication in journals;
- publication and presentation at conferences (which are subject to peer review and often have low acceptance rates; such conference are often, but not always, more competitive than corresponding journals);
- textbooks or other books (although vanity presses must be distinguished from well-respected, commercial publishers);
- book chapters (for which an invitation to contribute, by itself, can signify a professional distinction);

- patents;
- grants (where acceptance rates may vary from quite mild to extremely competitive); and
- technical reports (although the review process must be clarified);

Other forms of scholarship may be more difficult to assess, and are likely to need to be addressed in a case-by-case basis, often in innovative ways. These forms of scholarship include

- articles in professional magazines;
- some forms of interdisciplinary work, particularly ones in which the goal of the work may be rather different from computing;
- consulting projects; and
- software.

A The roles of peer-reviewed conferences

Within computer science, peer-reviewed presentations with accompanying archival publication provide a valid mechanism to disseminate scholarly results. The best conferences, for example, are extremely selective, often accepting only 25%-35% of the papers submitted. Historically, the field of computer science has been viewed as changing so quickly that conferences have often provided the primary forum for the communication of many results; books and other monographs simply take too long to appear.*

Since peer-reviewed publication in computer science normally follows a different process than is used in many other disciplines, this section describes the process in some detail and identifies the mechanisms for peer review. To reinforce our own perspectives and experience, we also draw upon statements of the National Academy of Science, the computing professions, and the College's own consultants.

Peer review and publication in computer science

To clarify the scholarship and peer review process within this discipline, we describe a typical research scenario.

1. A researcher begins work.
2. When partial results are obtained, the researcher may propose a poster for presentation at a regional, national, or international conference. Depending upon the conference, this poster submission may or may not be reviewed.
3. As significant results emerge from the research, the researcher integrates them into a research paper and submits the paper to a conference. Typically, the submission is 4–6 months before the conference.
4. At all conferences of any stature, the paper is subject to a peer-review process, typically involving at least three reviewers.
5. The researcher receives notification of acceptance or rejection along with reviews of the paper. Acceptance rates of 30% and lower are not unusual. (For example, acceptance rates for the prestigious *CHI Conference on Human Factors in Computing Systems* have been as low as 15%, and ranged between 22% and 25% for the years 2005–2009 [4].)

* Adapted from *Scholarship Expectations*, adopted October 11, 2004, by the combined Department of Mathematics and Computer Science at Grinnell College and from a December 9, 2008, letter to Associate Dean Kathleen Skerrett and the Personnel Committee from the senior members of the Department of Computer Science.

6. Due to time constraints, only minor revisions in a paper are allowed. The final paper is due within a few weeks of receiving reviewer comments and typically 2-3 months before the conference.
7. The paper appears in a volume of proceedings, published before the conference and available at the conference. The paper often also appears in an online digital library.
8. The researcher presents ideas from the published paper during a conference session, to disseminate ideas quickly and to gain insight for new research.

Since conferences are the primary venue for the dissemination of results, conference papers represent the final form of presentation within computer science. In computer science, conference publications are peer reviewed, edited, revised, and printed before the conference presentation. For journal articles, on the other hand, the lead time varies considerably from one journal to another, and is often on the order of months or even years—far too long to have maximal impact. As a result, there simply is no expectation of a subsequent journal paper; the conference paper is itself considered archival.

Thus a conference paper is not, in any sense, an abstract. In the published proceedings, a brief summary of the paper is usually prepended to its text, often under the heading “Abstract,” but this heading denotes only the summary (typically about 250 words), not the entire published paper. For example, for most conferences of the Association for Computing Machinery (the oldest professional organization in the field), conference papers are five or more pages long, in nine-point type, single-spaced and in two-column format. Such a paper represents substantial work, particularly in a technical discipline like computer science. It is also a significant amount of writing. To be more specific, a recent five-page paper that Sam Rebelsky submitted is 4800 words long. When reformatted in “manuscript” form (12 point, Times New Roman, double-spaced), this corresponds to about seventeen pages of text.

Given this pattern of publication in computer science, it is important to stress that conference proceedings provide a vital source for current research. When computer scientists look for important results, and when we send our students to look for important results, the sources consulted are conference proceedings as often as journals.

Altogether, the publication of a conference paper represents a significant peer-reviewed accomplishment, based on substantial work. Such an article might run to fifteen pages or more in manuscript form, and publication precedes the accompanying conference presentation. Furthermore, the conference paper is the final form for results in computer science. After a conference presentation, researchers move on to their next challenges; they often do not consider taking the time to rewrite their work for an archival journal, because such articles may simply take too long to appear.

In addition, our professional society, the Association for Computing Machinery (ACM) notes that it often expects conference publications to be the final, archival repository for a piece of research. ACM guidelines indicate that

The Board and the Conferences and Symposia Committee concur that authors of papers submitted to ACM Conferences should be able to anticipate to the extent possible whether publication in conference proceedings will or will not jeopardize future publication in formal ACM publications. As such, we are seeking cooperation in defining in advance, in the Call for Papers, whether or not the conference record will be “widely disseminated”. [2]

That is, when ACM designates a conference as “widely disseminated”, the paper in the proceedings of that conference is expected to serve as the final and archival version of the work.

The majority of conferences in which Grinnell Faculty publish, including the ACM Symposium on Computer Science Education (SIGCSE), the ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE), and the ACM Conference on Human Factors in Computing Systems (CHI), consider their papers “widely disseminated”.

Some conferences also include non-archival tracks such as workshops, works-in-progress, or position statements. Presentations in these tracks typically require a much shorter, sometimes abstract-length, paper submission, and the submission deadline is much closer to the date of the conference. The submission still undergoes peer review, but it is considered an incomplete work. Although the submission may appear in an online digital library, this does not preclude a later, final publication—even as a paper in the same conference a year or two later. Indeed, some conferences, such as the *CHI Conference on Human Factors in Computing Systems*, explicitly distinguish between archival and non-archival conference tracks [8].

Comments from the National Academy of Science

The National Academy of Science (NAS) is the science advisory board to the U.S. Congress and provides objective and well-considered reports to the general public. Section 4, *Evaluating Research in ECSE*, and Appendix B, *Comparing Journal and Conference Publication*, of a NAS report [17] affirm that conference proceedings have equal or stronger weight than journal publications. NAS notes, for example, that

A substantial majority of respondents to the CRA–CSTB survey of ECSE [Experimental Computer Science and Engineering] faculty preferred conferences as the means of dissemination by which to achieve maximum intellectual impact; many fewer preferred journals. [17, p. 62]

Comments from the Computing Research Association

The Computing Research Association (CRA) is a membership organization, composed

primarily of research departments of computing in universities and industry. The organization is dedicated to research in the discipline, and its elected Board of Directors is dominated by well-known researchers at Ph.D.-granting institutions.

In August 1999, the Board of Directors of the Computing Research Association approved a Best Practices Memo on “Evaluating Computer Scientists and Engineers For Promotion and Tenure” [18]. The memo itself was prepared by David Patterson (University of California, Berkeley), Lawrence Snyder (University of Washington), and Jeffrey Ullman (Stanford University), three of the most prominent computer scientists in the world. The memo describes practices for both theorists and experimentalists.

For theorists, the CRA document states, “conference publication is highly regarded in the theoretical community.” The memo continues,

For experimentalists conference publication is preferred to journal publication, and the premier conferences are generally more selective than the premier journals [Academic Careers, 94]. In these and other ways experimental research is at variance with conventional academic publication traditions.

The reason conference publication is preferred to journal publication, at least for experimentalists, is the shorter time to print (7 months vs 1-2 years), the opportunity to describe the work before one’s peers at a public presentation, and the more complete level of review (4-5 evaluations per paper compared to 2-3 for an archival journal) [Academic Careers, 94]. Publication in the prestige conferences is inferior to the prestige journals only in having significant page limitations and little time to polish the paper. In those dimensions that count most, conferences are superior [18].

Additionally, it is important to note that the work of many computer science faculty at undergraduate institutions, including Grinnell, falls in the second category (experimental, rather than theoretical). Thus, publication in conference proceedings may be a preferred forum for peer-reviewed publication by faculty in this department.

Comments from our 2008 external reviewers in computer science

In 2008, Grinnell’s Department of Computer Science engaged in its periodic external review. The review team consisted of Lisa Meeden from Swarthmore College, Lea Wittie from Bucknell University, and Herbert Dershem from Hope College. Section VII of the team’s final report, “Evaluation of Scholarship in Computer Science,” begins:

The review team was asked to address the issue of the evaluation of scholarship in the discipline of Computer Science. The team endorses the views expressed in the document “Scholarship Expectations” written and endorsed by the Grinnell College Department of Mathematics and Computer Science in October 2004. In particular, we concur with three essential principles expressed in that document of engagement, breadth of scholarship, and the importance of

peer reviews. [10, p. 7]

The first part of this section on scholarship concludes:

The members of the review team affirm that at their home institutions, publication of Computer Science Scholarship in conference proceedings is given as much weight as publication in archival journals. [10, p. 7]

An example

As an example of the difference of impact between conference papers and journal articles, let's consider one of the more important approaches to storage from the past few decades. RAID (Redundant Arrays of Inexpensive Disks) was an idea proposed to deal with the huge differences in both price and reliability between high-end and low-end disks. The work on RAID not only led to implementations that are now readily available on consumer-grade computers, but also to a reconsideration of the ways that we might provide reliability.

The first paper on RAID [19] was a short, seven-page conference publication. A few years later, the authors of the first paper followed it up with a forty-page journal article [7] that tied together ideas from that first paper and many subsequent improvements.

Even though a decade and a half have passed since the publication of the journal article (more than enough time to overcome the delay in publication), the conference paper still receives many more citations. As of 20 September 2010, our professional society lists 433 citations to [19] (about 45 in the past year and a half) and only 193 to [7] (about 31 in the past year and a half). In terms of current usage, the conference paper was downloaded from the ACM Digital Library 539 times in the twelve months before 20 September 2010, while the journal article was downloaded 483 times.

We certainly see similar patterns in other cases in which both conference paper and journal articles are published. However, it is frequently the case that the conference paper stands as the final, authoritative version of a piece of work.

Summary

Grinnell has long acknowledged that different disciplines have different models of publication. The College has long emphasized that we accommodate these differences, provided that peer review is at the center of publication.

The Faculty Handbook clearly states that faculty publication should be “in final form usual to the discipline.” [13, IV.C] The above resources are clear, explicit, and authoritative that peer-reviewed conference proceedings are precisely a “final form usual to” computer science.

B The roles of software

Computer science differs from many other disciplines in that scholarship often includes the creation of computer software, “artifacts” that can be used. At the time Grinnell’s Computer Science major was proposed, the faculty who designed the major suggested that software was just a small part of computer science and that faculty scholarship would take more traditional forms. However, the role of software in the discipline has changed significantly. Consider some recent software that has had a huge impact on the world: The World Wide Web, Mozilla (the first graphical Web browser), Facebook, Twitter. These artifacts stand for themselves, and are not generally the subject of scholarly papers (at least not scholarly papers by the authors of the software).

As Hafer and Kirkpatrick suggest, “Academic computer science has an odd relationship with software: Publishing papers about software is considered a distinctly stronger contribution than publishing the software” [15]. Although the paper is valued, the software often contributes more to the community than the paper. A new model is needed.

Of course, not all software written by computer science faculty should be considered a scholarly product. For example, a simple Web site or programming utility is unlikely to contribute to the discipline. However, software that falls within one of Boyer’s four modes of scholarship (discovery, integration, application, or teaching) [5] can and should certainly be considered applicable, provided its scholarly merit can be assessed.

One approach to assessing software is scholarship is through peer review, a form of assessment similar to the scholarly review that accompanies more traditional forms of scholarship. Review by other practitioners of the discipline helps affirm the value of the scholarship, whether that scholarship be software or paper.

Open source software provides one obvious form of peer review. In a typical open source project, new components of the project are vetted by a small group of project administrators. In addition, other contributors to the project regularly review code as they make their own additions. Hence, components of an open source project that continue to be included in the project reflect significant peer review.

While peer review is important, the College has noted that other aspects of a research project are important. In particular, in determining the value of a publication for a merit review, the Faculty Budget Committee considers the *impact* of the project. As noted above, software can have a significant impact on the community it serves.

Grinnell’s Department of Computer Science affirms the role of software as a scholarly product, whether or not the software is accompanied by a scholarly paper about the software. However, the department also affirms that in order for software to be judged as scholarly, it should be subject to appropriate assessment.

References

- [1] ITiCSE conferences. <http://www.sigcse.org/events/iticse>. Accessed August 31, 2011.
- [2] ACM. ACM Policy on pre-publication evaluation. http://www.acm.org/publications/policies/prepub_eval. Accessed October 19, 2010.
- [3] ACM / IEEE-CS TASK FORCE ON THE CURRICULUM. *Computing Curricula 2001*. ACM and the IEEE Press, 2002.
- [4] ACM Digital Library. <http://www.acm.org/dl>. Accessed June 26, 2009.
- [5] BOYER, E. L. *Scholarship reconsidered: Priorities of the professoriate*. Princeton University Press, Lawrenceville, N.J., 1990.
- [6] BROOKS, JR., F. P. The computer scientist as toolsmith II. *Communications of the ACM* 39, 3 (Mar. 1996), 61–68.
- [7] CHEN, P. M., LEE, E. K., GIBSON, G. A., KATZ, R. H., AND PATTERSON, D. A. RAID: high-performance, reliable secondary storage. *ACM Comput. Surv.* 26, 2 (1994), 145–185.
- [8] CHI 2009 Call for Papers. <http://www.chi2009.org/Authors/>, 2008. Accessed June 30, 2009.
- [9] CORBET, J. The life cycle of a patch. <http://ldn.linuxfoundation.org/book/22-the-lifecycle-a-patch>, 2008. Accessed October 18, 2010.
- [10] DERSHEM, H., MEEDEN, L., AND WITTIE, L. *Report of External Review Team, Computer Science Department, Grinnell College*. March 2, 2008.
- [11] GLASSICK, C. E., HUBER, M. T., AND MAEROFF, G. I. *Scholarship assessed: Evaluation of the professoriate*. Jossey-Bass, San Francisco, CA, 1997.
- [12] GRINNELL COLLEGE. Expanding Knowledge Initiative. <http://www.grinnell.edu/offices/dean/eki>. Accessed October 4, 2010.
- [13] GRINNELL COLLEGE. *Faculty Handbook*. <http://www.grinnell.edu/offices/dean/handbook>. Accessed August 20, 2010.
- [14] GRINNELL COLLEGE. Mentored Advanced Projects. <http://www.grinnell.edu/offices/dean/map>. Accessed October 19, 2010.

- [15] HAFER, L., AND KIRKPATRICK, A. E. Assessing open source software as a scholarly contribution. *Commun. ACM* 52 (December 2009), 126–129.
- [16] INTERNET ENGINEERING TASK FORCE. Request for Comments (RFC) Pages. <http://www.ietf.org/rfc.html>. Accessed August 31, 2011.
- [17] NATIONAL ACADEMY OF SCIENCE. *Academic careers for experimental computer scientists and engineers*. National Academy Press, Washington, D.C., 1994.
- [18] PATTERSON, D., SNYDER, L., AND ULLMAN, J. *Best Practices Memo: Evaluating computer scientists and engineers for promotion and tenure*. April 7, 2000.
- [19] PATTERSON, D. A., GIBSON, G., AND KATZ, R. H. A case for redundant arrays of inexpensive disks (RAID). In *SIGMOD '88: Proceedings of the 1988 ACM SIGMOD international conference on management of data* (New York, NY, USA, 1988), ACM, pp. 109–116.