Social Networks Generate Interest in Computer Science

Casey Alt, Owen Astrachan, Jeffrey Forbes, Richard Lucic, Susan Rodger
Department of Computer Science
Duke University, Durham, NC 27708
{casey.alt,ola,forbes,lucic,rodger}@duke.edu

ABSTRACT

For forty years programming has been the foundation of introductory computer science. Despite exponential increases in computational power during this period, examples used in introductory courses have remained largely unchanged. The incredible growth in statistics courses at all levels, in contrast with the decline of students taking computer science courses, points to the potential for introducing computer science at many levels without emphasizing the process of programming: leverage the expertise and role-models provided by high school mathematics teachers by studying *topics that arise from social networks and modeling to introduce computer science as an alternative to the traditional programming approach.* This new approach may capture the interest of a broad population of students, crossing gender boundaries. We are developing modules that we hope will capture student interest and provide a compelling yet intellectually rich area of study. We plan to incorporate these modules into existing courses in math, statistics, and computer science at a wide variety of schools at all levels.

Categories and Subject Descriptors K.3.2 [**Computers & Education**]: Computer & Information Science Education — *Computer Science Education*

General Terms Languages, Algorithms, Human Factors

Keywords Social Networks, Programming, Computer Science Education

1. MOTIVATION FOR AN ALTERNATIVE TO PROGRAMMING

Programming courses fail to attract a wide constituency of students intending to major in biology, economics, public policy, mathematics, chemistry, and other natural and social sciences at all levels. High school teachers without extensive education in computer science often must work diligently to stay just ahead of students. Programming gets harder while it gets easier, e.g., new language features that are part of Java 5 make writing loops simpler, but require discussing more advanced concepts earlier. Many textbooks confine their examples to the territory circumscribed by activities such as managing bank accounts and employee data processing. Rather than leveraging the contributions that computation and algorithms have made to areas such as genomics, we have clung to our data processing roots and missed the opportunity to capitalize on our students' widespread interest in the life sciences, particularly among female students.

Although individual programs are exemplary, e.g., Guzdial's work at Georgia Tech [13, 12, 31], and work with Alice by Pausch and others [7, 25], when we look at the first year of computer science as taught at the vast majority of schools we see a course very similar to the Fortran and Pascal courses from 20 years ago. Changing existing introductory programming courses is not a viable option. Despite the pessimism conveyed above, our current courses do a reasonably good job of preparing students to major in and practice traditional computer science. That being said, these courses do a remarkably poor job of attracting previously uninterested students to the field. Programming in high schools, and AP Computer Science in particular, is not providing a sense of excitement and an interest in further study, particularly among women. The number of students taking AP Computer Science is declining. It is the only AP program with decreasing enrollment. At the same time, other mathematically-oriented courses such as AP Statistics are booming. Enrollment in AP Statistics has surpassed BC calculus and, if measured by current trends, will reach the same level as Biology in roughly ten years [1]. Anecdotal observations of the online listserv used by high school teachers indicates that that the AP Statistics course is "easy" (or easier than calculus), that it is relevant to many disciplines, and that math teachers are comfortable teaching it after a brief introduction (and often a re-introduction) to the field. This assessment starkly contrasts with programming and computer science—areas in which many (though not all) high school mathematics teachers have little or no background.

We are developing materials based more on mathematics and modeling than on programming. We anticipate that such materials will appeal to a wide range of students, particularly to women. We want materials related to interesting, real-world phenomena and relevant examples. We are not proposing to transform traditional first courses in computer science which have relied on programming and which will likely continue to do so. Nor are we proposing to change higher level courses. What we are proposing is a new curricular model to introduce new materials into existing courses at all levels. Only some of the students in these courses will continue with computer science, but many more students

SIGCSE'06, March 1–5, 2006, Houston, Texas, USA.

will take a computer science course if this proposal succeeds since they will have an avenue that broadens the appeal of the discipline.

2. THE SCIENCE OF NETWORKS

We have chosen *The Science of Networks* as the overarching theme on which to develop materials and modules that form an alternative introduction to Computer Science. One reason for choosing this is its roots in mathematics, computer science, sociology, and operations research. Academic centers, e.g., CASOS [5] address this area in the context of computer science and related fields while INSNA, the *International Network for Social Network Analysis* provides more of a social science perspective. Additionally, recently popular works like *Small Worlds* [37], *Linked* [2] and *Six Degrees* [38], more formal treatises like [36, 24] have brought this area into both popular and academic focus.

From [38] we have the following broad description of the approach we are proposing. *In a way, nothing could be simpler than a network. Stripped of its bare bones, a network is nothing more than a collection of objects connected to each other in some fashion. ... On the other hand, the sheer generality of the term network makes it slippery to pin down precisely, and this is one reason why a science of networks is an important undertaking. Understanding networks in this more universal fashion is an extraordinarily difficult task. Not only is it inherently complicated, but also it requires different kinds of specialized knowledge that are usually segregated according to academic specialty and even discipline. Physicists and mathematicians have at their disposal mind-blowing analytical and computational skills, but typically they don't spend a whole lot of time thinking about individual behavior, institutional incentives, or cultural norms. Sociologists, psychologists, and anthropologists, on the other hand, do. And in the past half century or so they have thought more deeply and carefully about the relationship between networks and society than anyone else — thinking that is now turning out to be relevant to a surprising range of problems from biology to engineering. But, lacking the glittering tools of their cousins in the mathematical sciences, the social scientists have been more or less stalled on their grand project for decades .... The science of networks must become, in short, a manifestation of its own subject matter, a network of scientists collectively solving problems that cannot be solved by any single individual or even any single discipline. It's a daunting task, made all the more awkward by the long-standing barriers separating scientists themselves.*

3. BROADENING PARTICIPATION

Our proposed approach to significantly increase the numbers of women enrolling in undergraduate computer science courses is strongly supported by previous research in the area of computing and gender. Two of the most salient trends that emerge from a review of relevant literature are:

1. women often form long-term attitudes about computer science at the high school level
2. women tend to respond most favorably to computer science when it is demonstrated that the field is not just about programming, but rather is deeply interconnected to other already familiar academic areas and can be concretely applied to help solve important "real-world" problems.

Fisher, Margolis, and Miller [9] were among the earliest researchers to observe the importance of early intervention in girls' attitudes regarding computer science finding that most girls have already made up their minds about computer science prior to their first year of undergraduate study. Margolis and Fisher's initial findings have been strongly reinforced by their more recent, wide-scale study of undergraduate computer science students at Carnegie Mellon University in 2002. In *Unlocking the Clubhouse: Women in Computing* [20], they observed that "for almost one-third of the women in our study, a high school programming course became the deciding factor in their decision to major in computer science" compared to only 9% of males for which this was true. In an analogous 2003 application of the Carnegie Mellon model to Stanford University undergraduates, Lilly Irani similarly concluded that "socialization experiences prior to college play a major role in students' decisions to pursue computer science." [15]

Our proposal builds upon research that women respond more favorably to approaches that emphasize the broader, practical applicability of computer science as a discipline beyond merely programming. Teague and Clarke first diagnosed such a misinformed view of computer science in their survey of Australian high school girls in 1993. When asked what the students thought about computing as a career, none of the girls "had a clear idea about computing careers," assuming that such careers entailed "lots of programming ... and working alone with a computer." [34] Such a widespread lack of understanding of computer science among high school girls still persists today and is by no means limited to Australia. Jim Foley, Board Chair of the Computing Research Association, announced in the May, 2005, issue of *Computing Research News* that the CRA has formed a new *Image of Computing Task Force*, whose expressed goal is to "help high school students and others know there is more to computing than AP programming." [10] In the same issue of the *Computing Research News*, Lenore Blum and Carol Frieze similarly underscored the importance of addressing high school students' perceptions of computer science, [3] arguing that "The fundamental misconception, of course, is that computer science equals programming." We want to explicitly leverage students' interest in mathematics to broaden their conception of computer science as a discipline. Other educators have long advocated such an approach to introductory courses in computer science, including Henderson's 1990 discussion of the merits of offering discrete mathematics as a bridge between traditional high school mathematics and computer science courses. [14] More recently, De Palma has explicitly suggested [28] that computer science educators pay more attention to the comparatively equitable distributions of female undergraduate mathematics majors, who since 1970 have consistently "receive[d] almost half of the undergraduate degrees in mathematics." As a solution to the eminent "shrinking pipeline" crisis, De Palma's recommendation is "to make computer science more attractive to women, make it more like mathematics."

Our program will broaden interest in computer science by engaging students in the wider, "real-world" applicability of computer science via social networking applications. The importance of rooting abstract discussions of computer science in more familiar and accessible experiences has repeatedly been recognized as a crucial aspect in increasing women's interest in computer science. In 1996, Liu and Blanc reported [17] that in their survey of Cal Poly State University female computer science/engineering majors their respondents were "motivated by the functionality of what is being taught," adding that "[a] large number of women expressed the desire to be shown the relevance of the course material to 'the real world'... because they could not see themselves programming for the rest of their lives, simply for the sake of programming." The data of Margolis and Fisher [20] similarly supports this trend in that "[f]orty-four percent of the women we interviewed and nine percent of the men link their interest in computing to other arenas."

In basing our alternative approach to computer science in appealing, real-word examples, it is important to attract: under-represented groups, especially women. Given the recent explosion of interest in social networking applications, we anticipate that our proposal will offer an extremely compelling vehicle. In the wake of the immediately popular `friendster.com` founded in 2002, a plethora of competitor sites, such as `tribe.net`, `orkut.com`, `linkedin.com`, `myspace.com`, `spoke.com`, `ryze.com`, `meetup.com`, `emode.com`, and `thefacebook.com`, have drawn massive numbers of young users to their communities. According to research gathered from comScore Media Matrix, 16.9 million Americans visited at least one of four (`myspace.com`, `friendster.com`, `bebo.com`, and `thefacebook.com`) social networking sites, with the current largest of the four, `myspace.com` drawing nearly 9 million visitors in a single month. [21] Even more strikingly, they report that "Perhaps unsurprisingly, these sites tend to be particularly popular among younger users. Visitors to Friendster and MySpace are nearly twice as likely as the average Internet user to be age 18 to 24. Visitors to TheFaceBook and MySpace are slightly more likely to be female, while Friendster skews slightly toward males." [21] "Blogs and their technology cousins, social networking sites, are all the rage among young internet users. About half of all blogs are authored by teenagers, according to a 2003 study by Perseus Development Corp.; and according to comScore Media Metrix, a majority of the top 15 sites visited by teens 17 and under in January 2005 were either blogs or social networking sites." [33] These observations regarding the appeal of online social spaces to high school students is consistent with daily statistics logs at `livejournal.com`, one of the most popular blog community sites. Of the 7,290,097 registered blog accounts at `livejournal.com`, the latest statistics show that 67.3% of the regularly maintained blog accounts are registered to women. In considering age demographics, the age distribution is heavily weighted towards the teenage population, with a total of 1,354,024 of the sites (46% of the total number) belonging to members between the ages of 13-18: [18]

4. EXAMPLES OF MODULE CONTENT

Examples of some of the modules we are working on appear in the sections that follow. We are in the beginning stages of this project and we plan on making the full modules and assignment available to the community in addition to reporting on how they work in our classes.

4.1 Power Law Distributions

In our course for non-majors we have illustrated Zipf's law using a *how many people named Smith* experiment from [11]. We illustrate the same law in our CS2 course by asking students to plot word occurrences from all of Shakespeare's plays. We plan on developing modules based on this material that could be incorporated into Algebra II in explaining both exponential and logarithm functions. The example provides a motivating use of computational power with algorithmic tradeoffs in how frequencies are calculated, maintained, and visualized. The fit of a curve easily fits into statistics courses at all levels. We will continue with power laws, building on examples from [29] to show how simulation, analytic solutions, and real data for category-specific web pages (e.g., university homepages) converge to show interesting distributions, being unimodal on a log-scale. This can lead to questions for courses ranging from Algebra II, to Discrete Math, to statistics to programming. We are developing background reading, questions, and stories for each of these different courses all based on Zipf's law and power law distributions. These materials also motivate computational methods, algorithms, and tools for understanding the web.

4.2 From Erdös to Bacon to bin Laden

Bacon Numbers at `http://oracleofbacon.org` provide a well-known and popular example of the *six degrees of separation* [38] model of social networks familiar to academics in part due to Erdös numbers which are recognized by Newman [22, 27, 26, 23] as a *social network* In [36] the concept of *betweenness centralization* is formalized. This is termed *funneling* in [27] in the context of academic collaboration. *... are all of your collaborators equally important for your connection to the rest of the world, or do most paths from others to you pass through just a few of your collaborators. ... Collaboration with just one or two senior or famous members of one's field could easily establish short paths to a large part of the collaboration network, and all of these short paths would go through those one or two members.*

This concept is analyzed in [19] in the context of a community of dolphins in New Zealand. The temporary disappearance of a "well-placed" dolphin leads to a fission in the dolphin community. The re-appearance of the same dolphin leads to the reunion of the two communities back into one community.

This same concept appears in the context of terrorist networks [30] and other similar so-called *Dark Networks*. Removing the key players can cause the dissolution of a network, and network analysis helps identify the key players.

All these examples lead to a discussion of networks and network science in courses ranging from discrete/finite math courses to more traditional programming courses. We have tested a module built on these concepts in a seminar on problem-solving intended for students with no experience in programming or computer science, but who will take such courses as part of a year-long experience for such students. We ask these students to understand the difference between popularity as measured by number of links or friends and centrality as measured by shortest, average minimal-path length to connected vertices. Students have been motivated to understand the examples based on their experiences with `duke.thefacebook.com`. They honed their problem-solving skills in trying to create graphs whose center was not the vertex with the greatest degree.

We are continuing to refine examples in this domain leveraging real-world data from `thefacebook.com` as the source of interest for students.

4.3 Collaborative Filtering and Visualization

Software Tools, such as such as Krackplot [16] UCINET and Netdraw [4] TouchGraph [32], NetVis [6] and Pajek [8] help visualize and reason about social networks and will be an important component of the modules we are developing. For example, using Amazon's *Collaborative Filtering* techniques, which are made available via the TouchGraph site, we can view books and related articles as well as visualizing connections based on the "people that bought this also bought that" data available from Amazon. Collaborative filtering is an example of statistical techniques that form the basis of much current research in data mining and artificial intelligence, students will see the power of these techniques in a familiar context. We are working on integrating these visualizations into the modules and tools under development.

We have tested an assignment using a recommender system from `www.last.fm` and `audioscrobbler.net` called Audioscrobbler. The sytem tracks users' listening habits, uses collaborative filtering to find users with similar musical interests, and suggest new tracks and artists. As part of the Duke Digital Initiative students in our course are given Apple iPods which we integrate into course assignments. In this case, they listen to songs on their iPods, then upload a list of recently-played songs to the Audioscrobbler database automatically via a plugin. All students in our class are part of the same Audioscrobbler group, which the site facilitates. Students then chart and predict their behavior and that of their musical "neighbors" to get recommendations and understand how algorithms for the system might work. Neighbors are based on a *match value* which is calculated using a proprietary algorithm. As part of the assignment, students hypothesize about how the match values are calculated and design an experiment to test their hypotheses. By changing their listening habits and noticing differences in rankings, students learn about recommender systems and collaborative filtering.

Reasoning about networks may be possible using visualizations for small networks, but analysis tools like Pajek [8] will be necessary to reason about larger networks. Our modules will allow students to reason about networks at many levels fostering an appreciation for the tools and techniques that computer science makes available for coping with large data sets.

5. OUTREACH

We hope to situate our modules and approach in high schools. Computer Science courses in high schools have traditionally been grounded in programming, this is certainly the case for AP Computer Science. Recent efforts by the ACM and the newly formed Computer Science Teachers Association (CSTA) have led to the formation of a new high school curriculum. We plan to develop our modules so that they will fit into existing math, statistics, and computer science courses in high schools and mesh well with curricula currently in place. Our materials and modules can be part of two areas from the ACM K-12 Model Curriculum [35] *Level III*: Discrete Math and *Projects based courses*: emerging technologies.

At the same time, we are not targetting only computer science courses for outreach, but the growing population of students in statistics courses. As one example of the differences in how courses in programming and statistics differ in high schools, consider the descriptions of the Advanced Placement versions of these courses from `apcentral.collegeboard.com`.

> **Teaching the Course** The AP Statistics course lends itself naturally to a mode of teaching that engages students in constructing their own knowledge. For example, students working individually or in small groups can plan and perform data collection and analyses where the teacher serves in the role of a consultant, rather than a director. This approach gives students ample opportunity to think through problems, make decisions, and share questions and conclusions with other students as well as with the teacher.
>
> Important components of the course should include the use of technology, projects and laboratories, cooperative group problem-solving and writing, as a part of concept-oriented instruction and assessment. This approach to teaching AP Statistics will allow students to build interdisciplinary connections with other subjects and with their world outside school.

Contrast the description above with the description of computer science courses below.

> **The Courses** Because the development of computer programs to solve problems is a skill fundamental to the study of computer science, a large part of the course is built around the development of computer programs or parts of programs that correctly solve a given problem. The course also emphasizes the design issues that make programs understandable, adaptable, and, when appropriate, reusable.
>
> **Teaching The Courses**
>
> The teacher should be prepared to present a college-level first course in computer science. Each AP Computer Science course is more than a course on programming. The emphasis in these courses is on procedural and data abstraction, object-oriented programming and design methodology, algorithms, and data structures.
>
> Because of the dynamic nature of the computer science field, AP Computer Science teachers will continually need to update their skills. ... Teachers should endeavor to keep current in [the evolution of programming languages] by investigating different programming languages.

The tone of these descriptions corresponds to the the view of many high school teachers: statistics is more relevant, easier to teach, and more approachable by both faculty and students. Our proposed modules should help bring these characteristics to computer science courses at all levels.

6. FUTURE WORK

We are in the beginning stages of what we hope is a long-term project to develop and use materials related to social networks to make computer science more attractive to a larger audience. As our materials are developed we will make them available at `www.cs.duke.edu/csed/socialnet`.

7. REFERENCES

[1] `http://apcentral.collegeboard.com`, 2005.

[2] Albert-László Barabási. *Linked: The New Science of Networks*. Perseus Publishing, 2002.

[3] Lenore Blum and Carol Frieze. Expanding the pipeline: In a more balanced computer science environment, similarity is the difference and computer science is the winner. *CRA Newsletter*, May 2005.

[4] Steve Borgatti. `http://www.analytictech.com`.

[5] Kathleen M. Carley. Casos. `http://www.casos.cs.cmu.edu`.

[6] Jonathon Cummings. NetVis module - dynamic visualization of social networks, 2005.

[7] Wanda P. Dann, Stephen Cooper, and Randy Pausch. *Learning to Program with Alice*. Prentice Hall, 2005.

[8] Wouter de Nooy, Andrej Mrvar, and Vladimir Batagelj. *Exploratory Social Network Analysis with Pajek*. Cambridge University Press, 2005.

[9] A. Fisher, J. Margolis, and F. Miller. Undergraduate women in computer science: Experience, motivation and culture. *The Papers of the Twenty-Eighth SIGCSE Technical Symposium on Computer Science Education*, 29(1):106–110, 1997.

[10] Jim Foley. Computing, we have a problem: Message from the chair. *CRA Newsletter*, May 2005.

[11] Malcolm Gladwell. *The Tipping Point*. Little Brown, 2000.

[12] Mark Guzdial. A media computation course for non-majors. *Proceedings of the ITiCSE Conference*, 35(3):104–108, 2003.

[13] Mark Guzdial. *Introduction to Computing and Programming in Python: A Multimedia Approach*. Pearson Prentice Hall, 2005.

[14] Peter Henderson. Discrete mathematics as a precursor to programming. *The Papers of the Twenty-First SIGCSE Technical Symposium on Computer Science Education*, 22(1):17–21, 1990.

[15] Lily Irani. A different voice: Exploring stanford computer science. Master's thesis, Stanford University, 2003.

[16] David Krackhardt. KrackPlot: a social network visualization program. `http://www.andrew.cmu.edu/user/krack/krackplot/krackindex.html`.

[17] M. Liu and L. Blanc. On the retentention of femail computer science students. *The Papers of the Twenty-Seventh SIGCSE Technical Symposium on Computer Science Education*, 28(1):32–36, 1996.

[18] `http://www.livejournal.com/stats.bml`.

[19] David Lusseau and M.E.J. Newman. Identifying the role that individual animals play in their social network. *Proceedings of the Royal Society Biology Letters*, 2004.

[20] M. Margolis and A. Fisher. *Unlocking the Clubhouse: Women in Computing*. MIT Press, 2002.

[21] `http://www.imediaconnection.com/content/5437.asp`, April 07 2005.

[22] M.E.J.Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Science*, 98(2):404–409, 2001.

[23] M.E.J.Newman, S.H. Strogatz, and D.J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(026118), 2001.

[24] Peter Monge and Noshir Contractor. *Theories of Communications Networks*. Oxford University Press, 2003.

[25] Barbara Moskal, Deborah Lurie, and Stephen Cooper. Evaluating the effectiveness of a new instructinal approach. *The Papers of the Thirty-fifth SIGCSE Technical Symposium on Computer Science Education*, pages 75–79, 2004.

[26] M.E.J. Newman. Scientific collaboration networks.i. network construction and fundamental results. *Physical Review E*, 64(016131), 2001.

[27] M.E.J. Newman. Scientific collaboration networks.ii. shortest paths, weighted networks, and centrality. *Physical Review E*, 64(016132), 2001.

[28] P. De Palma. Viewpoint: Why women avoid computer science. *Communications of the ACM*, 44(6):27–29, June 2001.

[29] David M. Pennock, Gary W. Flake, Steve Lawrence, Eric J. Glover, and C. Lee Giles. Winner's don't take all: Characterizing the competition for links on the web. *Proceedings of the National Academy of Science*, 99(8):5207–5211, 2002.

[30] Jörg Raab and H. Brinton Milward. Dark networks as problems. *Journal of Public Administration Research and Theory*, 13(4):413–439, 2003.

[31] L. Rich, H. Perry, and M. Guzdial. A CS1 course designed to address interests of women. *The Papers of the Thirty-Fifth SIGCSE Technical Symposium on Computer Science Education*, 36(1):190–194, 2004.

[32] Alexander Shapiro. Touchgraph. `http://www.touchgraph.com`.

[33] Bob Sullivan. Kids, blogs and too much information. MSNBC, April 29 2005. `http://www.msnbc.msn.com/id/7668788`.

[34] G.J. Teague and V.A. Clarke. Attracting women to tertiary computing courses: Two programs directed at secondary level. *The Papers of the Twenty-Fourth SIGCSE Technical Symposium on Computer Science Education*, 25(1):208–212, 1993.

[35] Allen Tucker, editor. *A Model Curriculum for K-12 Computer Science*. CSTA/ACM, 2003. `http://www.acm.org/education/k12/k12final1022.pdf`.

[36] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[37] Duncan Watts. *Small Worlds*. Princeton University Press, 1999.

[38] Duncan Watts. *Six Degrees: The Science of a Connected Age*. W.W. Norton, 2003.