Computing in SCIENCE & ENGINEERING

Vol. 18, No. 3 | May/June 2016

IEEE

AIP
cise.aip.org

IEEE computer society
CELEBRATING 70 YEARS
www.computer.org/cise/

Computing in SCIENCE & ENGINEERING

Vol. 18, No. 3 | May/June 2016

IEEE

AIP
cise.aip.org

IEEE computer society
CELEBRATING 70 YEARS
www.computer.org/cise/

computing in SCIENCE & ENGINEERING

EDITORIAL OFFICE

Publications Coordinator: cise@computer.org

COMPUTING IN SCIENCE & ENGINEERING
c/o IEEE Computer Society
10662 Los Vaqueros Circle, Los Alamitos, CA 90720 USA
Phone +1 714 821 8380; Fax +1 714 821 4010
Websites: www.computer.org/cise or http://cise.aip.org/

May/June 2016, Vol. 18, No. 3

Computing in SCIENCE & ENGINEERING

BEST OF RESPECT, PART 2

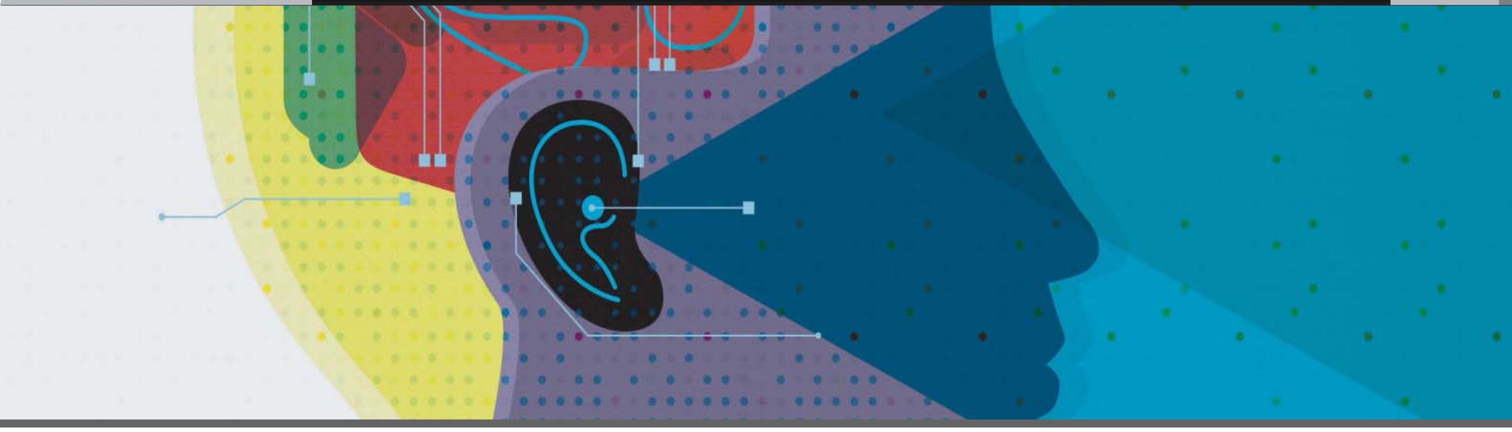
Cover illustration: Andrew Baker
www.debutart.com/illustration/andrew-baker

STATEMENT OF PURPOSE

Computing in Science & Engineering (*CiSE*) aims to support and promote the emerging discipline of computational science and engineering and to foster the use of computers and computational techniques in scientific research and education. Every issue contains broad-interest theme articles, departments, news reports, and editorial comment. Collateral materials such as source code are made available electronically over the Internet. The intended audience comprises physical scientists, engineers, mathematicians, and others who would benefit from computational methodologies. All articles and technical notes in *CiSE* are peer-reviewed.

For more information on these and other computing topics, please visit the IEEE Computer Society Digital Library at www.computer.org/csdl.

SUSTAINABLE FORESTRY INITIATIVE Certified Sourcing www.sfiprogram.org SFI-01042



COLUMNS

DEPARTMENTS

RESOURCES

Editorial: Unless otherwise stated, bylined articles, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *Computing in Science & Engineering* does not necessarily constitute endorsement by IEEE, the IEEE Computer Society, or the AIP. All submissions are subject to editing for style, clarity, and length. IEEE prohibits discrimination, harassment, and bullying. For more information, visit www.ieee.org/web/aboutus/whatis/policies/p9-26.html. **Circulation:** *Computing in Science & Engineering* (ISSN 1521-9615) is published bimonthly by the AIP and the IEEE Computer Society. IEEE Headquarters, Three Park Ave., 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Cir., Los Alamitos, CA 90720, phone +1 714 821 8380; IEEE Computer Society Headquarters, 2001 L St., Ste. 700, Washington, D.C., 20036; AIP Circulation and Fulfillment Department, 1NO1, 2 Huntington Quadrangle, Melville, NY, 11747-4502. Subscribe to *Computing in Science & Engineering* by visiting www.computer.org/cise. **Reuse Rights and Reprint Permissions:** Educational or personal use of this material is permitted without fee, provided such use: 1) is not made for profit; 2) includes this notice and a full citation to the original work on the first page of the copy; and 3) does not imply IEEE endorsement of any third-party products or services. Authors and their companies are permitted to post the accepted version of IEEE-copyrighted material on their own web servers without permission, provided that the IEEE copyright notice and a full citation to the original work appear on the first screen of the posted copy. An accepted manuscript is a version that has been revised by the author to incorporate review suggestions, but not the published version with copy-editing, proofreading and formatting added by IEEE. For more information, please go to: http://www.ieee.org/publications_standards/publications/rights/paperversionpolicy.html. Permission to reprint/republish this material for commercial, advertising, or promotional purposes or for creating new collective works for resale or redistribution must be obtained from IEEE by writing to the IEEE Intellectual Property Rights Office, 445 Hoes Lane, Piscataway, NJ 08854-4141 or pubs-permissions@ieee.org. Copyright © 2016 IEEE. All rights reserved. **Abstracting and Library Use:** Abstracting is permitted with credit to the source. Libraries are permitted to photocopy for private use of patrons, provided the per-copy fee indicated in the code at the bottom of the first page is paid through the Copyright Clearance Center, 222 Rosewood Dr., Danvers, MA 01923. **Postmaster:** Send undelivered copies and address changes to *Computing in Science & Engineering*, 445 Hoes Ln., Piscataway, NJ 08855. Periodicals postage paid at New York, NY, and at additional mailing offices. Canadian GST #125634188. Canada Post Corporation (Canadian distribution) publications mail agreement number 40013885. Return undeliverable Canadian addresses to PO Box 122, Niagara Falls, ON L2E 6S8 Canada. Printed in the USA.

by Douglass E. Post
Associate Editor in Chief

The Periodic Table of Elements, an Early Example of "Big Data"

The latest fad in computing seems to be "big data." I got an email ad today that proclaimed, "Last Chance: Fast Data—The New Big Data." It's only one of the many ads that most of us get every day from conference organizers, contractors, and others implying that our company/university/laboratory—and we personally—will miss out on the latest "gold rush" unless we come to their conference or engage their services to learn: "What is 'big data'? What can we do with it? Why we can't afford to ignore its potential. How it will revolutionize our world. How we can make a lot of money with it." And so on. Much of this is about social media data, credit card fraud detection, polling data, machine learning, or making money off other people's data. But it has a great deal of promise for scientific research. Given the deluge of data that is becoming available due to ever-better sensors, the rapidly declining cost of data storage, and the exponential growth in computing power since World War II, analyzing and mining data collections to provide information to guide decisions has become important and even potentially lucrative.

However, a great deal of caution is necessary. I don't think it's true, as someone remarked, that "we no longer need to understand what's going on, we only have to look it up with a search engine." One drawback of relying on data analysis alone without understanding the underlying scientific principles is that it's very easy to confuse correlation with causation. Type "Correlation does not imply causation" into your favorite search engine, and the Wikipedia article that comes back describes a number of examples of current medical and social science experiments in which correlations were used to support actions that turned out to be counterproductive once further experiments were conducted that proved that the correlations were spurious. However, a reasonable sense of perspective is needed. The correct analysis of valid data can lead to useful decision data, but until the underlying processes and causes are identified, those decisions can only be viewed as tentative, subject to further revision. Truly reliable predictions can only be made from an understanding of the basic principles of nature, not from raw data.

An early example is the cholera epidemic in London in 1854. The prevailing medical understanding was that disease was caused by miasma (bad air), which didn't suggest any actions could be taken to stop the epidemic. John Snow suspected a causal relationship between poor sanitation and cholera, but he couldn't prove it. He supported this theory by taking a map of London and drawing a point on it at the location of each known cholera case. This way of plotting the data showed that a large portion of the cholera victims were located in an area where people drew their water from the Broad Street public water pump. The pump was padlocked shut, and the epidemic was stopped, although the outbreak might have been in decline by then. This action, though effective, was highly controversial at the time. The data supported it but didn't directly prove that polluted water was the cause of the epidemic until 1886, when the discovery of the bacterium *Vibrio cholerae* confirmed Snow's theory. The statistical data was suggestive (and highly useful) but not conclusive. Snow was initially severely criticized at the time but is now viewed as one of the founders of epidemiology. His methods weren't generally adopted until after the identification of the cholera bacterium.

While big data analysis is being used for fraud detection, the identification of cultural trends and potential customers, predictions of credit worthiness, and so on

 1521-9615/16/$33.00 © 2016 IEEE Copublished by the IEEE CS and the AIP

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1 H 1.01																	2 He 4.0
3 Li 6.94	4 Be 9.01											5 B 10.81	6 C 12.01	7 N 14.01	8 O 16.00	9 F 19.00	10 Ne 20.18
11 Na 22.99	12 Mg 24.31											13 Al 26.98	14 Si 28.09	15 P 30.97	16 S 32.07	17 Cl 35.45	18 Ar 39.95
19 K 39.10	20 Ca 40.08	21 Sc 44.06	22 Ti 47.88	23 V 50.94	24 Cr 52.00	25 Mn 54.94	26 Fe 55.85	27 Co 58.93	28 Ni 58.69	29 Cu 63.55	30 Zn 65.39	31 Ga 69.72	32 Ge 72.61	33 As 74.92	34 Se 79.96	35 Br 79.90	36 Kr 83.80
37 Rb 85.47	38 Sr 87.62	39 Y 88.91	40 Zr 91.22	41 Nb 92.91	42 Mo 95.94	43 Tc 98	44 Ru 101.1	45 Rh 102.9	46 Pd 106.4	47 Ag 107.9	48 Cd 112.4	49 In 114.8	50 Sn 118.7	51 Sb 121.8	52 Te 127.6	53 I 126.9	54 Xe 131.3
55 Cs 132.9	56 Ba 137.3	57 –71	72 Hf 178.0	73 Ta 180.9	74 W 183.4	75 Re 186.2	76 Os 190.2	77 Ir 192.2	78 Pt 195.1	79 Au 197.0	80 Hg 200.6	81 Tl 204.4	82 Pb 207.2	83 Bi 209.0	84 Po 209	85 At 210	86 Rn 222
87 Fr 223	88 Ra 226	89 –103	104 Rf 267	105 Db 268	106 Sg 271	107 Bh 272	108 Hs 270	109 Mt 276	110 Ds 281	111 Rg 280	112 Cn 285	113 Uut 289	114 Fl 289	115 Uup ?	116 Lv 293	117 Uus ?	118 Uuo ?

Atomic number
Symbol
Atomic weight

57 La 138.9	58 Ce 140.1	59 Pr 140.9	60 Nd 144.2	61 Pm 145	62 Sm 150.4	63 Eu 152.0	64 Gd 157.3	65 Tb 158.9	66 Dy 162.5	67 Ho 164.9	68 Er 167.3	69 Tm 168.9	70 Yb 173.0	71 Lu 175.0
89 Ac 227	90 Th 232.0	91 Pa 231.0	92 U 238.3	93 Np 237	94 Pu 244	95 Am 243	96 Cm 247	97 Bk 247	98 Cf 251	99 Es 252	100 Fm 257	101 Md 258	102 No 259	103 Lr 262

Figure 1. Dimitri Mendeleev's periodic table of elements. Sc, Ga and Ge, missing in Mendeleev's original table, are highlighted in yellow.

that don't have much to do with science and engineering, data analysis (big and small) plays a key role in the scientific method. For the rest of this essay, I'll concentrate on the analysis of scientific data. After all, this is an issue of *Computing in Science & Engineering*, not a discussion of social networks or fraud detection. *CiSE* has even devoted several special issues to the topic (such as our "Science Data Management" special issue in May/June 2013).

All of this attention and promotion tends to evoke healthy skepticism, but as I watched a PBS series entitled *The Mystery of Matter*, which aired on 19 August 2015, I realized that a good example of a small-scale version of big data is familiar to all of us, namely, the periodic table of elements (Figure 1). The video was about the history of Dimitri Mendeleev's invention of the table, the chart that hangs on the wall of every chemistry lab and classroom in the world (and probably in the universe). To me, the periodic table offers several interesting lessons about data and its value and impact that are relevant to today's big data. Science and engineering are, at their core, based on the analysis of data whose goal is the identification of the underlying causes of the phenomena being studied. It's not sufficient to see patterns in the data and act on them. A scientist or engineer uses the patterns in the data to develop a hypothesis for the causes that lead to the observed patterns in the data. This understanding is captured in scientific theories that have predictive power, something that the history of the periodic table shows eloquently.

Discoveries in the 19th Century

Mendeleev was a chemistry professor at St. Petersburg University who was writing a chemistry textbook for his courses. He wanted to present the 63 known elements (circa 1869) in a coherent and structured way for his students. Earlier in the 19th century, From the chemical and physical properties of the neighboring elements, Mendeleev was able to predict the density and many other physical and chemical properties of the elements that would fill those gaps.

chemists had identified several organizing principles for the elements, including their atomic weight (a measure of their weight compared to hydrogen), their reactivity, the temperatures at which they melted and boiled, and their density. Elements with similar chemical properties could be grouped together and ordered by atomic weight. Building on earlier work, John Newlands organized 62 elements into 8 groups (circa 1863), but he included some non-elements and some inaccurate atomic weights, and he didn't allow room for undiscovered elements.

In 1869, Mendeleev's periodic table corrected the earlier work in several fundamental ways. His periodic table was more complete and more accurate than prior tables. Even more importantly, he put the elements in the right places in the table, organizing them as an ordered system that motivated chemists and physicists to ask fundamental questions about why the elements were related this way. And best of all for his credibility, his table had predictive power: it had gaps where there was no known element. From the chemical and physical properties of the neighboring elements, Mendeleev was able to predict the density and many other physical and chemical properties of the elements that would fill those gaps. In particular, he predicted that there should be elements that corresponded to silicon, aluminum, and boron. He called them eka-silicon (germanium), eka-aluminum (gallium), and eka-boron (scandium) and left space for them in his chart.

There were other missing elements as well, but by 1886, these three elements had been discovered and their properties measured. Gallium was discovered by Paul-Emile Lecoq de Boisbaudran in 1875. However, Lecoq de Boisbaudran's measured density and atomic weight differed significantly from Mendeleev's predictions. Mendeleev had the confidence to suggest that Lecoq de Boisbaudran repeat his measurements, which he did (the new measurements agreed with Mendeleev's predictions), and gallium's density at room temperate has remained at 5.91 g/cm^3 ever since. Scandium was discovered in 1879 and germanium in 1886. The later discovery of the noble gases appeared at first not to fit into the periodic table's organizational scheme. However, they soon found a home to the right of the halogens completing each row of the table.

Moving Forward in the 20th Century and Beyond

In 1913, Henry Moseley used X-ray spectroscopy to point out that the elements were ordered by atomic number (the positive charge in the atomic core), a more tidy and suggestive scheme than atomic weight. The periodic table focused attention on the ways that atoms could combine to form molecules, leading to a much deeper understanding of the role of valences in bonding. With the discovery of the electron by J.J. Thomson in 1897 and the nucleus by Ernest Rutherford in 1911, the nuclear and electronic structure of atoms became clearer.

The invention of quantum mechanics in the early 20th century led to further advances in the understanding and application of atomic and molecular physics. Several elements on the periodic table that were still missing were identified as radioactive isotopes, leading to the development of nuclear physics and eventually nuclear power. The focus on the properties of the nucleus led to investigations of its structure and to the investigation and discovery of the constituent particles of protons and neutrons (quarks, and so on). In the 1940s through the 1960s, Glen Seaborg and others pioneered the extension of the periodic table focusing on the creation and understanding of the transuranic elements. The final result is that the seventh row of the periodic table has recently been filled out by joint experiments in laboratories at Dubna, Livermore, and Oak Ridge.

The periodic table has played a major role in motivating and inspiring research and discovery about the materials the world is made of since Mendeleev first published it in 1869. Our understanding of materials is greatly advanced over Plato's original concepts of earth, water, air, and fire as the fundamental elements of matter. Mendeleev's table has clearly played an important role in that advancement.

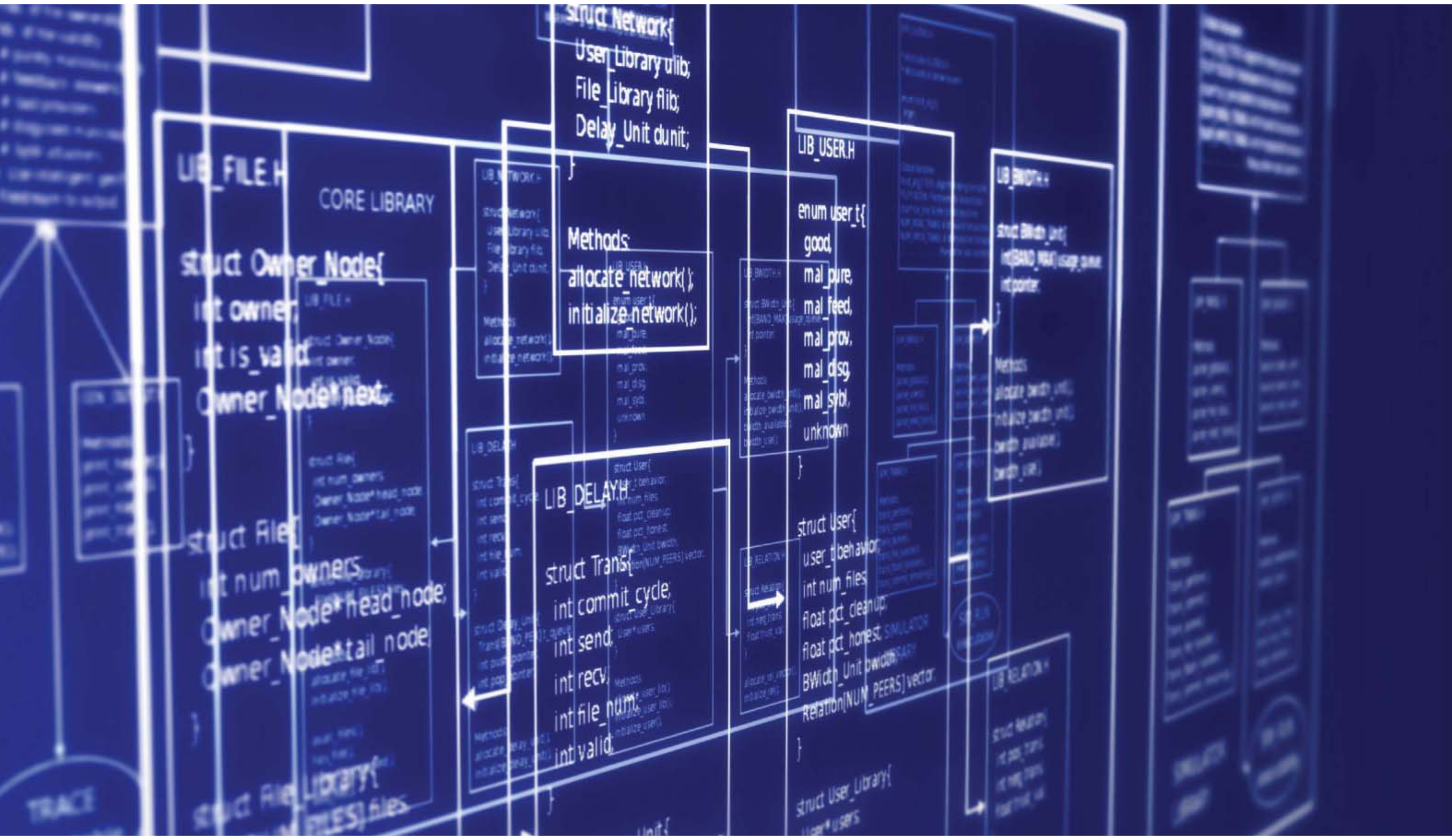
Beyond its original purpose, we can take some points away from its history for big data as it relates to science and engineering:

- The data collection should be as complete as possible. Mendeleev's periodic table was more complete than prior tables.
- The data must be correct, which might require some flexibility. Measurements can be wrong or incomplete—Mendeleev took into account the uncertainties of the atomic weight measurements and didn't always follow the recommended atomic weights when he thought they weren't consistent. As a result, he was able to put the data in the right places and thus see important patterns more easily.
- It helps to group items with common properties together. Mendeleev's grouping elements together with common chemical and physical properties facilitated the identification of key features of atomic and molecular structure.
- Sequence the data where possible. Sequencing the elements in atomic weight turned out to be a good start; sequencing by atomic number was even better.
- Identify gaps in the data that suggest the possibility that new important data could exist. Mendeleev predicted the properties of undiscovered elements that were in the gap. When they were discovered, the credibility of his arrangement of the elements was solidly established.
- Be alert for emergence of new data. Modify your data arrangement to include it in your scheme. If you can't, maybe you need a new scheme.
- Look for patterns that suggest research questions whose answers can improve our understanding of the causes of those patterns. The periodic table was a key motivator of research that led to the understanding of chemical bonding, quantum mechanics, atomic and molecular structure, materials science, nuclear physics, and particle physics, all key elements of our understanding of the material universe in which we live.

Above all, look for patterns that suggest research questions whose answers can lead to understanding the causes of those patterns. The periodic table was a key motivator of research that led to the understanding of chemical bonding, quantum mechanics, atomic and molecular structure, materials science, nuclear physics, and particle physics, all key elements of our understanding of the material universe in which we live. ■

Douglass E. Post is an associate editor in chief of *Computing in Science & Engineering*. His day job is associate director for CREATE with the DoD High Performance Computing Modernization Program. He is on loan to the DoD as an IPA from the Carnegie Mellon University Software Engineering Institute. Contact him at post@ieee.org.

Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page



Python and Physical Modeling

By Kevin Thielen and Vivienne Tien

Programming has rapidly become an essential tool for nearly all students in the physical sciences. One of the most widely used languages in this realm is Python, which has swiftly gained popularity due to its readability and intuitive syntax. The core philosophy or "Zen of Python" dictates, "simple is better than complex, complex is better than complicated" (https://www.python.org/dev/peps/pep-0020/). Because of this motto, learning how to use Python effectively on your own is a very doable task, given the right resources.

This is where *A Student's Guide to Python for Physical Modeling* by Jesse M. Kinder and Philip Nelson comes in. The text serves as an excellent stepping stone into the world of using Python in computational science for undergraduate students with a strong background in mathematics. After working through the chapters and their accompanying exercises, readers can expect not only to know how to write and read Python but also to achieve a thorough understanding for developing complex physical models and calculations.

J.M. Kinder and P. Nelson, *A Student's Guide to Python for Physical Modeling*, Princeton Univ. Press, 2015.

Approachability and Organization

From the get-go, this aptly named "student's guide" presupposes no prior knowledge of programming. Appendix A contains instructions on how to install, launch, and set up everything Python that you'll need for this book, including the suggested development environment (Spyder) and relevant packages (such as Anaconda). For slightly more experienced users, the text offers advice and information on extensions for auto updating, acceleration, and FFmpeg. The book accounts for backward incompatibility (Python 2 and 3) in Appendix B, shedding light—in simple terms—on how the reader can overcome errors raised by this issue.

1521-9615/16/$33.00 © 2016 IEEE Copublished by the IEEE CS and the AIP

Much of the first chapter, "Getting Started with Python," is written for those who have little or no background in programming in general. It establishes basic ideas such as algorithmic thinking, clarifies the use and nature of algorithms by comparing and contrasting them with mathematical proofs, and discusses and demonstrates the use of common mathematical symbols in Python. In the last few pages, it succinctly introduces the use of modules, error resolution, variable creation, and functions through try-it-yourself snippets of code.

Learning through experimentation is highly emphasized in this book, which states outright that "reading this tutorial won't teach you Python. You can teach yourself Python by working through all the examples and exercises here, and then using what you've learned on your own problems." New concepts are almost always introduced first through coding exercises, then explained and elaborated upon afterward. The authors also provide and encourage readers to use online resources such as stackoverflow.com, where Python programmers go to resolve errors, find more efficient methods, and learn new methods.

The language used throughout the book is simple and conversational, with concepts presented and explained in a much less intimidating manner than, say, that used in a conventional textbook. The text is very thoughtfully organized and covers the essentials for physical modeling. Sometimes, examples of the application of new concepts appear before their explanation, helping students develop intuition to better understand Python constructions. The chapters are organized so that structure and control are covered first, then data handling (calculations and graphics) and functions, before using these basic concepts as a gateway to more complex Python constructions (such as contour plots, numerical solutions to nonlinear equations, vector fields, image processing, and animation).

However, in contrast to how approachable the authors make programming in Python, the rather complex physics and math models in later chapters could be confusing for undergraduate students lacking this same background, as the knowledge of these topics are assumed to be a priori.

Insights and Analysis

One of the inconveniences (probably intentional) in this book is how some chapters are challenging to follow and learn from without actively working through the examples in a Python integrated development environment (IDE) as you read. In particular, examples within chapters lack line numbers, but the text still refers to line numbers in the example, which makes it hard to follow unless you have the IDE open with the example loaded. Moreover, the text doesn't always reveal what to expect after running an example code and instead encourages readers to make the discovery themselves.

Although the book is intended for students with some or no experience with computer science, the potential for a more technical understanding of programming, and some of the more technical aspects of Python, are made available through the appendices as well as the supplied online materials. As students with some experience with Python, we found the appendices to be some of the more interesting parts of the book. Here, the authors briefly but thoroughly go over topics such as debugging as well as the more mysterious, less intuitive, but still prevalently encountered "inner workings" of Python. The appendix section on debugging categorizes and interprets some of the most common errors that Python programmers come across. Some errors of particular interest include topics such as `ZeroDivisionError` versus `RuntimeWarning`, and the manipulation of `AssertionError`.

Appendix D, titled "Under the Hood," takes a straightforward and practical approach to exploring the mystifying topic of how Python handles variables and objects internally in light of exception handling and interacting with more advanced code. This section, in layman terms, sheds light on how assignment statements for arrays when used incorrectly can sometimes result in perplexing results. Kinder and Nelson also touch on memory management, the interaction between variables and objects within functions, and finally how Python keeps track of variables using namespaces.

We found the examples and problems extremely constructive and relevant to the text. The greater than 10 lines of try-it-yourself code snippets sprinkled generously throughout the chapters is as essential as the text itself. These small test codes encourage readers to not only learn new concepts but also gain intuition on how to effectively communicate with Python. They also help build confidence by letting students write, run, and successfully debug small fragments of code. A total of 17 problems (referred to as "your turn") appear over the five chapters, with an additional three "computer lab" chapters sandwiched between sections. The difficulty of the "your turn" questions ranges from array creation and writing simple functions to multipart word problems;

Computing in SCIENCE & ENGINEERING | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page | Qmags THE WORLD'S NEWSSTAND®

Appendix E holds the answers to these problems. The computer labs are much more complex, walking readers through the creation of a physical model. They reinforce and put learned concepts from previous chapters to use for creating a more complex model.

In fact, some of our favorite models were from the second and third computer labs. The first part of the second computer lab dealt with various computations, trajectory plots, and displacement distributions for 2D random walks. The lab is structured so that assignments lead the reader through these tasks while the text sandwiched between them explains the theory, meaning, and significance of the results. The third computer lab focuses on image manipulation and explores local averaging, decreasing noise, and feature amplification through filters. We thought this was a great final lab because image distortion and editing is very familiar to the audience that the book is intended for. It's exciting to be able to create programs and explore the mathematical and computational side of science that's so relevant and prevalent in this day and age.

Overall, we feel the text offers a great primer for mathematically inclined undergraduate students (potentially advanced high school students as well) to get started in Python, programming, and physical modeling. For a full semester-long course, the book might fall somewhat short on material and rigor if used as the sole material. However, if it's used along with instructor exercises or other surplus material, the text could become a starting point from which to branch out to more sophisticated topics or augment the proffered topics with increased complexity. ■

Kevin Thielen is an undergraduate student at Eckerd College majoring in mathematics and physics. He has been exposed to using Python for physical modeling through school courses as well as through several research projects. Contact him at kdthiele@eckerd.edu.

Vivienne Tien is an undergraduate student at Eckerd College at St. Petersburg in Florida majoring in mathematics and minoring in physics. She has learned and used Python through research projects and courses. Contact her at wtien@eckerd.edu.

Best of RESPECT, Part 2

Tiffany Barnes | North Carolina State University
Jamie Payton | University of North Carolina, Charlotte
George K. Thiruvathukal | Loyola University
Kristy Elizabeth Boyer | University of Florida
Jeff Forbes | Duke University

We're delighted to bring you this special issue on the best of RESPECT, part 2! As we stated in part 1, the IEEE Special Technical Committee on Broadening Participation conference, Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT), was founded on the belief that engaging diverse groups of people in computing is a matter of equity—all people deserve the opportunity to solve increasingly complex global challenges. The inaugural RESPECT 2015 conference, held 13–14 August 2015 in Charlotte, North Carolina, was co-organized by the STARS Computing Corps BPC Alliance and collocated with the STARS Celebration to leverage and engage the existing activist-oriented community in broadening participation (BP) research. The RESPECT and Celebration conferences shared a joint theme, "RESPECT for Diversity," that you will find throughout this two-part special issue. The five articles in this second part of the two-part series include the remaining best papers from RESPECT 2015.

computing in SCIENCE & ENGINEERING | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page | Qmags THE WORLD'S NEWSSTAND®

The articles in this two-part series highlight the ways diverse populations experience and perceive computing, along with some ways to encourage engagement. These results are critical to the success of CS for All and for the field of computing in general.

The first two articles look at individual perceptions in an attempt to understand the lack of participation for diverse groups: African-American girls and Lesbian, Gay, Bisexual, Transgender, and Queer (LGBTQ) students. While women of color comprise 35 percent of the general US population (www.catalyst.org/knowledge/women-color-united-states-0), fewer than 10 percent of bachelor's in computing degrees are awarded to them (http://cacm.acm.org/magazines/2011/7/109907-the-status-of-women-of-color-in-computer-science/fulltext#UT1). The article by Ashley Robinson, Manuel Perez-Quinones, and Glenda Scales, "African-American Middle School Girls: Influence on Attitudes toward Computer Science," explores the factors that impact the attitudes of African-American middle school girls about computing. The authors found that, in line with other broader studies of middle school girls, the African-American girls participating in this study had a negative perception of computing, citing common negative stereotypes about the work and the people that perform it. Importantly, the work points to four factors that can have a positive influence on attitudes of African-American middle school girls toward computing: participation in a computing intervention, such as a workshop; the intervention content domain; the facilitation of performance accomplishments; and participant characteristics.

In further studies of individual perceptions of computing, Jane Stout and Heather Wright found that LGBTQ students with a low sense of belonging in the computing community were more likely to consider leaving the field in their article, "Lesbian, Gay, Bisexual, Transgender, and Queer Students' Sense of Belonging in Computing: An Intersectional Approach." As the theory of intersectionality would predict, women LGBTQ students reported the lowest sense of belonging among all student groups in the study samples. These results highlight the need to promote a stronger sense of community and inclusivity in computing, which is particularly important for students who are members of more than one underrepresented group.

Minority status can negatively impact perceptions of and sense of belonging in computing, but there are promising approaches for keeping diverse groups engaged in STEM degree programs. In "Julian Scholars: Broadening Participation of Low-Income, First-Generation Computer Science Majors," Gloria Childress Townsend and Kay Sloan study Julian Scholars, a scholarship program designed to recruit and retain low-income, first-generation college students into STEM degrees. Seventy-nine percent of the participating Julian Scholars have completed an undergraduate degree in a STEM discipline, and many chose to major in computer science. The authors' findings show that the program of a week-long summer research experience bridging high school and college, common classes for each cohort, mentoring, one-on-one resume and internship/research counseling, and scholarships were highly effective at engaging these students with computing.

The STARS Computing Corps has applied a community-building approach as well. In "STARS Computing Corps: Enhancing Engagement of Underrepresented Students and Building Community in Computing," Jamie Payton and her colleagues reported on the STARS Computing Corps, a national community that develops college faculty and students as leaders who work to broaden participation in computing. The key finding is that undergraduate computing students felt that STARS positively impacted them in their academic, career, and personal lives, with students from underrepresented groups experiencing the most benefit.

In addition to motivating women of color to pursue the study of computing, RESPECT authors address the issues these women face once they enter computing degree programs. In "Enacting Agency: The Strategies of Women of Color in Computing," Apriel Hodari and her colleagues examine how women of color employ their agency to apply strategies that have a direct impact on their own success in computing degree programs. Narrative analysis of interviews and case studies reveals four navigational approaches that women of color have employed to find the motivation and courage to persist in computing: acknowledging barriers to success, connecting their technical computing work to their unique personal experiences, developing soft skills and using them to address diversity and race, and creating technology to promote social activism.

In January 2016, President Obama called on Congress to fund the "CS for All" initiative that would provide all K–12 children with access to high-quality education in computing. The articles in this two-part series highlight the ways diverse populations experience and perceive computing, along with some ways to encourage engagement. These results are critical to the success of CS for All and for the field of computing in general. We invite you to attend or present your work at RESPECT 2016, which lasts from 11–13 August 2016 in Atlanta, Georgia. Find out more at http://respect2016.stcbp.org and help us develop interdisciplinary partnerships to promote CS for All. ■

Acknowledgments

We thank the USNational Science Foundation for funding the RESPECT'15 conference and the IEEE Special Technical Committee on Broadening Participation (stcbp.org) under award number CNS-1042372. RESPECT'15 was technically cosponsored by the IEEE Computer Society, and in cooperation with the ACM SIGCSE.

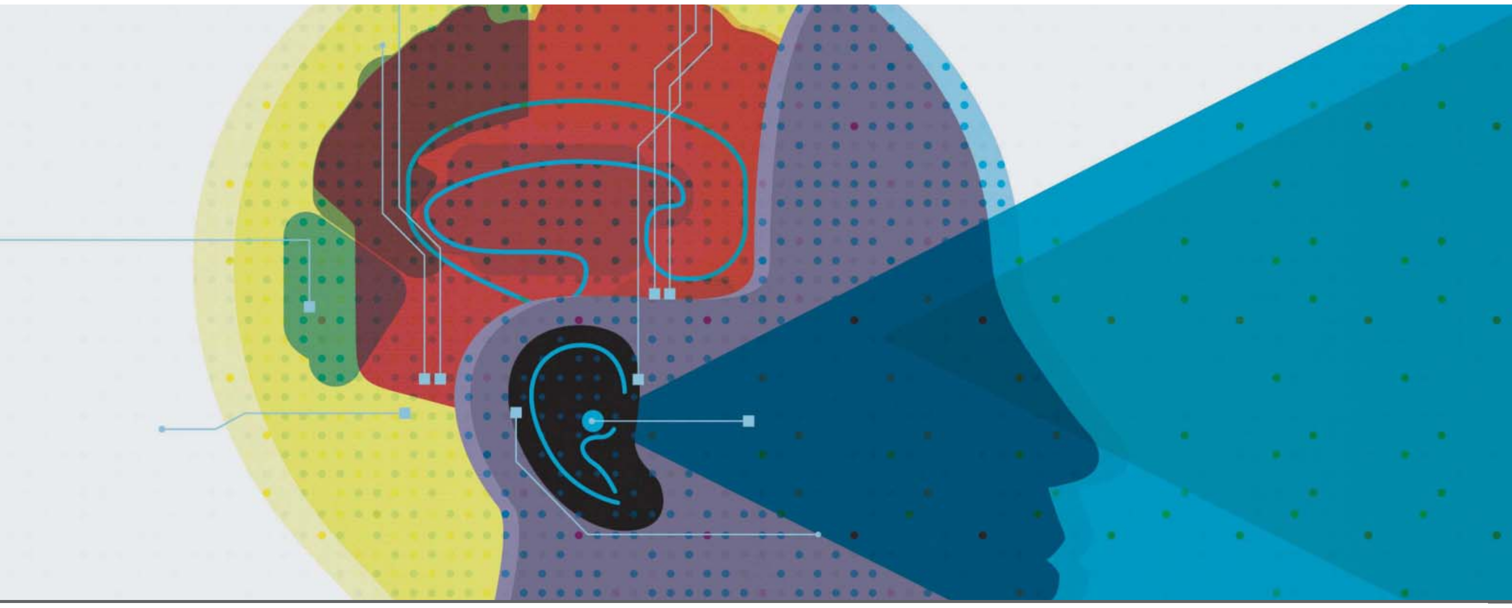
Tiffany Barnes is an associate professor in the Department of Computer Science at North Carolina State University. Her research interests include AI in education, computer science education, and broadening participation. Barnes received a PhD in computer science from North Carolina State University. Contact her at tiffany.barnes@gmail.com.

Jamie Payton is an associate professor in the Department of Computer Science at the University of North Carolina, Charlotte. Her research interests include pervasive computing, broadening participation in computing, and computer science education. Payton received a DSc in computer science from the Washington University in St. Louis. Contact her at payton@uncc.edu.

George K. Thiruvathukal is a professor in the Department of Computer Science at Loyola University; he's also visiting faculty at Argonne National Laboratory. His research interests include high-performance distributed systems, computational science, and digital humanities. Thiruvathukal received a PhD in computer science from the Illinois Institute of Technology. Contact him at gkt@cs.luc.edu.

Kristy Elizabeth Boyer is an associate professor in the Department of Computer and Information Science and Engineering at the University of Florida. Her research interests include natural language dialogue systems, intelligent tutoring, and computer science education. Boyer received a PhD in computer science from North Carolina State University. Contact her at keboyer@ufl.edu.

Jeff Forbes is an associate professor and associate dean of Trinity College of Arts and Sciences in the Department of Computer Science at Duke University. His research interests include computer science education, social information processing, and learning analytics. Forbes received a PhD in computer science from the University of California, Berkeley. Contact him at forbes@cs.duke.edu.



African-American Middle School Girls: Influences on Attitudes toward Computer Science

Ashley Robinson | Virginia Tech
Manuel A. Pérez-Quiñones | University of North Carolina at Charlotte
Glenda Scales | Virginia Tech

The number of women in computing is significantly lower than the number of men, with African-American women making up an even smaller segment of this population. A recent study reveals that African-American middle school girls generally have negative attitudes toward computer science, but that those feelings can change through intervention.

There's a lot of interest in increasing the number of women in computer science. According to the US National Science Foundation,[1] there were 47,960 computer science BS degree recipients in 2012, yet women only accounted for 18.2 percent (8,730) of the students earning those degrees. Of these women, only 1,460 were Black or African-American, adding up to just 3 percent of computer science BS degree recipients. Not only are African-American females underrepresented in computer science, increasing their numbers in computer science is an understudied topic (see the "Related Work in Camps and Workshops" sidebar). The research reported in this article helps cover this gap and can play a role in gaining insight into the attitudes of African-American middle school girls toward the field.

Study Participants

We selected participants from two national outreach organizations in the Southeast Hampton Roads area of Virginia because they represent a diverse population and already planned for summer day programs for the target age group with a non-computing-based focus. This provided the unique opportunity to gain access to females who might not be initially interested in computer science, rather than those who

 1521-9615/16/$33.00 Copublished by the IEEE CS and the AIP

voluntarily sign up for computing-based activities. Five Boys & Girls Club sites and two YMCA sites participated in the study. The research study participants consisted of 37 primarily African-American middle school females, ages 11 to 13, who attended the summer camps of the two aforementioned national outreach organizations. Among the participants, 81.1 percent were African-American, 2.7 percent were Native American, 2.7 percent were Asian, and 13.5 percent were of multiple ethnicities.

Instructional Intervention

We designed and implemented an instructional intervention consisting of two types of workshops to introduce computer science to primarily African-American middle school girls. The workshops lasted one hour per day for five days. Building on existing literature,[2] we created a user interface design and evaluation workshop (called the human-computer interaction [HCI] workshop from here forward). We also created a second type of workshop (algorithms) to represent the traditional approach to computer science and expand the breadth of data collected from the intervention. We incorporated computational thinking and problem-based learning into the instructional design of both workshops (see Figure 1 for a summary). Twenty-three girls participated in the HCI workshop, and 14 girls participated in the algorithms workshop.

Data Collection

We collected data from the participants through a background questionnaire, surveys, and focus group interviews.

Participant Background Information

Participants' parents/guardians completed a background questionnaire prior to the study. It contained one fill-in-the-blank question requesting the child's age and eight multiple-choice questions asking for the child's grade level, the child's ethnicity, types of computer devices in use at home, availability of Internet access at home, the highest education level completed by the child's mother and father, the child's free/reduced lunch status, and the child's school grades.

Computer Confidence Survey

We used a subscale of Brenda Loyd and Douglas Loyd's Computer Attitude Scale[3] to measure computer confidence. The Computer Attitude Scale consists of 40 items divided into four subscales: computer anxiety, computer confidence, computer liking, and computer usefulness. Each subscale consists of 10 items rated on a 4-point Likert scale, ranging from "strongly agree" to "strongly disagree." The items of each subscale include both positively and negatively worded items distributed throughout the instrument. Loyd and Loyd[3] indicated that the high reliability subscales for each coefficient suggest

Related Work in Camps and Workshops

Previous research has shown that interventions at the middle school level can positively influence interest in computer science. S. Khoja and colleagues[1] conducted a four-week long, four-hour per day, computer science camp for middle school girls.[1] Participants learned about a different topic every week, including robotics, programming, and hardware. The results showed that attending the camp increased student confidence and skills in using computers. E. Ashby Plant and colleagues exposed middle school girls to a 20-minute narrative delivered by a computer-generated female agent.[2] This narrative included positive statements and counteracted negative stereotypes of engineering, and the girls who watched the video reported an increased interest in engineering. Neither study controlled for ethnicity.

In Sarita Yardi and colleagues' study, 10 students, ages 11 to 13, designed a digital desktop prototype and evaluated it through user studies and heuristic evaluations.[3] Their results revealed that practicing human-computer interaction increased the participants' interest in taking future computer-related courses, raised awareness of the role that computers plays in their everyday lives, and influenced their perceptions of computing. This study didn't control for gender or ethnicity.

Two of the authors of the main text conducted an interactive prototyping workshop with 19 minority middle school girls and found that the ability to visualize computer applications via prototyping positively influenced the girls' perceptions of computer science and reduced stereotypes.[4]

References

1. S. Khoja et al., "Changing Girls' Attitudes towards Computer Science," *J. Computing Science in Colleges*, vol. 28, no. 1, 2012, pp. 210–216.
2. E.A. Plant et al., "Changing Middle-School Students' Attitudes and Performance regarding Engineering with Computer-Based Social Models," *Computers & Education*, vol. 53, no. 2, 2009, pp. 209–215.
3. S. Yardi et al., "An HCI Approach to Computing in the Real World," *ACM J. Educational Resources in Computing*, vol. 8, no. 3, 2008; http://dl.acm.org/citation.cfm?id=1404938.
4. A. Robinson and M.A. Pérez-Quiñones, "Underrepresented Middle School Girls: On the Path to Computer Science through Paper Prototyping," *Proc. 45th ACM Tech. Symp. Computer Science Education*, 2014, pp. 97–102.

Computing in Science & Engineering | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page

HCI	
Day 1	Introduction to computer science and related careers
Day 2	HCI and user interface evaluation introduction
Day 3	Paper prototyping social networking app
Day 4	Digital prototyping with iPads and group presentations
Day 5	Workshop wrap up and focus group interviews
Algorithms	
Day 1	Introduction to computer science and related careers
Day 2	Algorithm design and evaluation introduction
Day 3	Deeper look into algorithm design through iPad gaming
Day 4	Algorithm design through gaming and group presentations
Day 5	Workshop wrap up and focus group interviews

Figure 1. HCI and algorithm workshop summaries.

that the subscales are sufficiently stable enough to be used as separate scores. When administered to 37 sixth- through eighth-grade girls, Cronbach's alpha for the 10 items were .75, indicating reasonable internal consistency. Both this survey and the computational thinking survey were conducted on the first and last days of the workshop.

Computational Thinking Survey

Albert Bandura's social cognitive theory[4] is the overarching theoretical framework used in this study. Specifically, we used the theoretical component of self-efficacy as a lens through which to design the instrumentation and interpret results. Self-efficacy, the belief in one's capabilities to achieve desired outcomes, influences persistence as well as other psychological processes. Interest, confidence, and self-efficacy are closely related. A loss of confidence and self-efficacy often leads to a loss of interest.[5] Self-efficacy has an even deeper influence, often affecting performance, personal goals, expended effort, perseverance, resilience to failures, and career choice.[4]

The ratings for the computational thinking survey's 30 items use a 4-point scale ranging from "strongly agree" to "strongly disagree." We based the phrase structure of the survey items on Bandura's framework.[6] To ensure that capability was measured rather than intention, the items were phrased in terms of "can do" rather than "will do," as recommended by Bandura.[6] We used computational thinking practices[7] for the survey domain to minimize the technical details of computer science in the items and focus on computer science concepts to which middle school students can relate. Each computational thinking practice (for example, connecting computing, developing computational artifacts, abstracting, analyzing problems and artifacts, communicating, and working effectively in teams) was represented by five items on the survey to make it specific to the computational thinking domain. When administered to 37 sixth- through eighth-grade girls, Cronbach's alpha for the 30 items were .91, indicating good internal consistency.

Focus Group Interviews

We conducted focus group interviews with participants on the last day of the workshop. With the standardized open-ended interview approach, we determined in advance the interview questions. We also subsidized the standardized open-ended interview with an informal conversational interview approach to improve flexibility and contextual relevance. Participants voluntarily answered focus group questions; therefore, every participant didn't respond to each and every focus group interview question. We used content analysis to analyze the transcribed data obtained from 26 focus group interview participants. We calculated Krippendorff's alpha with a reliability of $\alpha = 0.7213$.

Results

We analyzed quantitative data through inferential statistics and qualitative data through content analysis.

Pre- and Postworkshop Survey Score Comparisons

We used paired sample *t*-tests to see if there was a significant difference between the pre- and postworkshop survey scores for the computer confidence (CC) and computational thinking (CT) instruments. The overall preworkshop CT survey scores were significantly less than ($p < 0.01$) the postworkshop survey scores. To further investigate the main effect, we ran separate paired sample *t*-tests for both the HCI and algorithms groups. The preworkshop CT survey scores for the HCI group were significantly less than ($p < 0.05$) the postworkshop survey scores. Both significant pairs represent a moderate practical significance ($d = .554$ and $d = .605$, respectively). There was no significant difference between the pre- and postworkshop CC survey scores.

Participant Characteristics and Survey Score Predictions

We performed stepwise multiple regression for four cases. Each of the preworkshop and gain survey scores were the dependent variables, and the four cases used the participant characteristics as the independent variables (see Table 1). We calculated the gain survey scores by subtracting the postworkshop survey scores from the preworkshop survey scores.

Computer confidence. In case 1, we conducted stepwise multiple regression to evaluate whether any participant characteristics could help predict preworkshop CC survey scores. The prediction model contained 2 of the 11 predictors and was reached in two steps with no variables removed. The model was statistically significant, $F(2, 28) = 7.666$, $p < .01$, and accounted for approximately 31 percent of the variance in preworkshop CC survey scores (adjusted $R^2 = .308$). Preworkshop CC survey scores were predicted by a lower socioeconomic status and higher school grades. Socioeconomic status accounted for approximately 27 percent of the variance in preworkshop CC survey scores beyond the variance accounted for by school grades, which accounted for approximately 26 percent of the variance in preworkshop CC survey scores beyond the variance accounted for by socioeconomic status.

In case 2, we conducted stepwise multiple regression to evaluate whether any participant characteristics could be used to predict gain CC survey scores. The prediction model contained 3 of the 11 predictors and was reached in three steps with no variables removed. The model was statistically significant, $F(3, 22) = 9.943$, $p < .001$, and accounted for approximately 52 percent of the variance in gain CC survey scores (adjusted $R^2 = .518$). Gain **//in?//** CC survey scores were predicted by a higher socioeconomic status, lower school grades, and the use of a smartphone at home. Socioeconomic status accounted for approximately 30 percent of the variance in gain CC survey scores beyond the variance accounted for by the other two predictors. School grades accounted for approximately 23 percent of the variance in gain CC survey scores beyond the variance accounted for by the other two predictors. Smartphone use at home accounted for approximately 17 percent of the variance in gain CC survey scores beyond the variance accounted for by the other two predictors.

Prior to the workshop, the participants' socioeconomic statuses and school grades influenced computer confidence scores. Participants from low socioeconomic backgrounds were predicted to have greater CC survey scores before the workshop. This finding is consistent with the participant statements in the focus group interviews. In the postworkshop focus group interviews, participant responses

Table 1. Stepwise multiple regression analysis variables.

Case	Dependent variable	Independent variables
1	Preworkshop computer confidence	Desktop, laptop, tablet, smartphone, videogame console, other device, home internet, father's education, mother's education, socioeconomic status, school grades
2	Gain computer confidence	
3	Preworkshop computational thinking	
4	Gain computational thinking	

about perceptions of computer science before the workshop were primarily divided along the lines of socioeconomic background. Participants from low socioeconomic backgrounds generally thought that computer science was boring or admitted to not knowing what it was, whereas several participants from medium/high socioeconomic backgrounds believed that computer science was hard before the workshop. Participants with higher grades were predicted to have greater computer confidence before the workshop. Good grades are evidence of performance accomplishments. Bandura[8] states that past successes or performance accomplishments positively influences how people feel about themselves and approach situations.

Computational thinking. In case 3, we conducted stepwise multiple regression to evaluate whether any participant characteristics could be used to predict preworkshop CT survey scores. There was no discernible linear relationship between preworkshop CT survey scores and any of the participant characteristics; $F(11, 19) = 1.153$, $p = .378$, accounting for approximately 5 percent of the variance of preworkshop CT survey scores (adjusted $R^2 = .053$).

In case 4, we conducted stepwise multiple regression to evaluate whether any participant characteristics could help predict gain CT survey scores. The prediction model contained 2 of the 11 predictors and was reached in two steps with no variables removed. The model was statistically significant, $F(2, 23) = 6.717$, $p < .01$, and accounted for approximately 31 percent of the variance in gain CT survey scores (adjusted $R^2 = .314$). Gain CT survey scores were primarily predicted by a higher mother's education and the use of a videogame console at home. Mother's education accounted for approximately 11 percent of the variance in gain CT survey scores, beyond the variance accounted for by videogame console use at home, which accounted for

Computing in SCIENCE & ENGINEERING | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page | Qmags THE WORLD'S NEWSSTAND®

Most of the participants had positive perceptions of computer science after the workshop, describing the workshop as "fun," "cool," "exciting," "easy," "important," and "not boring."

approximately 21 percent of the variance in gain CT survey scores beyond the variance accounted for by mother's education.

The workshop participants' mother's education and whether participants used a videogame console influenced the gains in CT survey scores after the workshop. Participants whose mother had a higher level of education were predicted to have greater gains in CT survey scores after the workshop, demonstrating the significant role that parents, especially mothers, play in the lives of adolescents. Participants who used a videogame console at home were predicted to have greater gains in CT survey scores after the workshop. Literature states that African-Americans spend more time playing videogames than nonminorities.[9] The data collected from the focus group interviews reflect the influence of videogame consoles with statements such as, "I can make a dance game" when asked if she wanted a computer science career and why.

Emerging Themes

Through content analysis, six themes emerged that provide insight into the attitudes of primarily African-American middle school girls and how the workshop influenced these attitudes. The themes are perceptions of computer science, collaboration, workshop activities, computer science and the real world, empowerment, and future plans in computer science.

Perceptions of computer science. Eighteen participants expressed the perceptions they had about computer science before the workshop, reporting either a negative perception of it or not really knowing what it was. Differences in the types of opinions participants had tended to be based on socioeconomic backgrounds. Participants from low socioeconomic backgrounds generally considered computer science to be boring before the workshop, with one participant stating that it was for "geek people." Participants from higher socioeconomic backgrounds also believed that computer science was boring, but they made additional statements, such as computer science being "nerdy" or "for nerds," "hard," and "pretty stupid." One participant said, "At first, I thought computer science was going to be really hard and you have to be super smart and stuff for it."

Before the workshop, seven participants misunderstood or didn't know what computer science was. Four participants from low and medium/high socioeconomic backgrounds associated computer science with doing the traditional sciences, such as life sciences, on the computer, and also associated it with math. One participant stated, "I thought it was going to be a bunch of math and junk, and like you gotta type it into a computer and make some stuff." Of the participants who misunderstood what computer science was about, two stated that computer science was "boring" and "pretty stupid." A few participants from low socioeconomic backgrounds admitted to not knowing anything about computer science.

Most of the participants had positive perceptions of computer science after the workshop, describing the workshop as "fun," "cool," "exciting," "easy," "important," and "not boring." Three of the seven participants who didn't know about computer science or misunderstood it indicated that they knew what it was about postworkshop. One participant stated, "I think of it as doing programs and stuff and getting to know the computer."

Three participants had a negative perception of computer science after the workshop, with two saying that it can "sometimes be hard." One participant from the algorithms group thought that computer science was too rigid, stating, "I just don't want to hear people telling me, 'you have to do this, you have to do that,' in order to be right. I don't want nobody telling me that." However, all participants who had negative perceptions of computer science after the workshop indicated that they also had positive perceptions. Those who said that computer science can sometimes be hard also made comments such as, "Some stuff is easy." While participants admitted that they had fun, they feared that their friends who didn't attend the workshop would think it was "nerdy and boring." These participants were also interested in learning more about computer science or were interested in a computer science career.

Eight participants had negative opinions of the workshop activities, with most saying they didn't like the research aspect (completing surveys and writing for data collection) or the activities that weren't very interactive (the introduction, looking at the screen).

Collaboration. Twenty participants spoke of their experiences working in a group with their peers. Most of them enjoyed it, saying that it was "helpful," "fun," "great," "awesome," "encouraging," and "a good experience." Those who thought it was helpful referred to the work getting done quicker with statements such as, "You can get more work through faster and quicker, and it won't be no rush." Others who thought it was encouraging explained how they supported one another: "We all listened to each other and supported each other and helped each other if we had bad answers," and "When I didn't know the answers, I could always like ask someone for the answers." While one participant had fun working in a group, she stated, "At the same time, if they don't cooperate, we not [gonna] have the team effort."

Four participants indicated a lack of team effort or displeasure with team members. Social dynamics were the reasons for the negative collaboration experiences in the HCI group, with one participant saying, "My partner, she tried to hold my iPad ... and she didn't think I could use it so she had hauled the iPad from me, but I had got it back," while the partner stated, "Most people got annoying." The lack of ability for teammates to work together appears to be the primary reason for the negative collaboration experiences in the algorithms group. One participant stated, "My partner didn't help me none at all," while another participant stated, "People didn't catch up quick enough."

Workshop activities. Twenty participants stated their opinions about workshop activities, with 17 rating them positively. They enjoyed looking at the videos, using the iPad, playing the Cargo-Bot game (the assigned task for the algorithms group), taking pictures (a component of the activity for the HCI group), answering questions, and viewing a PowerPoint slideshow that was used to introduce the lessons. Ten of the participants with positive opinions of workshop activities revealed feeling a sense of accomplishment. Participants spoke of how they "made their own apps," "made a social network," "put programs together," "controlled the robots," "handled computers," and "told the computer what to do."

Eight participants had negative opinions of the workshop activities, with most saying they didn't like the research aspect (completing surveys and writing for data collection) or the activities that weren't very interactive (the introduction, looking at the screen). Others didn't like some activities, such as "writing because I don't know how to draw" or the algorithm design activity on day two of the algorithms workshop. Both activities involved writing. Five participants who had negative opinions also had positive opinions, with two indicating a sense of accomplishment.

Computer science and the real world. Eight participants indicated that they were able to connect computer science to the real world. After the workshop, participants thought that computer science was useful, as revealed in the following statements:

- "It helps scientists, and it helps firefighters, and it helps police officers, and other people in the world."
- "If you have a job, and it's really important, and you gotta give a presentation, you can do visual aid; like you can touch it and make sure that it's professional and stuff."
- "It is a way to chat and communicate with other people."
- "It can get us farther places in life, and we can go to better colleges and stuff."

They also made references to robots helping people with statements such as, "It [computer science] helps us go back into the past in some ways and it can always help with things in the future like robots pads and stuff, and mechanical houses and stuff, and robots that can do everything for you that might take over the world."

Empowerment. Eight participants revealed a sense of empowerment after the workshop. Most believed they could create something after the workshop, as indicated by the following statements:

Our research findings indicate that the intervention positively influenced the attitudes of African-American middle school girls toward computer science along with several other factors.

- "You can create a lot of things for other people to use like for school, and you can create apps and stuff."
- "I can make a dance game: Michael Jackson—owww!"
- "I can download this on my iPad at home and make my own apps, and maybe one day will come true."
- "I think one day I might make another Facetime—Facetime 2."
- "I want to make something like all the kids, all like the preteen kids and all, just like Facebook, but it's more like you can endure more like you can, like everybody experience[s] it."

Some participants spoke about the ability to do computer science, stating that they "learned that anybody can do computer science" and "you can actually want to be a computer science technologist and try to make some more apps and more things." Other participants spoke about being able to troubleshoot computer problems on their own, with statements such as, "If you need help with computers you just can do it on your own," and "Now when I go home, if I ever have trouble on the computer, I have more of a feeling of what to do." Other participants referenced learning, saying, "If you want to learn about computer science, you can do it now," and "If they have a class about computers at my school, I am definitely going to sign up for it. ... If they don't have a class, I am going to make them give us a class."

Future plans in computer science. Seventeen participants expressed their willingness to learn more about computer science or have a computer science career. Of those 17 participants, 10 indicated that they were interested in learning more about computer science and 6 participants indicated they might be interested in learning more about computer science. When asked if they wanted a computer science career, four participants said yes, three participants said they were considering a career in computer science, and two participants said they didn't want a career in computer science. Both of the participants who didn't want a computer science career said they still wanted to learn more about computer science.

Discussion

Our research findings indicate that the intervention positively influenced the attitudes of African-American middle school girls toward computer science along with several other factors, including the intervention content domain, the facilitation of performance accomplishments, and participant characteristics such as socioeconomic status, mother's education, school grades, and the use of smartphones and videogame consoles at home.

Participating in an intervention. In this study, we found that participation in an intervention (the workshop) positively influenced the participants' attitudes toward computer science and reduced the negative stereotypes, also found in the literature,[10] they had prior to participating. Exposure to computer science through the workshop balanced perceptions about computer science between the participants from different socioeconomic backgrounds, with most participants indicating that they had positive perceptions afterward. All the participants who had negative perceptions of computer science after the workshop indicated that they also had positive perceptions, were interested in learning more about computer science, or were interested in a computer science career. Participants were also able to make a connection between computer science and the real world, which empowered and inspired them to do independent computer science work.

Intervention content domain. The intervention contained two different content domains, HCI and algorithms, which tended to differ in terms of their influence on the participants' CT survey scores and in the challenges to collaboration that participants experienced. The HCI user interface design and evaluation domain had a positive significant influence on participants' CT survey scores, while the algorithms domain did not significantly influence those scores. Although participants in both domains had challenges in collaboration, the challenges in the algorithms content domain were directly related to the workshop activity. On the other hand, the challenges in the HCI content domain appeared to be primarily related to social dynamics. The differences in the survey results

Participants' existing knowledge and interests. Participants were already users of mobile devices, knew how to draw, were familiar with recording homework assignments, and were interested in social networking. The interface design and evaluation activities taught them a new way to combine and utilize their existing skillsets.

and collaboration experiences of participants in the HCI and algorithms content domains could be attributed to the domains' abilities to facilitate the use of existing skillsets. In the HCI group, participants could utilize preexisting skills, such as drawing, and their familiarity with technology. Although the algorithms group was also familiar with the technology, applying algorithmic thinking to a computer device is a skill that many had not been introduced to prior to the workshop.

Facilitation of performance accomplishments. According to Bandura, successfully completing assigned tasks can be the most influential source of self-efficacy information. Within the context of this research, the tasks were creating a prototype of a social networking application and solving problems in the Cargo-Bot game. The main task for the HCI group was to create a low-fidelity prototype of a social networking app using the Prototyping on Paper (POP) app (https://popapp.in). The algorithms group played the Cargo-Bot game (http://twolivesleft.com/CargoBot), where they had to duplicate a goal pattern by providing the robot with the appropriate instructions for moving colored boxes on a platform.

Several factors can facilitate or hinder performance accomplishments, including the adequacy of the resources available and assistance provided by others. In this research, performance accomplishments were facilitated by technology resources provided to do the assignments (iPads) and by assistance from others (by working in groups).

Most of the participants mentioned how much they enjoyed using the iPads, with some indicating that it was their favorite part of the workshop. One participant stated, "I didn't even like [the workshop] until we got on the iPads," while another participant stated, "[The workshop] started getting fun 'cause we started working with the iPads." The iPads made the workshop "fun," "nice," and "exciting," according to participants.

The technology enabled participants to connect design to computer science. This was revealed when participants referred to the activity as "putting programs together" and to their user interfaces as social networking "apps" rather than just drawings on sheets of paper or a collection of pictures. In the algorithms group, participants not only enjoyed using the iPads, they also believed that the iPad technology could be used as a motivator to get their peers interested in the workshop.

The technology helped participants visualize the robot receiving and following instructions. Participants were able to understand concepts of the game, such as "how you pick up the block and move it to the next" and "how we really had to be specific on how we did the game." They were also able to translate the concept of controlling the robot to "handling computers" and "telling the computer what to do." The use of the iPads enhanced the participants' abilities to understand concepts and accomplish the assigned tasks in both groups.

We instructed participants to work in small groups, where they could receive assistance from peers. Most of the participants enjoyed working in groups, describing the experience as "fun," "good," "great," "awesome," and "encouraging." Participants also thought collaborating with their peers was very helpful, realizing that it assisted in their performance accomplishments. One participant indicated, "When I didn't know the answers, I could always like ask someone for the answers." Group members "listened to each other," "supported each other," "shared ideas," "helped each other if we had bad answers," and made "something with all of our ideas." This enabled participants to "get stuff done quicker," get "a lot done," and do "a good job." However, collaborating with peers didn't always facilitate performance accomplishments. There were instances when "people got annoying," "partners didn't listen to ideas," and when partners didn't provide assistance when needed. In these instances, the lack of assistance from peers might have hindered performance accomplishments.

Participant characteristics. Some participant characteristics, including socioeconomic status, mother's education, school grades, and the use of smartphones and

Computing in SCIENCE & ENGINEERING | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page | Qmags THE WORLD'S NEWSSTAND®

In order to attract this demographic to computer science, the exact type of device used doesn't seem to be so important; what does matter is that the device is one that participants often interact with and enjoy using.

videogame consoles at home, influenced participants' attitudes toward computer science through their CC and CT survey scores. Additionally, socioeconomic status influenced the participants' perceptions of computer science before the intervention, as indicated by the focus group interview analysis results. The perceptions of those from low socioeconomic backgrounds before the intervention were either nonexistent or negative; however, they weren't as critical as those from medium/high socioeconomic backgrounds. Those from low socioeconomic backgrounds might have also had a more positive approach to the intervention.[11]

Our research study revealed that introducing African-American middle school girls to computer science through user interface design and evaluation produces good outcomes. The user interface design and evaluation activities built on participants' existing knowledge and interests. Participants were already users of mobile devices, knew how to draw, were familiar with recording homework assignments, and were interested in social networking. The interface design and evaluation activities taught them a new way to combine and utilize their existing skillsets. With a foundation based on preexisting skills, performance accomplishments were obtainable for participants. These performance accomplishments are a source of self-efficacy that can, in turn, lead to increased interest in computer science.

The home life of African-American middle school girls is influential in their attitudes toward computer science. The factors of socioeconomic background and mother's education influencing attitudes are consistent with existing literature.[12,13] These factors can be equated to degrees of exposure to different experiences. Although we would expect those with more experiences to initially have more positive attitudes toward computer science, that isn't necessarily the case with this demographic, which is inconsistent with literature that states that those from high socioeconomic backgrounds are more likely to choose male-dominated occupations and have high self-efficacy expectations.[13]

At home, computer science might not necessarily be encouraged in the homes of African-American middle school girls from privileged backgrounds. However, this group does benefit the most from computer-related interventions. Perhaps this is because they began with more negative attitudes initially when compared to their counterparts from underprivileged backgrounds. School grades also were a factor and consistent with current literature.[14]

The use of videogames and smartphones at home positively influences African-American middle school girls' attitudes toward computer science. Smartphone use appearing as a factor is consistent with the literature stating that women, regardless of race, are the most intense cell phone users.[15] Because the literature states that males are the most intense videogame players regardless of race,[15] it was a bit surprising that videogames were also revealed as an influencing factor for the African-American middle school female study participants. This could indicate that African-American females play videogames more than their Caucasian female counterparts. Additionally, playing videogames was found to be a negative predictor of academic performance in the literature.[15] However, playing videogames were beneficial to the self-reported beliefs of the participants in this study.

Videogames and smartphones were key to attracting African-Americans girls to computer science in our study. In order to attract this demographic to computer science, the exact type of device used doesn't seem to be so important; what does matter is that the device is one that participants often interact with and enjoy using. Using devices in our intervention aided in the scaffolding process where new skills were learned within the context of existing skillsets. ■

Acknowledgments

We thank the Boys & Girls Clubs and the YMCAs in the Southeast Hampton Roads area of Virginia for allowing us to do the workshops in their facilities. We also thank Tonya Smith-Jackson, Andrea Kavanaugh, and

Scott McCrickard for their thoughtful comments on the preparation of the materials for the workshop. This article is an elaboration of work previously published in the 2014 Special Interest Group in Computer Science Education (SIGCSE) and the 2015 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT) Conference Proceedings.

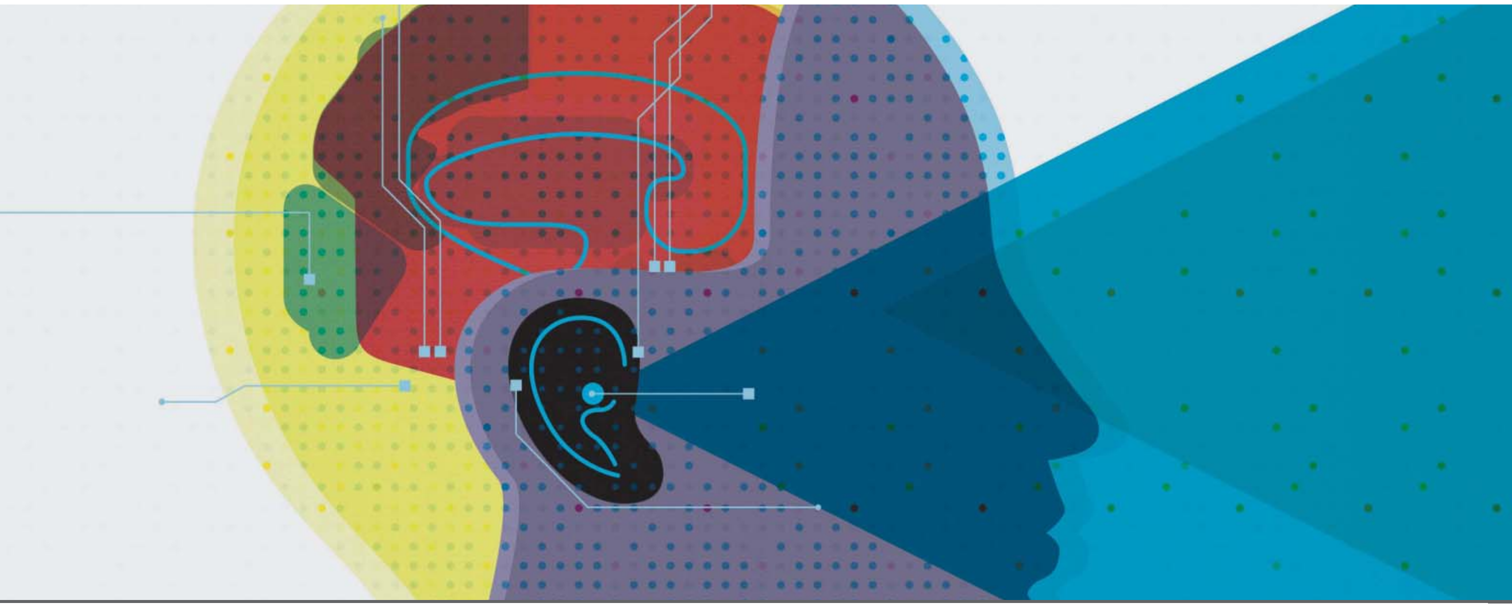
References

1. *Women, Minorities and Persons with Disabilities in Science and Engineering*, tech. report, US Nat'l Science Foundation, 2015; www.nsf.gov/statistics/wmpd/.
2. S. Yardi et al., "An HCI Approach to Computing in the Real World," *ACM J. Educational Resources in Computing*, vol. 8, no. 3, 2008; http://dl.acm.org/citation.cfm?id=1404938.
3. B.H. Loyd and D.E. Loyd, "The Reliability and Validity of an Instrument for the Assessment of Computer Attitudes," *Educational and Psychological Measurement*, vol. 45, no. 4, 1985, pp. 903–908.
4. A. Bandura, "Self-Efficacy," *Encyclopedia of Human Behavior*, V.S. Ramachaudran, ed., Academic Press, 1994, pp. 71–81.
5. J. Margolis and A. Fisher, *Unlocking the Clubhouse: Women in Computing*, MIT Press, 2002.
6. A. Bandura, "Guide for Constructing Self-Efficacy Scales," *Self-Efficacy Beliefs of Adolescents*, vol. 5, F. Pajares and T. Urdan, eds., Information Age Publishing, 2006, pp. 307–337.
7. The College Board, "Computer Science: Principles Computational Thinking Practices Big Ideas, Key Concepts, and Supporting Concepts," 2011; www.collegeboard.com/prod_downloads/computerscience/ComputationalThinkingCS_Principles.pdf.
8. A. Bandura, *Self-Efficacy: The Exercise of Control*, W.H. Freeman and Company, 1997.
9. S. Yardi and A. Bruckman, "Income, Race, and Class: Exploring Socioeconomic Differences in Family Technology Use," *Proc. SIGCHI Conf. Human Factors in Computing Systems*, 2012, pp. 3041–3050.
10. S. Yardi and A. Bruckman, "What Is Computing? Bridging the Gap between Teenagers' Perceptions and Graduate Students' Experiences," *Proc. 3rd Int'l Workshop Computing Education Research*, 2007, pp. 39–49.
11. R.H. Bradley et al., "Measuring the Home Environments of Children in Early Adolescence," *J. Research on Adolescence*, vol. 10, no. 3, 2000, pp. 247–288.
12. J.L. Kerpelman, M.F. Shoffner, and S. Ross-Griffin, "African American Mothers' and Daughters' Beliefs about Possible Selves and Their Strategies for Reaching the Adolescents' Future Academic and Career Goals," *J. Youth and Adolescence*, vol. 31, no. 4, 2002, pp. 289–302.
13. J.S. Hannah and S.E. Kahn, "The Relationship of Socioeconomic Status and Gender to the Occupational Choices of Grade 12 Students," *J. Vocational Behavior*, vol. 34, no. 2, 1989, pp. 161–178.
14. C.T. Amelink and P.S. Meszaros, "A Comparison of Educational Factors Promoting or Discouraging the Intent to Remain in Engineering by Gender," *European J. Eng. Education*, vol. 36, no. 1, 2011, pp. 47–62.
15. L.A. Jackson et al., "Race, Gender, and Information Technology Use: The New Digital Divide," *Cyberpsychology & Behavior: The Impact of the Internet, Multimedia and Virtual Reality on Behavior and Society*, vol. 11, no. 4, 2008, pp. 437–442.

Ashley Robinson is currently an industry user experience researcher. Her research interests include computer science and engineering education, and computer-supported collaborative learning. Robinson earned a PhD in Computer Science and Applications, with a concentration in HCI, from Virginia Tech. She has earned several merit-based awards, including the National Science Foundation Graduate Fellowship and the Google United Negro College Fund Scholarship. Contact her at arrobin@vt.edu.

Manuel A. Pérez-Quiñones is a Professor of Software and Information Systems and Associate Dean of the College of Computing and Informatics at the University of North Carolina at Charlotte. His research interests include personal information management, HCI, user interface software, and educational/cultural/diversity issues in computing. Pérez-Quiñones earned a DSc from George Washington University. He's an NSF CAREER Awardee. Contact him at Perez.Quinones@uncc.edu.

Glenda Scales serves as both Associate Dean for Global Engagement and CTO for the College of Engineering at Virginia Tech. She also serves as the director of integrated learning services for the college. Her research interest is in engineering education. Scales received a PhD in curriculum and instruction with a concentration in instructional technology from Virginia Tech. Contact her at gscales@vt.edu.



Lesbian, Gay, Bisexual, Transgender, and Queer Students' Sense of Belonging in Computing: An Intersectional Approach

Jane G. Stout and Heather M. Wright
Computing Research Association

The field of computing is rapidly developing, requiring a strong and diverse labor force. However, the results of two studies indicate that LGBTQ undergraduate and graduate students think about leaving computing degree programs due to a low sense of belonging in the computing community.

Queen Elizabeth II recently pardoned Alan Turing after he had been convicted of "gross indecency" with a man in 1952. A key figure in the development of computer science, Turing died in 1954 at the age of 41; his cause of death is widely believed to have been suicide.[1] This regrettable historic event illustrates untapped potential in the field of computing from an individual in the lesbian, gay, bisexual, transgender, and queer (LGBTQ) community. Six decades later, most cultures no longer publicly persecute individuals due to sexual orientation, as was the case with Turing, but bias against LGBTQ individuals persists. Biases can be subtle (perceptible glances or usage of dysphemisms such as "that's so gay") or explicit (heterosexist/homophobic harassment, institutional discrimination, or hate crimes).[2] In either case, bias is likely to make LGBTQ individuals feel as though they do not "belong" in the social milieu, which can have damaging effects on the self-concept, motivation, and achievement.

The need to belong is widely theorized to be a fundamental necessity for psychological and physical well-being.[3,4] When individuals do not feel a secure sense of belonging in academic settings, which are the

 1521-9615/16/$33.00 © 2016 IEEE Copublished by the IEEE CS and the AIP

We expected that LGBTQ undergraduate and graduate students would show a stronger relationship between thoughts about leaving computing and a low sense of belonging in the computing community than their heterosexual peers.

milieus of interest in our current work, individuals' motivation, achievement, and persistence tend to suffer.[5,6] Moreover, LGBTQ students report more instances of harassment than heterosexual students on college campuses,[7] even though LGBTQ students tend to choose to attend institutions that have reputations for being supportive of the LGBTQ community.[8] In addition to overall campus climate, some specific academic settings can unintentionally foster a particularly low sense of belonging among LGBTQ students. For instance, computing departments tend to foster a heteronormative social environment, in which heterosexuality is assumed to be the norm, such that institutions and policies are aligned with a heterosexual lifestyle.[9] Consistent with this, an interview study revealed that LGBTQ faculty members in computing departments perceived that heterosexuality was the assumed norm among colleagues.[10] Thus, existing theory and interview data among individuals in computing suggest that LGBTQ students pursuing postsecondary computing degrees have reason to feel a generally lower sense of belonging in their degree tracks than heterosexual students.

Within computing, women are also considered a marginalized group and are known to feel a lower sense of belonging therein than men.[11,12] Thus, our expectation was that women in the LGBTQ community would feel a particularly low sense of belonging compared to their peers, given that LGBTQ women belong to two social groups that are underrepresented and even stigmatized in computing. Such a hypothesis is consistent with intersectionality theory, indicating individuals' subjective experiences are subject to multiple (that is, intersecting) social identities.[13]

Importantly, extant theory and research suggest a thwarted sense of belonging among LGBTQ students in computing should increase their inclination to leave a computing career track.[3,4] Systematic attrition from computing among LGBTQ students is suboptimal for several reasons. For one, a dearth of LGBTQ individuals in computing means that this group of individuals' needs and interests become underrepresented in computing innovations. Moreover, this group of individuals brings a diverse perspective to the computing enterprise, which is associated with high innovation and productivity.[14,15] Finally, high dropout rates among whole subgroups of students will result in a low volume of qualified workers, impeding society's ability to build a strong computing labor force.

Assessing Belonging in Computing

Despite the implications of understanding LGBTQ students' sense of belonging in computing settings, to our knowledge, this topic has not yet been formally studied. In this article, we assess LGBTQ versus heterosexual students' sense of belonging in computing and its relation to students' intentions to persist in a computing career track. We also take an intersectional approach in our work by looking at whether and how students' gender interacts with their LGBTQ identity to influence their sense of belonging in computing. We collected data from undergraduate and graduate students, which allowed us to assess whether our findings replicate across two samples and generalize across individuals at different stages of their computing career preparation. We expected that LGBTQ undergraduate and graduate students would show a stronger relationship between thoughts about leaving computing and a low sense of belonging in the computing community than their heterosexual peers.

Study 1

We invited undergraduate computing students to complete an online survey sent to a national sample of colleges and universities during the fall academic semester of 2013.

Method

We recruited 857 undergraduate students majoring in a computing field from a sample of computing departments across the US to complete an online survey in exchange for being entered in a raffle to win a US$100 gift card. We define "computing field" as computer science, computing engineering or electrical and computer engineering, computing information systems, or another computing-related field including interdisciplinary fields with a strong computing component (such as computational biology or digital media). At the end of the survey, we asked students to provide demographic information such as sexual orientation, race, and gender. Eighty-six students self-identified as LGBTQ ($n = 9$ lesbian; $n = 28$ gay; $n = 45$ bisexual; $n = 1$ transgender; $n = 3$ queer), and 771 students self-identified as heterosexual.

Of the students in our sample, 63 percent attended institutions where the highest computing degree offered in their department was a PhD, 15 percent where the highest degree available was a terminal MS, and 20 percent where the highest degree available was a BS; 2 percent of students did not provide institution information. Twenty-five percent of our sample was women and 75 percent was men. The racial and ethnic makeup of the sample was 4 percent African-American, 12 percent Asian-American, 61 percent Caucasian, 9 percent Latina/Latino, 12 percent more than one race, and 2 percent other.

Students completed an online survey that included questions pertaining to students' thoughts about leaving their academic program and reasons for doing so. The following item assessed students' thoughts about leaving their major: "Since declaring or planning to declare your computing major, have you seriously considered changing to a noncomputing major?" Response options were yes or no.

Students who had thought about leaving their major were asked the following two follow-up questions: "How much do you disagree or agree with the following statements: I have considered changing to another major because "… I do not feel welcomed in the computing community"; "I do not feel like I 'fit' in the computing community," using a scale ranging from 1 (strongly disagree) to 5 (strongly agree). Items had good internal reliability ($\alpha = .89$),[16,17] so we created an average score of the two items and used this composite variable to assess students' endorsement of low belonging as an explanation for why they've considered leaving their program.

Finally, we asked students to report their current GPAs for their computing majors using a 4.0 scale, which served as a covariate in the following analyses.

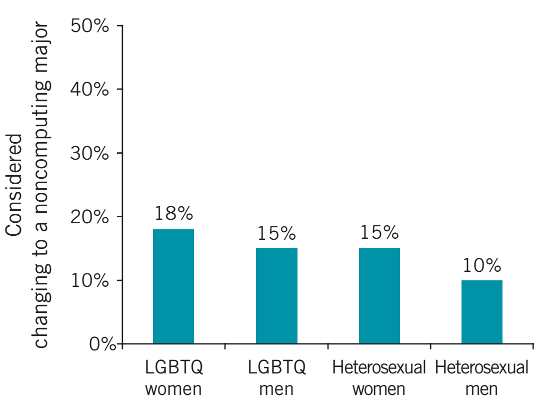


Figure 1. Percent of undergraduate students who thought about changing to a noncomputing major: LGBTQ × gender.

Results

Regarding students' thoughts about leaving, we first assessed whether LGBTQ students were more likely to think about leaving their major than heterosexual students and whether thoughts about leaving were particularly high among LGBTQ women. To do this, we ran a multiple logistic regression in which we regressed whether students had thought about changing to a noncomputing major ($0 =$ no; $1 =$ yes) on LGBTQ ($-1 =$ heterosexual students; $1 =$ LGBTQ students), gender ($-1 =$ women; $1 =$ men), and their interaction term (LGBTQ × gender). We found that LGBTQ group identification, gender, and students' interaction terms were not significant predictors of students' thoughts about changing their major, $ps > .23$ (see Figure 1).

Although LGBTQ students did not consider leaving their program to a stronger degree than heterosexual students, we expected that among students who did consider leaving ($n = 101$), LGBTQ students would be more likely than heterosexual students to indicate that their reason for doing so was due to a low sense of belonging in the computing community. Moreover, we expected that female LGBTQ students would be particularly likely to report having thought about leaving due to a low sense of belonging.

To test these hypotheses, we ran a LGBTQ × gender analysis of variance (ANOVA) on the degree to which students' thoughts about leaving were related to a low sense of belonging in computing. We found that, indeed, LGBTQ students were more likely to have thought about leaving their major due to a low sense of belonging in computing than heterosexual students, $F(1, 97) = 6.85$, $p < .05$. Furthermore, a main effect of gender indicated that women who had considered leaving their major were more likely to do so than men due to a low sense of belonging, $F(1, 97) = 13.60$, $p < .001$. We did not find a significant LGBTQ × gender interaction, $F(1, 97) = 2.89$, $p = .093$, but we suspected that this lack of an interaction was due to few LGBTQ women and men in this particular analysis ($n = 7$ women; $n = 7$ men), resulting in low statistical power to detect this specific effect. Although this interaction effect was not significant, we opted to test our a priori hypothesis that LGBTQ women would be more likely than their peers to think about leaving their computing major due to low belonging in computing by way of a series of post hoc Dunnett t-tests, where we compared LGBTQ women's responses to those of the remaining three student groups. In doing so, we found that, indeed, LGBTQ women were significantly more likely than their peers to report thinking about leaving their major due to a low sense of belonging in computing, $ps < .05$ (see Figure 2).

Because students' undergraduate GPA tends to be positively related to sense of belonging in achievement settings,[5] we reran our analysis, this time statistically controlling for students' reported major GPAs. Doing so did not change our results.

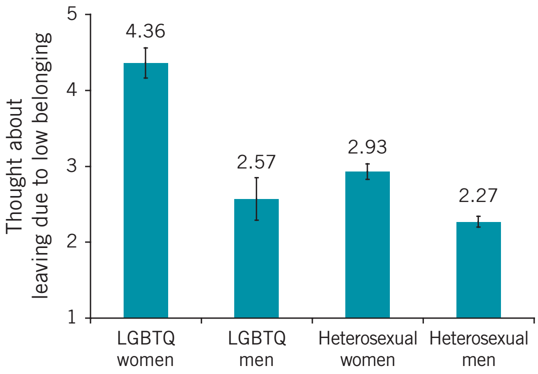


Figure 2. Undergraduate students' indication that thoughts about changing to a noncomputing major were due to a low sense of belonging in computing: LGBTQ × gender. Bars indicate mean responses for each group. Group means are displayed above each bar, and standard errors are displayed at the top of each bar.

Discussion

Study 1 provided empirical support for our hypothesis that LGBTQ students' thoughts about leaving are more strongly related to a lower sense of "fit" in computing compared to heterosexual students. Furthermore, this explanation for wanting to leave their major was most prevalent among female LGBTQ students who belong to two minority groups within computing. In a second study, we sought to conceptually replicate these findings among graduate students in computing using a slightly modified analytic design. An added benefit of focusing on graduate students in study 2 was that we could observe belonging and persistence intentions further into the computing career pipeline and note whether and to what degree LGBTQ graduate students might "leak" out of the academic pipeline relative to heterosexual students.

Study 2

We invited graduate students in computing programs to complete an online survey sent to a national sample of universities during the fall academic semester of 2013.

Method

We asked 45 LGBTQ ($n = 5$ lesbian; $n = 12$ gay; $n = 25$ bisexual; $n = 3$ queer) and 899 heterosexual graduate students pursuing either an MS or a PhD in a computing field from a sample of computing departments across the US to complete an online survey in exchange for being entered in a raffle to win a $100 gift card. The total number of students in this sample was 944.

Of the students in our sample, 85 percent attended institutions where the highest computing degree offered in their department was a PhD and 8 percent where a terminal MS was the highest degree offered; 7 percent of students did not report institution information. Fifty-eight percent of our sample was in terminal MS programs and 42 percent was in PhD programs. Thirty-eight percent of the sample was women, 61 percent was men, and 1 percent did not specify gender. The racial and ethnic makeup of the sample was 3 percent African-American, 48 percent Asian-American, 38 percent Caucasian, 3 percent Latina/Latino, 5 percent more than one race, and 3 percent other.

Students completed a survey containing questions pertaining to students' sense of belonging in the computing community and thoughts about leaving their graduate degree programs. The following question assessed students' thoughts about leaving their degree programs: "During your academic career, have you ever seriously considered leaving your graduate program?" Response options were "I have never seriously considered leaving" and "I have seriously considered leaving." Four questions assessed students' sense of belonging in computing: "I feel like I belong in computing," "I feel like an outsider in the computing community"

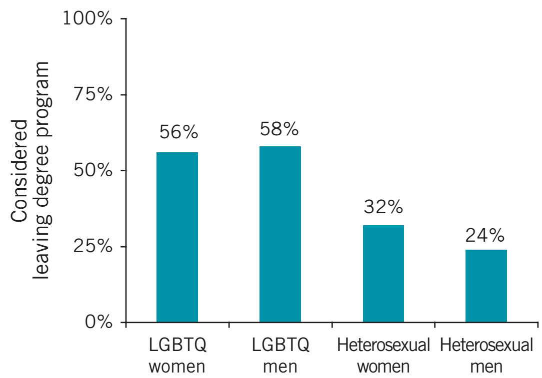


Figure 3. Percent of graduate students who thought about leaving their degree program: LGBTQ × gender.

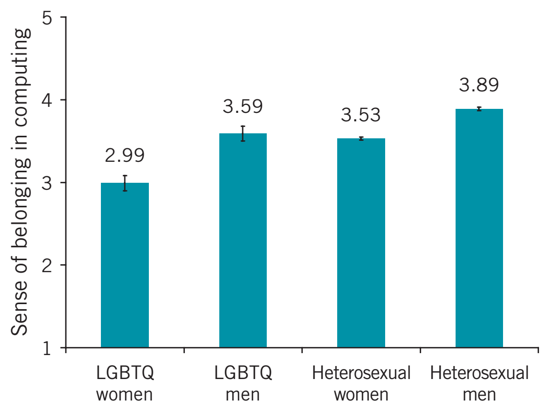


Figure 4. Graduate students' sense of belonging in computing: LGBTQ × gender. Bars indicate mean responses for each group. Group means are displayed above each bar, and standard errors are displayed at the top of each bar.

(reverse scored), "I feel welcomed in the computing community," and "I do not have much in common with the other students in my computing classes" (reverse scored), each using a scale ranging from 1 (strongly disagree) to 5 (strongly agree). These four items have good internal reliability ($\alpha = .73$), so we aggregated them and used their average as an index of belonging in the computing community.

Results

We first assessed whether LGBTQ students were more likely to think about leaving their graduate program than heterosexual students. Then we assessed whether thoughts about leaving were particularly high among LGBTQ women. We regressed whether students had thought leaving their degree program (0 = no; 1 = yes) on LGBTQ group (–1 = heterosexual students; 1 = LGBTQ students), gender (–1 = women; 1 = men), and their interaction term (LGBTQ × gender) via multiple logistic regression. We found that LGBTQ students were significantly more likely to have thought about leaving their degree program than heterosexual students, $B = .62$, $SE = .16$, χ^2 (1,944) = 14.65, $p < .001$. However, neither gender nor the LGBTQ × gender interaction were significant predictors of thoughts about leaving a degree program, *ps* > .41 (see Figure 3).

Because terminal MS programs versus PhD programs have important experiential differences (for example, given their shorter duration, MS programs offer less time to think about leaving), we reran our analysis, this time controlling for the type of degree program in which students were enrolled. The pattern of results did not change.

We next ran an LGBTQ × gender ANOVA on students' sense of belonging in computing and found that LGBTQ students reported a lower sense of belonging than heterosexual students, F (1, 935) = 11.16, $p < .01$. We also found that women reported a lower sense of belonging than men, F (1, 935) = 14.56, $p < .001$. Although there was not a significant LGBTQ × gender interaction, $p = .31$, we ran follow up Dunnett t-tests comparing LGBTQ women's sense of belonging to that of each of the three remaining student groups. Consistent with our a priori prediction, LGBTQ women reported a lower sense of belonging than LGBTQ men, $p < .05$, heterosexual women, $p < .05$, and heterosexual men, $p < .05$ (see Figure 4).

We next explored a possible explanation for LGBTQ students' greater tendency to think about leaving their degree program than heterosexual students: low belonging. Extant research and theory have found that feeling a secure sense of belonging in academic settings is associated with persistence therein.[5] We found that this was also the case in our data: students who felt a stronger sense of belonging were less likely to think about leaving their degree program, $B = -.59$, $SE = .09$, $\chi^2(1,944) = 41.49$, $p < .001$.

Having established that LGBTQ graduate students felt a lower sense of belonging in computing than their heterosexual peers and that feeling a secure sense of belonging in computing is associated with a lower tendency to think about leaving a degree program, we then tested whether LGBTQ students' lower sense of belonging might partially explain their comparatively greater overall tendency to think about leaving their degree programs than heterosexual students. Indeed, the original group disparity in thoughts about leaving a program ($B = 1.22$, $SE = .31$, χ^2 (1,944) = 15.61, $p < .001$) decreased in size after statistically controlling for students' sense of belonging ($B = 1.04$, $SE = .32$, χ^2 (1,944) = 10.68, $p < .01$). This indirect effect was significant, Sobel $Z = 3.09$, $SE = .08$, $p < .01$,[18] indicating that LGBTQ students' greater tendency to think about leaving was partially statistically explained by their lower sense of belonging in the computing community than that of heterosexual students (see Figure 5).

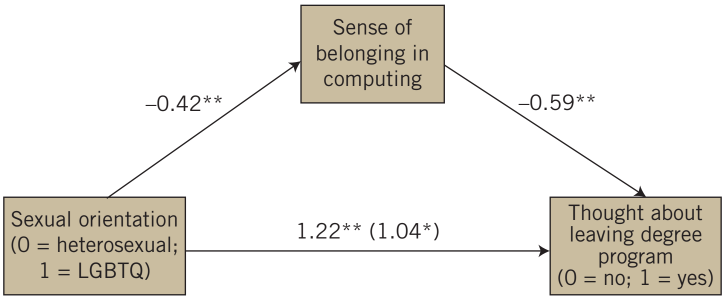


Figure 5. Indirect effect of "belonging" on LGBTQ versus heterosexual students' greater tendency to think about leaving their degree programs. Regression coefficients are unstandardized. Coefficient in parentheses is the effect of sexual orientation on thoughts about leaving, controlling for belonging, where *$p < .01$ and **$p < .001$.

General Discussion

Our current work suggests that LGBTQ students are more likely to think about leaving computing because they feel a lower sense of belonging compared to heterosexual students. Although we found that LGBTQ women were no more likely to report thinking about leaving their program than their peers, this group of women showed the lowest sense of belonging in computing compared to other students. Thus, our data suggest that whereas all members of the LGBTQ community might be at higher risk of leaving computing compared to heterosexual-identifying students, women within the LGBTQ community are particularly at risk, perhaps due to the fact that they belong to two marginalized groups: women and non-heterosexual individuals.

One puzzling finding in the current work is that whereas LGBTQ graduate students reported that they had seriously considered leaving their program to a greater degree than heterosexual peers, this pattern did not occur among undergraduate students. Note that while approximately 12 percent of all undergraduate students indicated that they had considered leaving their major, 28 percent of all graduate students considered leaving. Greater variability in graduate students' thoughts about leaving might have revealed sexual orientation disparities that were too difficult to detect in study 1 due to low variability in thoughts about leaving among undergraduate students.

In this work, we focused solely on LGBTQ students' sense of fit and thoughts about leaving within the context of computing. It is possible that LGBTQ students feel a more secure sense of belonging in fields such as the social sciences and humanities, which focus on understanding social issues (sexuality) in the curriculum.[9] Future work should compare LGBTQ students' sense of belonging and persistence across many different disciplines in order to assess whether LGBTQ students' sense of belonging is localized to specific disciplines or widespread across the academy. Such findings would help administrators and educators target their efforts to create inclusive environments for all students.

Our work does not pinpoint the vehicles behind LGBTQ students' low sense of belonging in computing—for example, does the computing culture subtly transmit a blanketed message that the LGBTQ community does not belong in computing via a lack of LGBTQ role models? This is one explanation for women's low sense of belonging in many STEM fields[19,20] and could be one mechanism behind LGBTQ students' tendency to feel as though they do not "fit" in computing as much as heterosexual students. Future research should explore these and other possible explanations for low belonging among the LGBTQ community in order to develop clear-cut intervention strategies to achieve greater inclusivity in the academy.

It is in the field's best interest to ensure that a broad array of individuals with a diversity of experiences and perspectives contribute to computing. To attract and retain a breadth of talent, computing must foster a safe and inclusive environment for all individuals. In this way, the computing field, and society more generally, can benefit from a sturdy, innovative computing labor force. Equally important, all individuals would be free to pursue a career in the financially and personally rewarding field of computing. ■

Acknowledgments

This research is sponsored by two grants from the National Science Foundation awarded to the Computing Research Association: CNS-1246649 and DUE-1431112. We thank Jessica Cundiff and Ama Nyame-Mensah for assistance with data collection. We also thank participating departments of the Data Buddies Project for contributing to the data source for this article (see http://cra.org/cerp/data-buddies/).

References

1. O. Wright, "Alan Turing Gets Royal Pardon for 'Gross Indecency'—61 Years after He Poisoned Himself," *The Independent*, 23 Dec. 2013; www.independent.co.uk/news/people/news/alan-turing-gets-royal-pardon-for-gross-indecency-61-years-after-he-poisoned-himself-9023116.html..
2. C. Kitzinger, "Heteronormativity in Action: Reproducing the Heterosexual Nuclear Family in After-Hours Medical Calls," *Social Problems*, vol. 52, no. 4, 2005, pp. 477–498.
3. R.F. Baumeister and M.R. Leary, "The Need to Belong: Desire for Interpersonal Attachments as a Fundamental Human Motivation," *Psychological Bulletin*, vol. 117, no. 3, 1995, pp. 497–529.
4. E.L. Deci and R.M. Ryan, "The 'What' and 'Why' of Goal Pursuits: Human Needs and the Self-Determination of Behavior," *Psychological Inquiry*, vol. 11, no. 4, 2000, pp. 227–268.
5. L.H. Anderman and T.M. Freeman, "Motivating Students, Improving Schools: The Legacy of Carol Midgley," *Advances in Motivation and Achievement*, M. Maehr and P. Pintrich, eds., Elsevier, 2004, pp. 27–63.
6. G.M. Walton and G.L. Cohen, "A Question of Belonging: Race, Social Fit, and Achievement," *J. Personality and Social Psychology,* vol. 92, no. 1, 2007, pp. 82–96.
7. S.R. Rankin, "Campus Climate for Gay, Lesbian, Bisexual, and Transgender People: A National Perspective," Policy Inst. Nat'l Gay and Lesbian Task Force, 2010.
8. M.S. Schneider and A. Dimito, "Factors Influencing the Career and Academic Choices of Lesbian, Gay, Bisexual, and Transgender People," *J. Homosexuality*, vol. 57, no. 10, 2010, pp. 1355–1369.
9. C. Ingraham, "The Heterosexual Imaginary: Feminist Sociology and Theories of Gender," *Sociological Theory*, vol. 12, no. 2, 1994, pp. 203–219.
10. D. Bilimoria and A.J. Stewart, "'Do Ask, Don't Tell': The Academic Climate for Lesbian, Gay, Bisexual, and Transgender Faculty in Science And Engineering," *Nat'l Women's Studies Assoc. J.*, vol. 21, no. 2, 2009, pp. 85–103.
11. S. Cheryan et al., "Ambient Belonging: How Stereotypical Cues Impact Gender Participation in Computer Science," *J. Personality and Social Psychology,* vol. 97, no. 6, 2009, pp. 1045–1060.
12. K. Lewis et al., "Fitting in to Move Forward: Using a Belonging Framework to Understand Gender Disparities in Persistence in the Physical Sciences, Technology, Engineering and Mathematics (pSTEM)," in review, *J. Educational Psychology.*
13. V. Purdie-Vaughns and R.P. Eibach, "Intersectional Invisibility: The Distinctive Advantages and Disadvantages of Multiple Subordinate-Group Identities," *Sex Roles*, vol. 59, no. 5, 2008, pp. 377–391.
14. K. Leung et al., "Multicultural Experience Enhances Creativity: The When and How," *Am. Psychologist,* vol. 63, no. 3, 2008, pp. 169–181.
15. D.A. Thomas, "Diversity as Strategy," *Harvard Business Rev.*, Sept. 2004; https://hbr.org/2004/09/diversity-as-strategy.
16. L.J. Cronbach, "Coefficient Alpha and the Internal Structure of Tests," *Psychometrika*, vol. 16, no. 3, 1951, pp. 297–334.
17. J. Nunnaly, *Psychometric Theory*, McGraw-Hill, 1978.
18. K.J. Preacher and G.J. Leonardelli, "Calculation for the Sobel Test: An Interactive Calculation Tool for Mediation Tests," 2010–2016; http://quantpsy.org/sobel/sobel.htm.
19. J.G. Stout et al., "STEMing the Tide: Using Ingroup Experts to Inoculate Women's Self-Concept in Science, Technology, Engineering, and Mathematics (STEM)," *J. Personality and Social Psychology,* vol. 100, no. 2, 2011, pp. 255–270.
20. N. Dasgupta, "Ingroup Experts and Peers as Social Vaccines Who Inoculate the Self-Concept: The Stereotype Inoculation Model," *Psychological Inquiry*, vol. 22, no. 4, 2011, pp. 231–246.

Jane G. Stout is the director of the Computing Research Association's Center for Evaluating the Research Pipeline (CERP), which conducts evaluation for diversity interventions in the computing community. She also leads basic social science research at CERP, aiming to understand predictors of persistence among underrepresented individuals in pursuit of computing careers. Stout received her PhD in social psychology from the University of Massachusetts Amherst. Contact her at jane@cra.org.

Heather M. Wright is research associate at the Computing Research Association's Center for Evaluating the Research are centered on promoting social justice among all groups of individuals, especially those who are underrepresented in computer science and other related fields. Wright received a BS in sociology from Radford University, with minors in technical and business writing and women's studies. Contact her at heather@cra.org.

IEEE computer society | CELEBRATING 70 YEARS

PURPOSE: The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field.
MEMBERSHIP: Members receive the monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.
COMPUTER SOCIETY WEBSITE: www.computer.org
OMBUDSMAN: Direct unresolved complaints to ombudsman@computer.org.
CHAPTERS: Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.
AVAILABLE INFORMATION: To check membership status, report an address change, or obtain more information on any of the following, email Customer Service at help@computer.org or call +1 714 821 8380 (international) or our toll-free number, +1 800 272 6657 (US):

- Membership applications
- Publications catalog
- Draft standards and order forms
- Technical committee list
- Technical committee application
- Chapter start-up procedures
- Student scholarship information
- Volunteer leaders/staff directory
- IEEE senior member grade application (requires 10 years practice and significant performance in five of those 10)

PUBLICATIONS AND ACTIVITIES

Computer: The flagship publication of the IEEE Computer Society, *Computer*, publishes peer-reviewed technical content that covers all aspects of computer science, computer engineering, technology, and applications.
Periodicals: The society publishes 13 magazines, 18 transactions, and one letters. Refer to membership application or request information as noted above.
Conference Proceedings & Books: Conference Publishing Services publishes more than 175 titles every year.
Standards Working Groups: More than 150 groups produce IEEE standards used throughout the world.
Technical Committees: TCs provide professional interaction in more than 45 technical areas and directly influence computer engineering conferences and publications.
Conferences/Education: The society holds about 200 conferences each year and sponsors many educational activities, including computing science accreditation.
Certifications: The society offers two software developer credentials. For more information, visit www.computer.org/certification.

NEXT BOARD MEETING

6–10 June 2016, Buckhead, Atlanta, GA, USA

COMPUTER SOCIETY OFFICES

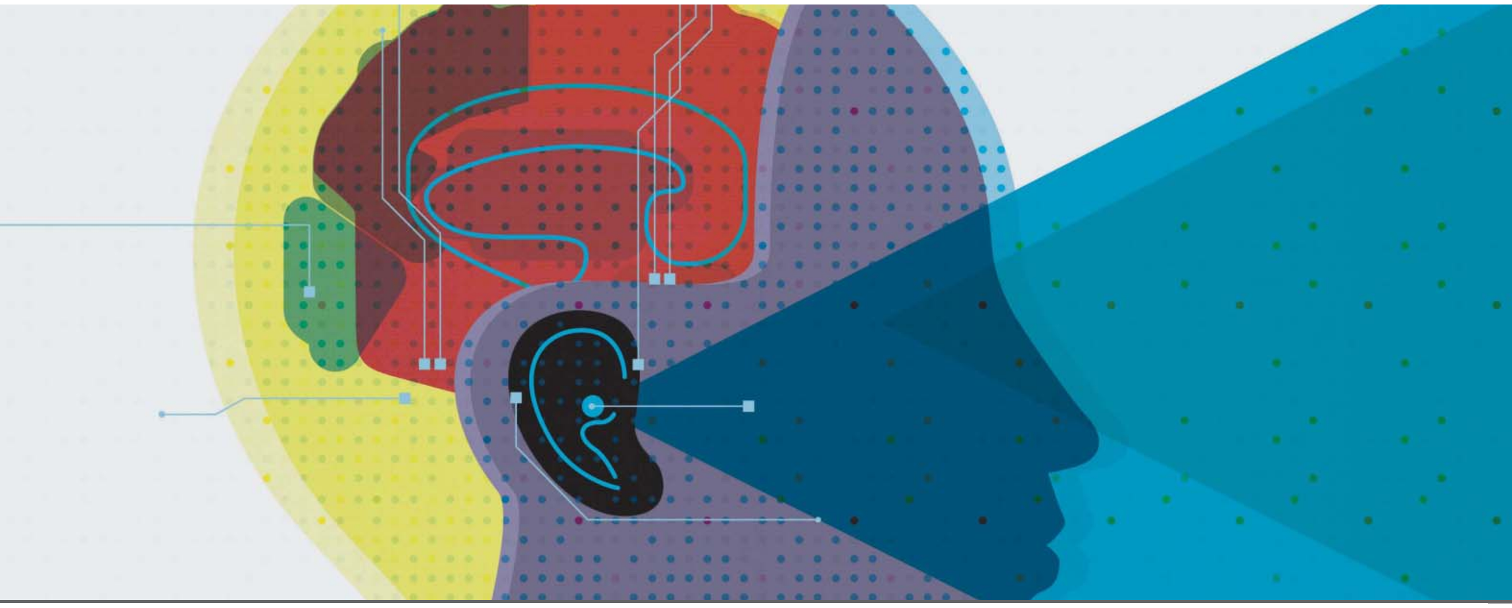
Washington, D.C.: 2001 L St., Ste. 700, Washington, D.C. 20036-4928
Phone: +1 202 371 0101 • **Fax:** +1 202 728 9614
Email: hq.ofc@computer.org
Los Alamitos: 10662 Los Vaqueros Circle, Los Alamitos, CA 90720
Phone: +1 714 821 8380
Email: help@computer.org

MEMBERSHIP & PUBLICATION ORDERS

Phone: +1 800 272 6657 • **Fax:** +1 714 821 4641 • Email: help@computer.org
Asia/Pacific: Watanabe Building, 1-4-2 Minami-Aoyama, Minato-ku, Tokyo 107-0062, Japan
Phone: +81 3 3408 3118 • **Fax:** +81 3 3408 3553
Email: tokyo.ofc@computer.org

IEEE

revised 6 April 2016



Julian Scholars: Broadening Participation of Low-Income, First-Generation Computer Science Majors

Gloria Childress Townsend | DePauw University
Kay Sloan | Rockman et al

Using funding from the US National Science Foundation, DePauw University launched a program for low-income, first-generation scholars in STEM fields. Cornerstones of the Julian Scholars program include a week-long summer research experience bridging high school and college, common classes for each cohort, mentoring, one-on-one resume and internship/research counseling, and scholarships.

In 2009, DePauw University received a four-year Scholarships in Science/Technology/Education/Mathematics (S-STEM) grant from the US National Science Foundation (NSF) to initiate the program "Julian Scholarships: Recruiting, Retaining and Supporting Local, First-Generation Undergraduate Scientists" (Julian Scholars). The project tapped a new pool of high school students for DePauw and provided a more diverse set of future STEM professionals. Project principal investigators (PIs) selected Julian Scholars with demonstrated financial need from high school students in rural counties surrounding DePauw University's campus in West-Central Indiana and from a nearby urban county that includes Indianapolis. The S-STEM scholarships completed a package of tuition, room, and board supplied by DePauw.

Percy Lavon Julian, a prominent African-American chemist and 1920 DePauw graduate, accomplished many things in his career, including the synthesis of cortisone; the synthesis of physostigmine, a glaucoma drug; and the creation of aerofoam, a flame retardant that saved many lives during World War II. As a

 1521-9615/16/$33.00 © 2016 IEEE Copublished by the IEEE CS and the AIP

The program's director did not access students' private financial information, and her database doesn't distinguish between parents' having some or no college experience, so it is impossible to compare statistics using exactly the same first-generation and low-income definitions.

first-generation student who overcame great odds to obtain his education, Julian is an exceptional role model for first-generation students today.

Underrepresented in STEM: Low-Income, First-Generation Computer Science Students

Reliable sources for data about low-income, first-generation students define terms differently from the Julian Scholars' definitions. The program's director did not access students' private financial information, and her database doesn't distinguish between parents' having some or no college experience, so it is impossible to compare statistics using exactly the same first-generation and low-income definitions. Similarly, sources break down categories differently for criteria such as institutional description, student race, age, and marital status. The sources[1,2] draw data from either the Postsecondary Education Transcript Study of the National Education Longitudinal Study of 1988 or the National Center for Education Statistics' Beginning Postsecondary Study.

In Table 1, column one displays explicit definitions of low-income, first-generation, institutional type, and major subject (along with sources of the definitions); column two contains the sources' graduation rates; and column three lists graduation rates for Julian Scholars. Although the column one definitions don't match perfectly, including Table 1 data is important: the Julian Scholars' percentages regarding both STEM and computer science majors exceed (by a large margin) each of the cited STEM and computer science rates for first-generation and low-income students, regardless of nuances in definitions of the two terms. Furthermore, the cited percentages for low-income and first-generation students provide documented ballpark ranges for comparing the Julian Scholars' percentages.

The last three rows of Table 1 reveal that all 24 of the Julian Scholars graduated from four-year colleges or will soon. Rounding errors account for the fact that the last three rows sum to more than 100 percent. Of the original 24 students, one graduated from DePauw with a degree in English, one is predicted to graduate with a degree in English from a nearby public institution, one earned a psychology degree from a large research university in Indianapolis, one earned a degree in biology from another private Indiana college, and one earned an engineering degree from a prestigious private school close to DePauw. Nineteen of the 24 finished the Julian Scholars program by graduating from DePauw with a STEM degree (79 percent). Of these 19, 12 are pursuing computing careers in industry or are in graduate school studying computer science, cyber law, or computer engineering (63 percent). Nine of the 12 work in industry, and the remaining three attend one of the three graduate school categories listed above. Ten of the 12 received DePauw degrees in computer science. Three of the 12 are female, and of those three, one is Asian, a minority for STEM majors but not a traditional minority for computer science.

Again, all Julian Scholars must display significant financial need and have parents who didn't complete an undergraduate education themselves. However the terms for low-income and first-generation are defined (and all definitions appear in column one of Table 1), the graduation rates for the Julian Scholars program (in column three) are larger than the percentages in column two—in most cases, much larger. In particular, the graduation rate for Julian Scholars who are now employed in the computing field or are enrolled in a computing-related graduate school program is 50 percent (50 percent of the Julian Scholars graduated or will graduate soon with majors that are not computing-related), while national computer science graduation rates (two different definitions, also in bold) for low-income, first-generation students are both less than 10 percent.

Description of the Julian Scholars Program

The Julian Scholars program operated from 2009–2015, with NSF funding in the years 2009–2013. The program's success in graduating STEM students and computer science students, in particular, suggests a closer investigation of the program's components.

Table 1. National-level and Julian Scholars' percentages of low-income, first-generation scholars.

Definitions of low-income, first-generation students, institutional types, and major subjects	Other (%)	Julian Scholars (%)
Graduate from four-year institution and parents have some college experience[1]	62.7	
Graduate from a four-year institution and parents have no college experience[1]	46.7	
Graduate from a four-year institution and parents' income < $25,000[1]	52.3	
Graduate from a four-year institution and parents' income between $25,000 and $49,999, inclusive[1]	62.1	
Graduate from a four-year institution majoring in a STEM field or architecture and parents have some college experience[1]	16.0	
Graduate from a four-year institution majoring in a STEM field or architecture and parents have no college experience[1]	13.1	
Graduate from a four-year institution majoring in a STEM field or architecture and parents have bachelor's degrees or higher[1]	20.5	
Graduate from a four-year institution majoring in computer science and parents have some college experience[1]	3.7	
Graduate from a four-year institution majoring in computer science and parents have no college experience[1]	2.0	
Graduate from a public four-year institution and parents have no college experience and parents' income < $25,000[2]	34.0	
Graduate from a private nonprofit four-year institution and parents have no college experience and parents' income < $25,000[2]	43.0	
Graduate from a four-year institution with a degree in a STEM field and parents have no college experience and parents' income < $25,000[2]	19.0	
Graduate from a four-year institution with a degree in computer science or engineering and parents have no college experience and parents' income < $25,000[2]	10.0	
Graduate from DePauw with a STEM degree and parents' income qualifies them for financial aid (*N* = 19)		79.0
Graduate from DePauw to enter computing career or computing-related graduate degree program and parents' income qualifies them for financial aid (*N* = 12)		50.0
Graduate from a four-year public or private institution with a STEM degree and parents' income qualifies them for financial aid (*N* = 22)		92.0
Predicted to graduate from a four-year public institution with an English degree and parents' income qualifies them for financial aid (*N* = 1)		4.2
Graduate from a four-year nonprofit private institution with an English degree and parents' income qualifies them for financial aid (*N* = 1)		4.2

For each of the three recruiting years, the list of students meeting the grant's criteria contained about 50 names.

Recruiting the Julian Scholars

DePauw recruited students to be Julian Scholars from high schools in Indianapolis (a 40-minute drive from the university) and from the six counties that surround Putnam County (where DePauw resides) and Putnam County itself, in West-Central Indiana. Guidelines for selection included an established aptitude and interest in science and mathematics, an unweighted GPA of 3.7, and combined mathematics and critical reading SAT scores of 1200 or higher. All students displayed significant financial need and had parents who didn't complete an undergraduate education. We chose to target these two populations based on studies suggesting that first-generation students prefer to attend undergraduate institutions in close proximity to their homes.[3] In addition, the preceding year's graduation roster included no first-generation students from half of the surrounding counties, two each from two targeted counties, and five from the remaining county, for a total of only nine graduating, first-generation students representing all the students living within a few miles of DePauw. Julian Scholarships helped correct the shortfall.

The Office of Admission sent a Julian Scholars brochure during its general recruitment of new students (from a purchased mailing list) in the targeted Indiana counties. For each of the three recruiting years, the list of students meeting the grant's criteria contained about 50 names. The grant's PI then ranked the students by GPA and SAT scores, personally recruited admitted students in rank order by telephone, and then interviewed them on campus. Many of the contacted students declined an interview. The interview questions purposely included conversational explorations of students' intellectual liveliness and curiosity and deflected more stressful inquiries common in highly selective honors programs, such as DePauw's Honor Scholar program. For each of the 3 recruiting years, the PI chose 8 qualified students from approximately 10 after campus interviews, choosing students on a rolling basis.

The NSF provided US$10,000 per year in scholarship funding for each Julian Scholar during the first and second years of college; DePauw University funded $20,000 per student in years three and four. The scholars maintained a general GPA of 2.5 or higher in a STEM major to remain in the program. DePauw continued to pay the scholarship funding for the lone student who dropped out of the Julian Scholars program to major in English and remain at DePauw.

Bridge to Science

After being accepted into the program, Julian Scholars spent one week in the summer preceding each cohort's fall matriculation conducting science research in one of three project groups: Science Research Fellows (SRF; an honors program for STEM students), internal grants, and external grants. The eight Julian Scholars in each year's cohort divided into two groups of four for their bridge to Bridge to Science (B2S) research projects. The groups included university students and professors already involved in laboratory and field studies, with the college researchers (students and professors) serving as informal peer mentors. One study evaluated summer programs for graduating high school seniors (such as the Drexel University Computing Academy), finding that students "enter fall with confidence, knowing how to get things done."[4] Another source similarly advocated programs such as B2S: "Summer bridge programs helped students gain experience with registering for classes, finding classrooms on campus, and going to the bookstore."[5]

The Julian Scholars met the science professors and their upper-class science student researchers (who acted as role models) in a relaxed and intimate environment. The SRF assistant director provided social event programming in the form of picnics, movies, and so on for the entire summer research community of approximately 100 students. The university community, families, and friends attended the poster session held at the conclusion of each B2S week. The poster sessions instilled confidence and communication skills, while providing line items for students' resumes. Professors displayed the posters in their respective science suites. Some students were able to attend discipline-specific conferences to present their posters a second time. All Julian Scholars presented their posters again in November in the school-wide poster session for summer research.

Panels of older first-generation students (including the mentors) explained how they obtained their research and internship experiences and described typical days doing research or working in an internship.

Registration Counseling, Common Classes, and Professor Selections

During B2S, whose timing was chosen to precede summer registration, Julian Scholars met with the PI as a group, and individually, to receive registration counseling. Ordinarily, first-generation students list online preferences for their first fall's classes during June's online registration—an unfortunate course-selection mechanism for students with parents who have little or no college experience. The PI stressed the importance of careful course selection and steered students toward an appropriate science class and additional courses based on their high school subject matter preferences as well as her knowledge of DePauw class content. In addition, communication with the DePauw registrar placed students in common science classes with instructors known to be especially helpful and outgoing with students. The study cited earlier[4] supported the methods described here by recommending an "intrusive advising process … in participants' initial course selection" for students in a project involving "pre-freshmen," low-income, first-generation scholars.

Julian Mentors in Addition to Regular DePauw Mentors

The PI chose (based on faculty members' recommendations and her own interviews) two student mentors for each cohort: upper-class DePauw STEM students to live with the Julian Scholars and provide evening activities (sports, movies, hiking, and so on) during the summer week. The mentors' demographic profiles matched Julian students' profiles, and during the academic year, they continued to informally monitor their four cohort members (meeting individually for coffee once a month and discussing items from a list provided by the director while chatting extemporaneously) and composed notes from each meeting, reporting back to the PI.

Once the fall school year began for first-year students, the university supplied additional student mentors (1 or 2 for each group of 12 to 14 first-year seminar students) and a well-coordinated series of social and academic activities with these mentors. DePauw requires each first-year students to enroll in a first-year seminar. Two or three student mentors (plus the PI) served each Julian Scholar, resulting in "intrusive mentoring," as described in a 2008 presentation.[6] The presentation concerned a study conducted with 205 Indiana 21st Century Scholars and 76 support staff members and administrators, and it recommended intrusive mentoring, a summer bridging experience, and learning communities, each of which was a component of the Julian Scholars program.

A Comprehensive Advisor

Julian Scholars received advising through their first-year seminars and in most cases changed advisors after declaring a major at the conclusion of their sophomore years. The PI maintained access to all electronic advising records for each student, acting as a second, comprehensive advisor for the Julian Scholars by communicating with the standard advisor, the career center, the financial aid office, students' instructors, and so on, as advocated in the literature.[7]

Tutors Who Are Also Mentors or Julian Scholars

The literature about first-generation students[5,8,9] calls for "strong tutorial and supplemental instruction." The PI added to the literature's commonsense advice by hiring tutors for the Julian Scholars, people who were also Julian student mentors or current Julian students themselves after they became upper-class students.

Personal Resume Consultation and Editing

The PI asked the Julian Scholars to visit the career center to write initial drafts of their resumes. During a fall meeting in their first semester, she then helped them craft line items for their resumes, including selection for the Julian Scholars program, B2S, and their two poster presentations. Following the group meeting, the PI held individual meetings with each Julian student to help the student perform final resume edits. Researchers recommend this sort of collaborative planning for all career transition processes that first-generation students undergo.[10]

Participants shared how they had used the various resources and support mechanisms and how these items had steered them through coursework, college life, and summer and semester research experiences.

Counseling for Research and Internship Opportunities

The PI held two separate seminars for the Julian Scholars, during each cohort's initial fall semester, to discuss both summer research and summer internship opportunities and to hand out written information and lists of STEM websites about those opportunities. Panels of older first-generation students (including the mentors) explained how they obtained their research and internship experiences and described typical days doing research or working in an internship. The director volunteered to write recommendations for each Julian Scholar based on B2S research and poster presentations, and all of the students successfully obtained internships and/or research opportunities, such as Research Experiences for Undergraduates from the NSF.

Community Building

One of the most important components of the Julian Scholars project sought to build community among its members. As one study pointed out, "first-generation students tended to report lower ratings of belonging,"[11] and another researcher wrote that "extracurricular activities bond students, faculty, staff, and the institution."[8] "Representing the university culture in terms of interdependence (i.e., being part of a community) reduced [the] sense of academic difficulty and eliminated the performance gap" between first-generation and continuing-generation students.[12] Each of the preceding research studies encouraged the PI to design social activities for the Julian Scholars and mentors. A recurring favorite of the students was the "science ice cream party," featuring ice cream made with liquid nitrogen.

Although frequent participation in social activities might be one avenue for integrating students into campus life, one study's results[3] suggested that participation and involvement in academic activities could be just as (or more) important to first-generation students than social activities, especially those involving interactions with faculty members. The literature also indicates that "first-generation students tend to be less engaged than other students, perhaps in part because they know less about the importance of engagement or how to get involved in productive activities."[13] Parents of first-generation students might not understand the importance of engagement and find many activities "off-putting."[14,15]

Accordingly, the Julian and SRF directors worked together to share programming for students in both programs (mainly STEM speakers) when appropriate, taking advantage of the SRF budget. The PI also encouraged all 11 of the female Julian Scholar to attend the Women in Science monthly lunches (with speakers). Additional communities that the Scholars were urged to join included the Underrepresented in Science club, the robotics club, the programming team, and computing organizations such as ACM, ACM-W chapters, and Students in Technology, Academia, Research, and Service (STARS) Alliance.

The Julian Scholars as a group seemed proud of their community, electing to wear graduation cords in the colors of Indiana's flag to represent their program. A large contingent spoke to a new first-generation scholars group at an ice cream social to give their advice for success at DePauw.

Evaluation

A two-year external study conducted by independent research and evaluation firm Rockman et al used both quantitative and qualitative methods to gauge the program's success in recruiting and supporting Julian Scholars. Although the report focuses largely on feedback gathered in year two, it also draws on the results from the year-one background and follow-up surveys to gauge changes from year to year.

Anonymous Student Surveys

The external study began in fall 2011, two years into the project, with an online presurvey completed by 83 percent of the 24 Julian Scholars (N = 20; 4 first-years, 8 sophomores, and 8 juniors). The survey, which gathered data on students' confidence, research experiences, and career aspirations, served as a baseline for cohort 3 students, then DePauw first-year students; a retrospective survey served as a

Computing in SCIENCE & ENGINEERING | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page | Qmags THE WORLD'S NEWSSTAND®

Julian Scholars also find the research, internship, and career opportunities open to them invaluable, but emphasize that without help managing their time, preparing resumes, and meeting application deadlines, they would have taken advantage of fewer opportunities.

baseline for the sophomores and juniors in cohorts 2 and 1, respectively. As part of the year-one data collection, students also completed a follow-up postsurvey in spring 2012 (N = 20; 4 first-years, 8 sophomores, and 8 juniors). For year two, students completed the same postsurvey in spring 2013 (N = 17; 6 sophomores, 5 juniors, and 6 seniors). The respondent groups varied slightly from survey to survey. (Nonrespondents included students who had graduated early or were studying abroad.)

Focus Groups

At the ends of both academic years (2012 and 2013), a sample of Julian Scholars took part in focus groups. Seven students, whose comments and quotes are included in this report, took part in the 2013 discussions, elaborating on some of the same questions asked in the surveys and providing more in-depth information about their experiences as Julian Scholars. Participants shared how they had used the various resources and support mechanisms and how these items had steered them through coursework, college life, and summer and semester research experiences, ultimately helping them prepare for and transition to careers and postgraduate work. Students also shared personal experiences and observations about how their experiences differed from other students'.

Interviews with the Project Director

Over the course of the two-year study, the evaluator conducted several informal interviews with the Julian Scholars program director about the program's evolution. The interviews helped the evaluator understand the program's components. As an example, she learned about extensive resume preparation help and subsequently designed survey questions to allow Julian Scholars to rate the resume aid they received.

Data Analysis

For a subset of scaled items pertaining to students' ratings of the importance of program components, their confidence in skills and knowledge, and their views on careers in science, the evaluator ran paired sample t-tests to look for statistically significant differences between the 2012 and 2013 (N = 16) results. In the findings reported here, the means vary slightly because the respondent pairs differed from analysis to analysis.

For the more qualitative data from open-ended survey questions and focus groups, the evaluator reviewed responses for emerging themes and developed coding systems based on recurrent comments or topics. The small numbers of Julian Scholars and focus group participants limited the analysis somewhat, but there were some clear and consistent trends across questions, years, and groups.

Evaluator's Summary

Scholarships made it possible for Julian students to attend DePauw, but the less tangible forms of support have enabled them to succeed. Julian Scholars indicate that they would not likely have attended or even considered DePauw University had it not been for the scholarship. They also say that the guidance they received, primarily from the director but also from student mentors and peers, is what helped them successfully steer through their undergraduate years and emerge with the confidence and research experience that will make them attractive candidates in competitive job markets and for graduate school admissions.

Julian Scholars also find the research, internship, and career opportunities open to them invaluable, but emphasize that without help managing their time, preparing resumes, and meeting application deadlines, they would have taken advantage of fewer opportunities. Whether they chose computer science or other STEM areas, Julian Scholars feel that the insistence, again from the director, that they start considering internships and other research opportunities early in their programs and each academic year gives them an advantage over other applicants. They also believe that, compared to other DePauw students, they have a stronger support system.

The Julian Scholars have gained confidence in various skills during their undergraduate programs, including the 21st-century skills routinely listed as those needed to succeed in the workplace. Comparisons

Table 2. Students' ratings of Julian opportunities and support, by year.

Julian Scholars program opportunities	2012 mean	2013 mean	Standard deviation	Statistical significance (*p* value)
Scholarship and financial aid	3.95	4.00	0.250	0.333
Career preparation	3.45	3.82	0.775	0.216
Director's resume preparation help	N/A	3.76	N/A	N/A
Opportunities for undergraduate research and internships	3.85	3.71	0.750	0.333
Being part of a learning community	3.60	3.59	0.574	0.669
Mentoring from the director	N/A	3.59	N/A	N/A
Future job placement	3.40	3.59	0.885	0.111
Support from older students and tutors	3.50	3.47	0.719	0.497
Summer research experiences	3.20	3.24	1.181	0.835
Access to university resources	3.55	2.82	1.147	0.008

between 2011 survey responses and those from 2013 show statistically significant gains in confidence in time management, presentation, and collaboration skills, as well as in academic skills, especially mathematics. Computer science majors reported greater confidence in presentation skills than biology/biochemistry majors. Changes in confidence from 2011 to 2013 were smallest for writing skills.

Julian Scholars often have double majors, in the sciences and in the humanities, and consider having a science scholarship to a liberal arts college one of the most enriching (and least expected) benefits of the program. Both computer science and other science majors have taken advantage of what a liberal arts institution offers, with second majors in history, music, and political science. Even those with a single major have taken advantage of humanities offerings in the form of classes in areas such as studio art, women's studies, Africana studies, art history, and creative writing.

The perception of community among Julian Scholars contributes to their sense of confidence and success. Julian students have formed strong bonds not just with the director but with each other and peer mentors. Bonds with student mentors appear to be strongest when Julian Scholars are in the same field, and often in the same lab.

Support and Success

The Julian Scholars program provides wide-ranging support for students, and during focus groups and in response to open-ended questions, students described the benefits of what one student called his "supporting cast." Some of the support is fairly structured, such as the summer bridge program, and some is more fluid, such as the ongoing interactions with peer mentors and community-building activities. The discussion below shares the 2013 survey ratings for the various support mechanisms, comparing some to the 2011 and 2012 ratings and describing differences based on students' choice of major.

Julian Scholars' feedback about the importance of the support they receive is consistent and positive. In both the 2012 (N = 20) and 2013 ($N = 17$) follow-up surveys, students assigned ratings of 3 or higher to almost all the forms of support listed (on a 4-point scale, where 1 was not very important, 2 was somewhat important, 3 was reasonably important, and 4 was very important). Table 2 shows the ratings in descending order, based on the means for the 2013 responses. The table also includes the standard deviations and significance levels from comparisons of the 2012 and 2013 responses.

The most highly rated form of support across years and majors was what brought students to DePauw in the first place: financial aid ($M = 3.95$ in 2012 and 4.00 in 2013). In 2013, students, regardless of major, assigned higher ratings to career preparation (overall, $M = 3.82$ versus $M = 3.45$) and future job placement ($M = 3.59$ versus $M = 3.40$) than they had in 2012, which is not surprising, Julian Scholars highly value the internship and career opportunities they are introduced to through the program, including opportunities to secure summer employment.

given that students were a year further along in their programs and closer to making career decisions or entering their chosen fields. Support mechanisms that received consistently high ratings across years, and majors, were the undergraduate research opportunities and internships (M = 3.85 and 3.71), experiences that create bonds among students (M = 3.60 and 3.59), and support from older students and tutors (M = 3.50 and 3.47).

When students elaborated on their ratings and experiences on the survey and during annual focus groups, they insisted that certain things be clear: first, that they wouldn't have attended DePauw or anywhere else had they not received the scholarship. Second, they feel strongly that "it's more than just money. It's almost like a club, where we all share something in common and help each other out."

Individual Support Activities

Tests for significant differences between the administrations of the survey showed only one difference was statistically significant: the drop in the rating for "access to university resources" (M = 2.82 versus M = 3.55; p = .008; see the last row in Table 2.)

The drop might be due to a change in the survey itself and an opportunity to specifically rate the support from the Julian Scholars program director. The 2012 focus group participants pointed out that they relied more on the program director than other university resources for support and advice, so a separate option was added to the 2013 survey. Responses reflected the importance of the director's guidance, which the 2013 focus group participants described as not only making the process more personal but also as giving Julian Scholars an edge in the very competitive process of securing research or intern positions. They noted, for example, that the director "gets students thinking about internships as soon as they arrive on campus" and encourages them to "get summer plans in place during winter term." She also, according to students, shows them how to craft their resumes to make themselves "stand out among other students looking to attend grad school" or re-cast certain skills and experiences. She also suggests they apply for positions they might not have considered.

The Julian Scholars believe that they feel a greater sense of connection with the director than friends have with their advisors. According to several students, they go to her not only for questions about research opportunities and classes but about social matters. As one noted, "We talk about everything, even my roommate." More than one Julian Scholar also noted that friends or roommates who were not in the program had turned to the director for advice or requested that she be their official advisor.

One 2013 focus group participant noted that although they rely on the director for many things, it's their student mentors who are "on campus when we are freaking out in the middle of the night." When the need is not so dire, the mentors are there to "break up the monotony of studying." One student explained that her mentor not only helped her with a challenging class but also "taught me how to study." Student mentors also share an honest opinion of professors when students are choosing classes. They often serve as role models, inspiring students to join a lab or research group or become a mentor or resident assistant.

Student mentors also play a key role in fostering a sense of community among the Julian Scholars. According to one, his mentor helped him establish an initial network of friends during his first semester on campus. This network, and the friendships made among Julian Scholars during bridge week, made the transition to college smoother.

During the 2012 focus groups, students mentioned that they would like the Julian brand to be more visible on campus. Other students knew the name because of the prominent Julian Science Building but not of the Scholars program. In 2013, students seemed less bothered by this lack of recognition. Some noted that because the Julian program is new and different from other scholarship programs, they feel a special bond, and believe they have a more close-knit group than other scholarship recipients whose programs "have been around forever."

During both the 2012 and 2013 focus groups, a few students said that they would like to have more opportunities to interact and share experiences with the other Julian Scholars, more frequent social get-togethers. They tend to see students who take the same classes or share majors more than those in other departments or fields and know those in their cohorts better than they know older or younger students.

Table 3. Comparison of 2011 and 2013 survey results for self-assessment of skills.

Self-assessment	2011 mean	2013 mean	Statistical significance *p*
Your science skills and knowledge	3.27	3.50	Not significant
Your mathematical skills and knowledge	2.82	3.31	$p < .05$
Your research skills and knowing your way around a lab	3.05	3.25	Not significant
Your collaboration skills or comfort in working on a team	3.36	3.81	$p < .05$
Your interactions with professors and staff members	3.64	3.81	Not significant
Your ability to plan and organize reports	3.32	3.62	Not significant
Your writing skills	3.14	3.25	Not significant
Your presentation skills	2.91	3.38	$p < .10$
Your time management skills	2.86	3.44	$p < .05$
Your ability to balance academic work and extracurricular activities	3.09	3.44	Not significant

Finally, the Julian Scholars highly value the internship and career opportunities they are introduced to through the program, including opportunities to secure summer employment. They say that what might be as valuable as the opportunities themselves is the insistence, from the director and student mentors, that they take advantage of them. Julian Scholars also noted that their peers are less aware of these opportunities and thus less likely to take advantage of them.

An Analysis of Skills and Confidence

Results from the spring 2012 follow-up survey showed higher levels of confidence in science, mathematics, and research skills among college juniors than among first-years and sophomores. The finding wasn't surprising, given that undergraduates might still be adjusting to university life and discovering where their strengths lie. What was noteworthy was that older students' confidence extended to other skills likely needed for successful science careers, such as the ability to manage their time, collaborate and work comfortably on a team, interact with professors and staff members, and make presentations.

To further explore changes or differences in levels of confidence across years, the evaluator conducted a series of comparisons with the datasets from the 2011 background survey and 2013 follow-up survey. About a third of the gains were statistically significant at the $p < .05$ level, and all means increased. Table 3 shows the means by year, along with the standard deviations and levels of significance for each category.

All three surveys (2011, 2012, 2013) also asked students where they saw themselves in a few years, after they had graduated from college. Again, plans or visions did not vary greatly from year to year (except for less tentativeness and the word "hopefully" was less present in the 2013 survey), and students seemed to have set their sights high and retained their optimism. They either wished to be in graduate school or gainfully employed. Those envisioning the latter typically saw themselves "owning their own software company" or "at a big tech company." By 2013, those plans had materialized:

> I have already been hired as a full-time software developer. I plan to work at the company for a year and then attend graduate school for a master's in computer science, specifically, robotics.

Some trends or progressions did emerge from year to year in how students described the Julian Scholars program's impact on their schooling and career plans. In the 2011 survey, students were more likely to mention the scholarship itself or the financial assistance and explain that it had made attending DePauw, taking advantage of liberal arts offerings, and pursuing a degree and career in science possible. Some referenced the summer bridge program, the potential for other research experiences, and the way the program had given them the confidence to pursue their interests and goals:

> The Julian Scholars program helps with financial problems that I would potentially face. It also helps build a small network with the professors of my intended major. The Julian Scholars program also helps [me] discover other students in [my] major and encourages [me] to work together with them.

In 2012, students talked more about the support they had received and opportunities to work with professors and other students, explore study or career options, and make connections. Some also mentioned that the program had stretched them and, again, built their confidence:

> It has provided me with the opportunities that I am interested in. Also, it has provided me with different opportunities (research, classes) that many of my peers have not been presented with. Most importantly, the Julian Scholars program has provided me with an ability to explore many different ways to learn and grow.

In 2013, closer to realizing their goals but with a broader perspective, students referred to the foundation the program had given them, what a student quoted called the "backbone to my goals." Students were also more specific about the practical benefits of the program, such as the "incredibly helpful" resume assistance and the "soft skills for the real world":

> It has allowed me to lay a foundation to grow from. I cannot tell you the amount of times I have relied on tools taught through the program to succeed as I look for internships [and] summer research, or just be a student. I honestly don't know where I would be without this program.

We emphasize that 23 of the 24 Julian Scholars recruited using the methods outlined earlier stated that they wished to become physicians or dentists as they interviewed for the program. One lone student indicated that he wanted to pursue a career in a computing-related field. The director speculates that first-generation students are less familiar with diverse STEM career options than other students are and that they (as the vast majority of the Julian Scholars) often enter college expressing interest in a career with which they are already familiar, typically in the health sciences. Having expressed interest in preparing for a health sciences career (in most cases) due to familiarity instead of enthusiasm, first-generation students with STEM aptitude can be easily recruited to a computer science major within a department composed of strong teachers and a welcoming Computer Science 1 (CS1) course. Without suggesting cause and effect, the project did attain a rate of 50 percent of its original students in computing careers or in computing-related graduate school degree programs.

If the director's speculation is accurate, then duplicating the program could increase participation in computing for low-income, first-generation students in a school setting where students declare majors after matriculation. A more practical and cost-free idea lies in creating an environment similar to that experienced by the Julian Scholars at preregistration. One paper describes a project in which all first-year women are invited to a "content preview" of CS1 at the beginning of the registration cycle each semester.[16] Role-modeling female computer science students talk very briefly to the attendees about their opportunities in computing; then, they sit with the first-year students and teach them on-on-one about the rudiments of the first laboratory in CS1, working through the entire laboratory. The experience dispels myths about computing, causes women to feel that they have a "head start" for CS1, and gives the first-year students familiarity with the laboratory environment, a computer science professor, the content of the class, and opportunities in computing.

Connecting the "content preview" concept and structure to the Julian Scholars discussion, a potential plan could include obtaining a list of all first-generation students or first-year students (and possibly sophomores) who expressed interest in premedical/predental programs on their school applications and inviting them to a preview of CS1. A key detail lies in sending the invitation immediately prior to registration, as in the instances of both the existing "content preview" and the director's counseling of the Julian Scholars—a point in time, when many students who aren't experiencing success or satisfaction with a major related to their incoming career choice are particularly receptive to new ideas.

The "content preview" requires only the cooperation of the registrar to deliver a distribution list of email addresses plus an hour's time from several students and a professor. The potential benefit from such a simple plan to the underrepresented community in computing is enormous, as the experience at DePauw University indicates. ■

Acknowledgments

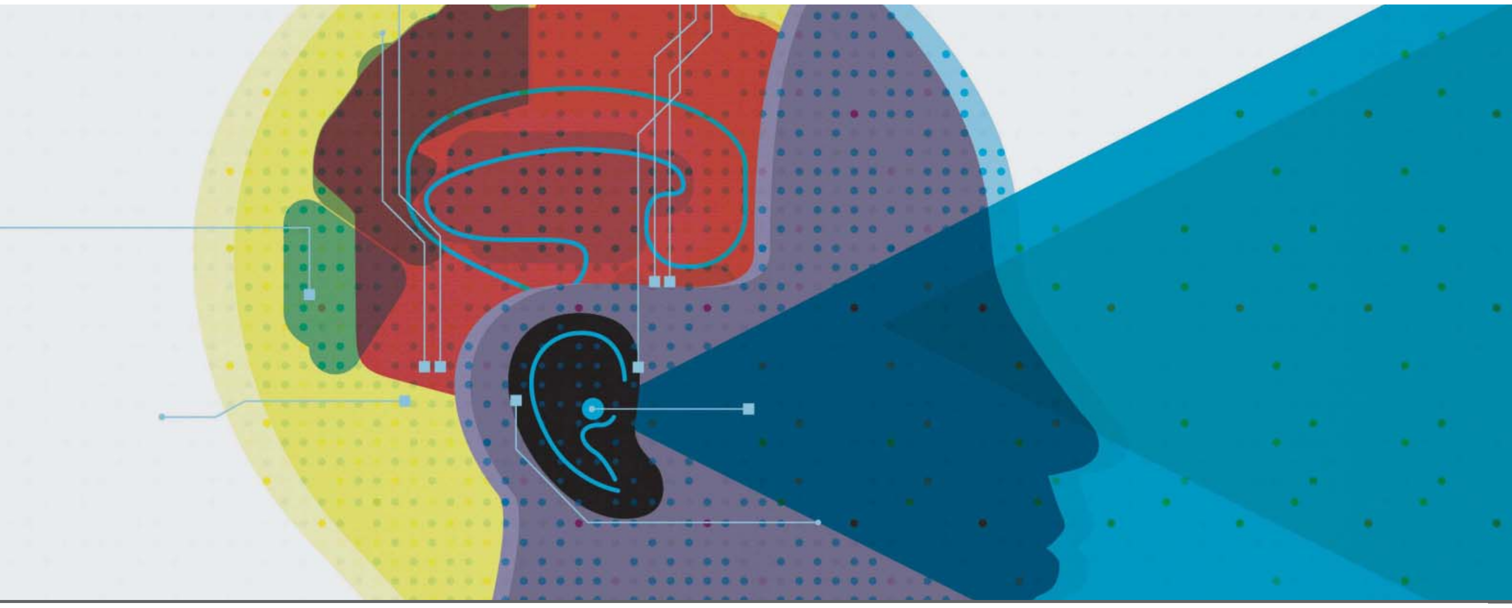
This work was supported in part by the US National Science Foundation under grant 0850184.

References

1. X. Chen and C.D. Carroll, "First-Generation Students in Postsecondary Education: A Look at Their College Transcripts," Nat'l Center Educational Statistics, July 2005; http://nces.ed.gov/pubs2005/2005171.pdf.
2. J. Engle and V. Tinto, "Moving beyond Access: College Success for Low-Income, First-Generation Students," Pell Int'l Study of Opportunity in Higher Education, Nov. 2008; http://www.pellinstitute.org/downloads/publications-Moving_Beyond_Access_2008.pdf.
3. M.M. Lohfink and M.B. Paulsen, "Comparing the Determinants of Persistence for First-Generation and Continuing-Generation Students," *J. College Student Development*, vol. 46, no. 40, 2005, pp. 409–428.
4. P.B. Thayer, "Retention of Students from First-Generation and Low-Income Backgrounds," *Opportunity Outlook*, May 2000, pp. 2–8.
5. J. Engle, A. Bermeo, and C. O'Brien, "Straight from the Source: What Works for First-Generation College Students," Pell Inst. Study of Opportunity in Higher Education, Dec. 2006; www.pellinstitute.org/downloads/publications-Straight_from_the_Source.pdf.
6. R.J. Helfenbein, R.L. Hughes. and J.S. Smith, "Twenty-First Century Scholars Qualitative Study," presentation at the Indiana Student Access and Success Conf., 4 Apr. 2008; http://education.iupui.edu/cume/publications/pdf/century.pdf.
7. P. Somers, S. Woodhouse, and J. Cofer, "Pushing the Boulder Uphill: The Persistence of First-Generation College Students," *Student Affairs Administrators in Higher Education Foundation J.*, vol. 41, no. 3, 2004, pp. 418–435.
8. T. Hicks, "Assessing Parental Involvement of First-Generation and Second-Generation College Students," *The ACT 101 J.*, vol. 9, no. 1, 2006, pp. 12–16.
9. E.C. Washburton, R. Buggarin, and A. Nuñez, "Bridging the Gap: Academic Preparation and Postsecondary Success of First-Generation Students," *Education Statistics Q.*, vol. 3, no. 3, 2001, pp. 73–77.
10. P. Tanjula, "Motivating First-Generation Students to Academic Success and College Completion," *College Student J.*, vol. 48, no. 1, 2014, pp. 133–140.
11. M. Stebleton, K. Soria, and R. Huesman, "First-Generation Students' Sense of Belonging, Mental Health, and Use of Counseling Services at Public Research Universities," *J. College Counseling*, vol. 17, no. 1, 2014, pp. 6–20.
12. N.M. Stephens et al., "Unseen Disadvantage: How American Universities' Focus on Independence Undermines the Academic Performance of First-Generation College Students," *J. Personality and Social Psychology*, vol. 102, no. 6, 2012, pp. 1178–1197.
13. G.R. Pike and G.D. Kuh, "First- and Second-Generation College Students: A Comparison of Their Engagement and Intellectual Development," *J. Higher Education*, vol. 76, no. 3, 2005, pp. 276–300.
14. M.E. Kenny and S. Stryker, "Social Network Characteristics and College Adjustment among Racially and Ethnically Diverse First-Year Students," *J. College Student Development*, vol. 37, no. 6, 1996, pp. 649–658.
15. H.B. London, "Breaking Away: A Study of First-Generation College Students and Their Families," *Am. J. Education*, vol. 97, no. 1, 1989, pp. 144–170.
16. G.C. Townsend, S. Menzel, and K. Siek, "Leveling the CS1 Playing Field," *Proc. 2007 ACM SIGCSE Tech. Symp.*, vol. 39, no. 1, 2007, pp. 331–335.

Gloria Childress Townsend is a professor of computer science at DePauw University. Her research interests include women in computing, robotics, and evolutionary computation. Townsend received a PhD in mathematics education from Indiana University. She's a member of ACM's Special Interest Group for Evolutionary Computation and Special Interest Group for Computer Science Education and its Council on Women in Computing. Contact her at gct@depauw.edu.

Kay Sloan is the managing research director at Rockman et al, an independent research and evaluation firm. Her interests include educational research, and she has designed and managed research projects and developed assessments in the fields of computing and educational technology, children's media, STEM, and the humanities. Sloan received a PhD in English and comparative literature from Indiana University. Contact her at kay@rockman.com.



STARS Computing Corps:
Enhancing Engagement of Underrepresented Students and Building Community in Computing

Jamie Payton | University of North Carolina at Charlotte
Tiffany Barnes | North Carolina State University
Kim Buch, Audrey Rorrer, and Huifang Zuo | University of North Carolina at Charlotte
Kinnis Gosha | Morehouse College
Kristine Nagel and Nannette Napier | Georgia Gwinnett College
Ebrahim Randeree and Lawrence Dennis | Florida State University

A study on the impact of participation in a national community for broadening participation in computing found many benefits for undergraduate computing students who engage in related projects, including academic, career, and personal benefits, with students who are underrepresented in computing experiencing the most benefit.

The Students & Technology in Academia, Research, and Service (STARS) Computing Corps is a nationally connected system of regional partnerships among higher education, K–12 schools, industry, and community organizations with a mission to broaden the participation of women, underrepresented minorities, and persons with disabilities in computing. The STARS approach is based on research that has shown the value of community and identity on fostering innovation,[1] increasing student learning outcomes,[2–4] and improving student retention and satisfaction, particularly for women and minorities.[3,5–7] STARS creates a community of practice across multiple institutions with a shared interest in developing college students and faculty into leaders who use computing to benefit society and seek to broaden participation in computing. Formed in 2006 by 10 colleges and universities,

 1521-9615/16/$33.00 Copublished by the IEEE CS and the AIP

Computing in Science & Engineering | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page | Qmags THE WORLD'S NEWSSTAND®

with funding from the US National Science Foundation's Broadening Participation in Computing Alliance program, STARS has become part of a national movement to democratize the computing field by increasing the pipeline and providing equity of access to computing education.[8] Since its inception, the STARS Computing Corps has grown to include 52 institutional members, including historically Black colleges and universities (HBCUs), women's institutions, undergraduate-focused institutions, and research-intensive universities. Computing Corps students have conducted more than 1,600 outreach events with more than 108,000 participants; these efforts have focused on extended interactions and mentoring, with STARS Corps students engaging in approximately 500,000 total contact hours with outreach attendees.

The goals of the STARS Computing Corps include recruiting, bridging toward professional and academic development, and retaining students, particularly those from underrepresented groups, in undergraduate and graduate computing degree programs. Figure 1 gives an overview of the STARS model. Within the STARS community of practice, each regional community is led by a STARS member university. Regional student-led activities are designed to advance the STARS central values of developing technical excellence, leadership, a sense of belonging to a computing community, and a sense of responsibility to use computing to benefit society through service and civic engagement. The national community convenes each year at the STARS Celebration, an annual conference that has been attended by more than 2,600 faculty and students from across the country in its 10-year history. The STARS Celebration offers student-focused sessions on developing leadership skills, technical skills, and preparing for graduate study or career advancement, as well as professional development sessions that focus on career advancement for faculty. At the Celebration, students and faculty also present the results of their STARS service and civic engagement projects, attend workshops on best practices and curriculum for computing outreach and service, and build community centered around common interests in civic engagement and broadening participation in computing.

Although each STARS institution shares the same goals and values, there's considerable diversity across institutions in how the STARS Computing Corps model is implemented. The Corps model ensures that each institution's intervention incorporates best practices in student engagement, as identified by several previous studies.[9–12] STARS students apply discipline-specific knowledge in service projects, beyond the scope of a traditional classroom experience,[11] enabling a dual focus on service to others as well as to their own personal and academic development.[12] Gains associated with service learning outcomes are greatest when they're discipline specific[9–11] and engage students in preparation and reflection.[9,10] These gains can be obtained across a variety of implementation strategies.[11] Outcomes derived from service learning have been demonstrated in academic performance,[9,10] self-efficacy,[9,10] leadership,[9] and career preparation.[9,11,12]

STARS students are provided opportunities for training in preparation of their service projects and for reflection of their experiences in multiple touch points throughout their Corps experience, both of which are shown to be critical components for enhancing academic outcomes.[9,10] For example, the STARS Celebration introduces new Corps students to service projects, provides them with structured training and planning, and gives veteran STARS students a platform for sharing successful service projects and discussing lessons learned.

The interventions of the STARS Computing Corps—specifically, service learning and community building—are designed to provide participants with a meaningful and impactful experience. In this article, we report the results of a study of the personal, academic, and professional benefits for students participating in these kinds of activities, as demonstrated by postintervention perceptions of their experience in the STARS Computing Corps.

Service Leadership Project Exemplars

Three examples of STARS cohorts showcase the range of institutions, implementation strategies, and service activities that can be derived from the STARS model. The common threads across all STARS Computing Corps are its implementation as a repeatable one-year program that begins and ends with an alliance-wide annual leadership conference (STARS Celebration). The Celebration inducts students, faculty, and partners into the Corps and provides support for guided planning of regional cohort activities built around the STARS central values. At some institutions, STARS operates as a co-curricular activity or club; at others, it's a curricular intervention incorporated into existing service learning programs or as a new credit-bearing computing course or seminar. Regardless

Computing in SCIENCE & ENGINEERING | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page

Figure 1. The STARS Computing Corps model. The STARS central values form the outer ring, student-led Corps activities that advance these values fall in the middle ring, and expected outcomes form the inner ring, as measured by leading (bottom) and lagging (top) indicators

of the institutional implementation mechanism, tiered mentoring is deployed throughout the Corps, and members engage in a minimum of five hours per week during each semester of participation. Local Corps meet weekly or biweekly, with goals of tiered mentoring, project planning, discussion, and reflection. Service projects apply computing principles in K–12 outreach, peer outreach and mentoring, community service, or research and internship opportunities. Each Corps selects the service areas that leverage theirs strengths and are compatible within their institutional contexts. The three types of STARS implementation featured in this section serve as exemplars, highlighting organizational flexibility in implementing service learning and engagement in a national community for broadening participation in computing.

Near Peer Mentoring

Georgia Gwinnett College (GGC) is a public, four-year access institution that opened its doors in 2006; since then, it has quickly grown from 118 to more than 11,000 students. The college is a minority-majority institution with one of the most diverse campus populations in the southeast region. Gwinnett County has the largest number of Hispanics and Asians in Georgia, and first-generation students make up 33 percent of the student population. GGC began implementing the STARS model through a co-curricular mechanism and recently received approval to integrate elements of the STARS model into a new service learning course. STARS project teams at GGC primarily conduct tiered mentoring through K–12 computing outreach, with programs that are designed to engage, excite, and introduce youth to computing concepts and potential careers. For example, the GGC STARS's Super Saturday series is a hands-on workshop designed to expose and inspire middle school girls to age-appropriate technology through creative and fun computing and science activities in a welcoming college setting. In 2014–2015, the GGC STARS Corps students and faculty hosted five Super Saturday events, reaching 181 middle school girls. The Super Saturday program represented GGC in the White House Near Peer Mentoring Challenge, part of First Lady Michelle Obama's Reach Higher Initiative, which aims to encourage all students to pursue education beyond a high school degree. The GGC STARS Corps entry (https://vimeo.com/120834578) was presented at the STARS Celebration and Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT) 2015 conferences as a way to encourage other STARS cohorts to connect their local outreach efforts to national and global initiatives.

Since 2012, GGC has engaged 44 college students to conduct outreach and service projects focused on tiered mentoring. Additionally, the GGC STARS Corps encourages and supports research, conference, and internship experiences. In the 2014–2015 academic year, three GGC STARS Corps students participated in research experiences for undergraduates (REUs), three received internships, and more than a dozen were financially supported to attend various research and student leadership conferences.

STARS provides opportunities and builds capacity for students to become more aware, committed, and involved in advanced computing-related extracurricular activities. For example, STARS evaluation assistants participate in a year-long distributed course in which they collect local STARS data and receive training in research methods. One GGC STARS evaluation assistant earned independent study credit for her qualitative study of female student perspectives in computing majors. This research experience was a direct result of this student's STARS evaluation assistant training and her access to STARS data.

Leadership Development through Outreach

Florida State University (FSU) is a public land-grant research university located in a state capital. The STARS Computing Corps at FSU operates as an institutionalized student leadership course, which provides fulfillment of education requirements for a leadership certificate available to undergraduate students. In addition, FSU STARS also operates as a registered student organization. With oversight by one faculty advisor and a university administrator, the student organization serves as the impetus for executing outreach projects, while the leadership course facilitates transferability of outreach and project management—students in the leadership course plan and drive the projects, with support from club members in carrying out activities. This tiered approach has enabled the FSU STARS Computing Corps to manage approximately 100 outreach events annually each year from 2010 to 2016, including coding camps and on-campus game design camps for middle school students and high school recruiting events across the region. FSU STARS has established a regional brand, with Corps students leading outreach and recruiting events in Orlando, Tampa, Jacksonville, and Pensacola.

The FSU STARS model uses a tiered approach to fulfill the mission of the STARS Computing Corps, with a core STARS leadership organization and a staggered launch of six partner subgroups that focus on various aspects of STARS and operate as independent student organizations. FSU STARS has reported that this approach has resulted in attracting a wider audience and generates more participation from the student body. An example of an FSU STARS subgroup is the Women in IT Sharing Experiences student organization, which focuses on mentoring and growing the female population in FSU computing degree programs. FSU credits this group with increasing the female representation in its IT courses from 18 to 45 percent of the student population. Another subgroup example is the Community Outreach Group, which focuses on holding computing-related community events, K–12 camps, and other civic engagement activities.

The STARS Computing Corps at FSU has been successful in building strong community and industry partnerships. Partnering with the Florida Technology Student Association, a group with more than 52,000 STEM-focused high school students, the FSU STARS Computing Corps received a new grant from the Florida Board of Governors to promote information technology, computer science, and computer engineering in the state. It now has relationships with 35 high schools, 6 community colleges, 2 universities, and 75 industry partners. The Florida IT Career Alliance (FITC) grant is built on the STARS model of outreach, and while there are only 12 STARS ambassadors at FSU, the FITC operates with 45 ambassadors at FSU and Florida A&M University, allowing for a wider reach to promote STEM.

One of the original STARS Alliance schools, FSU has had more than 90 students in its Corps since 2007. To date, the majority of the FSU STARS Computing Corps alumni have completed their college degree programs and are currently employed in computing in both industry and academia with companies that include Microsoft, Disney, Google, Cerner, Florida A&M University, and FSU.

Undergraduate Research

Morehouse College is an HBCU that provides liberal arts education to African-American males in the Atlanta, Georgia, area. At Morehouse, the STARS Computing Corps operates as an informal co-curricular experience where students meet regularly throughout the academic year with a faculty advisor to collaborate and develop their outreach roles. The primary activities for the Morehouse STARS Corps are peer mentoring and engagement in REUs. During the 2014–2015 academic year, Morehouse STARS research and outreach activities were culturally relevant—for example, working with a faculty mentor to advance research on the use of embodied conversational agents (also called avatars or virtual humans) to combat bullying in schools. Specifically, the STARS students developed the BullyShutdown Web application (www.bullyshutdown.com), which allows people to simulate the experiences of common bullying scenarios through conversations with avatars in an attempt to reduce the incidents of bullying in K–12 schools. Students, teachers, and administrators will embark on training each year via Bullyshutdown.com to change attitudes and behaviors, with emphasis on reducing bystander apathy.

The faculty Corps leader at Morehouse College, Kinnis Gosha, is a STARS Computing Corps alumnus, having participated in the Corps while a graduate student at Auburn University. Since the establishment of a STARS Computing Corps at Morehouse in 2012, Gosha has mentored 53 Corps students, 12 of whom have decided to pursue doctoral degrees in computing. This phenomenal achievement of recruiting young African-American men into graduate computing programs was featured on the HBCUbuzz.com website in June 2015 and on the HBCU Digest Website in August 2015.[13,14] It's particularly significant because only 93 African-American men participated in computing PhD programs in the US in 2014.[15]

At Morehouse College, STARS provides opportunities for students who might not otherwise be selected to work on undergraduate research or be selected for internship opportunities. Nationally, freshman STEM majors can be divided into three groups: those with a GPA of less than 2.0, who are unlikely to graduate despite substantial support efforts from their colleges; those with a GPA greater than 3.0, who graduate in high numbers and receive disproportionately high academic support; and freshmen who end the year with a 2.0 to 3.0 GPA, who make up almost half the total dropouts, and haven't traditionally been the target of support programs. According to Charlie Tyson, this 2.0–3.0 GPA group is at-risk and offers a high return-on-investment opportunity for colleges (https://www.insidehighered.com/news/2014/09/10/maximize-graduation-rates-colleges-should-focus-middle-range-students-research-shows). Emerging research suggests that targeting students with a 2.0 to 3.0 GPA could, in fact, provide the best strategy to increase institutional retention.[16] Many of the students selected at Morehouse College for participation in the STARS Computing Corps aren't typically considered to be high performers (most had GPAs between 2.7 and 3.2). The Morehouse STARS Computing Corps provides exposure for these often undersupported students to collaborate with others in computing, learn about emerging areas of computing, meet research faculty who can assist them in entering graduate computing programs, and talk to recruiters about internship and job opportunities by attending the national STARS Celebration as a cohort.

Study Method

The overarching research questions that helped guide our study are as follows:

- Does participation in student-led computing-related service projects as part of a national community have a positive impact on student participants?
- Do outcomes differ based on gender and ethnicity?
- Do outcomes differ based on role and time commitment to service projects?
- What factors correlate with outcomes?
- Does attending a national conference for broadening participation aid in forming a sense of community?

Here, we describe our approach to answering these questions by measuring the outcomes for STARS Computing Corps students.

Measures and Procedures

The STARS evaluation team developed a survey to measure the impact of participation in the Corps. The full postsurvey consisted of 47 demographic, Likert scales, and open response items that asked students to report how they perceived their participation in the Corps—how it impacted them personally, academically, and professionally.

Another set of items asked students to use a Likert scale (1 = strongly disagree to 6 = strongly agree) to rate specific aspects of their Corps project experience, which were classified as Corps process variables. These included student perceptions of the goals, as well as support for and success of their Corps projects. Demographic items asked students to indicate the type of Corps project they participated in, their class standing, their ethnicity, their gender, whether their Corps role was as a project leader or member, the amount of time spent weekly on their Corps project (more or less than three hours), and the total number of semesters they had participated in the Corps. Electronic surveys were administered to participants at the beginning of each academic year (presurveys) and again at the end (postsurveys). The average response rate across the five years of data collection was above 90 percent.

The STARS Celebration survey was developed by the STARS evaluation team to measure the impact of the annual STARS conference on students and faculty with questions relating to overall satisfaction, sense of belonging, and community. The current survey has been in use since 2012 and consists of a 23-item community scale, 14 items about conference experience, 2 gender and ethnicity demographic items, and 3 open-ended comment items. Rating scales are Likert-type, with a four-point scale for the community items and a six-point scale for the conference experience scale.

Participants

Although 943 total students completed the Corps postsurvey between 2009 and 2014, only undergraduate students (*N* = 811) who had participated in a Corps experience classified as outreach to K–12 (*N* = 326) or peer ambassadors (*N* = 106) were included in the current analysis. This excluded all graduate student participants (*N* = 132) and all undergraduates whose primary Corps projects were classified as research (*N* = 100), internship (*N* = 43), mentoring (*N* = 178), or other (*N* = 71). Of these 432 students, 262 were female and 228 were from underrepresented groups (African-American, Hispanic, American Indian or Alaskan Native, multiracial, or other).

Students who attend the STARS Celebration conference can participate in the postevent survey. The majority of event participants are students, with attendance ranging from 96 in one year to more than 240 in subsequent years. The evaluation team developed and deployed the survey in 2007, and revised the survey substantially in 2012. (Results from cumulative survey responses from 2012 through 2015 are discussed in the following section.) A total of 266 students have participated in the surveys in the past four years, 52 percent female, and 55 percent from underrepresented minority groups. Response rates have been between 25 and 40 percent for student responses. Surveys are distributed via email on the last day of the Celebration to all student attendees.

Results and Discussion

First, we calculated descriptive statistics for the subsample of interest (undergraduates classifying their primary Corps projects as K–12 outreach or peer ambassadors) for survey items asking students to rate their perceptions of the benefits of Corps participation. As Table 1 shows, students reported many positive effects of the Corps experience on a range of academic, personal, and professional outcomes.

We then calculated descriptive statistics for the survey items asking students to rate their perceptions of their Corps project process; Table 2 shows very favorable perceptions. Overall, students felt they had the support, time, and resources needed and acted in a way that enabled them to complete their projects successfully. Students also felt overwhelmingly positive about their Corps project experience, with 95 percent agreeing that their Corps project experience was meaningful, and 88 percent rating it as intellectually challenging. These findings support our first hypothesis and offer strong support that participation in a Corps outreach project to K–12 or peer students is perceived as a positive experience in terms of both process and outcome. This is consistent with previous research examining the impact of the STARS Corps model on participant attitudes, which found positive outcomes for self-efficacy, GPA, perceived social relevance of computing, and commitment to remain in computing disciplines.[17] These findings also corroborate with research that explores the benefits of other models of civic engagement and service learning[18,19] that have shown significant gains in attitudes pertaining to academics, civic engagement, and skill development. Additionally, research has shown that these gains are enhanced when the service component is intentionally and directly connected to the academic field,[18,19] as it is via the STARS Computing Corps model.

Next, we used *t*-test statistics to compare subgroups of the sample on each of the items reported above, starting with an exploration of possible differences in perceptions between male (*N* = 161) and female students (*N* = 262). As Table 3 shows, several significant differences ($p < 0.5$) were observed, with females rating several Corps outcome and process items more positively than male students. These findings are consistent with previous research showing that female STEM students respond more positively than males to team-based activities, especially those they perceive as socially relevant.[20]

Similar comparisons examined subgroup differences based on ethnicity, and significant differences in student perceptions were again observed. As Table 4 shows, students underrepresented in computing (*N* = 228) rated several Corps outcomes significantly more favorably than majority (White and Asian or Asian-American) students (*N* = 194). However, there were no significant differences in how the two groups perceived the Corps process. Again, these findings support the viability of the Corps model for engaging underrepresented students and extend previous work suggesting that civic engagement is a particularly effective pedagogy for underrepresented students.[10]

Next, *t*-test comparisons were made between students who served as leaders of their Corps projects (*N* = 138) and those serving as Corps project members (*N* = 294). Again, there were several Corps outcome and process ratings on which the two groups significantly differed, as Table 5 shows. These differences support the Corps peer leadership model and extend previous research about the positive effects of team-based projects on the development of important "soft" skills for STEM students.[21]

Table 1. Descriptive statistics for postsurvey items.

"The Corps …"	Percentage agree	Mean
Increased my collaboration with faculty.	93	5.1
Increased my work with peers.	94	5.2
Gave me more opportunities to work with people like me.	91	5.1
Provided me the chance to network professionally.	88	4.8
Allowed me to help others understand the value of computing.	94	5.2
Increased my interest in graduate education.	83	4.7
Increased my commitment to a computing major.	93	5.1
Allowed me to develop my computing skills and knowledge.	88	4.8
Made me feel more confident in my computing abilities.	88	4.9
Increased my awareness of career opportunities.	90	4.9
Made me feel more committed to a career in computing.	90	5.0
Made me feel more satisfied with my computing major.	93	5.1
Exceeded my expectations, with regard to its benefits.	86	4.8
Improved my leadership skills.	80	5.1
Was personally rewarding.	94	5.2
Helped improve my academic performance in computing.	76	4.4

*Note that "percentage agree" includes all respondents rating each item as slightly agree, moderately agree, or strongly agree.

Table 2. Descriptive statistics for postsurvey items.

Item	Percentage agree	Mean
My Corps project team was responsible.	96	5.3
The goals of my Corps project were met.	94	5.2
The goals of my Corps project were unrealistic.	26	2.7
I was able to commit the necessary time to my project.	91	4.9
My project was intellectually challenging and stimulating.	88	4.8
My project enhanced my passion about computing.	89	4.9
I believe my project was meaningful.	95	5.3
I had the resources needed to complete my project.	94	5.2
I had the support needed from my Corps faculty advisor to complete my project.	94	5.2

To explore the possible impact that time investment in the Corps project might have on both Corps outcome and process variables, we used *t*-tests to compare perceptions of students spending more than three hours per week on their projects ($N = 73$) and those spending three or fewer hours

Table 3. Gender differences in perceptions of Corps outcomes and process.

"The Corps …"	Males	Females	*p*
Increased my work with peers	5.0	5.3	0.02
Gave me more opportunities to work with people like me.	4.9	5.2	0.03
Exceeded my expectations, with regard to its benefits.	4.6	5.0	0.01
Improved my leadership skills.	4.9	5.3	<0.01
Was personally rewarding.	5.1	5.3	0.04
Was intellectually challenging and stimulating with regard to my project.	4.6	4.9	0.05
Gave me a project that I believe was meaningful.	5.1	5.4	0.02

Table 4. Differences in perceptions of Corps outcomes and process by ethnicity.

"The Corps …"	Majority	Underrepresented	*p*
Provided me the chance to network professionally.	4.6	5.0	<0.01
Increased my interest in graduate education.	4.5	4.9	<0.01
Increased my commitment to a computing major.	5.0	5.2	0.02
Increased my awareness of career opportunities.	4.7	5.0	0.01
Made me feel more committed to a career in computing.	4.8	5.1	0.02
Exceeded my expectations, with regard to its benefits.	4.7	5.0	0.02
Improved my leadership skills.	5.0	5.2	0.05
Helped improve my academic performance in computing.	4.2	4.5	0.04

(N = 359). As Table 6 shows, significant differences were observed, indicating that students benefited the most when they invested more time and effort into the project. It's important to note that all Corps projects require a semester-long time commitment, which exceeds the minimum recommendation of 25 hours for maximum benefits to accrue from a service learning experience,[22] but our findings suggest incremental gains as time investment increases beyond this minimum level. Overall, these results suggest that participation in the Corps increases academic performance as well as female and underrepresented minority students' sense of belonging, leadership skills, professional networking, and interest and commitment to computing.

Next, we used correlational analyses to explore possible relationships between perceived benefits and characteristics of the Corps experience. As Table 7 shows, significant correlations indicate that characteristics of the Corps project process were associated with perceived benefits. Among the most important characteristics were having faculty support for the project and having adequate resources and time to complete the project. A less important but still significant characteristic was having realistic project goals.

We did a final correlational analysis to examine the relationship between the number of semesters spent in the Corps (ranging from 1 to 7, with a mode of 2) and the perceived benefits of the Corps experience. Modest but significant correlations between semesters in the Corps and the following outcomes were observed: improved academic performance (r = .11, p = .01), improved leadership skills (r = .15, p = .003), and benefits exceeded expectations (r = .10, p = .03). No other outcomes were significantly correlated with the number of semesters a student spent in the Corps, but the lack of significance could be linked to range restriction (70 percent of the sample had one or two semesters in the Corps).

Computing in SCIENCE & ENGINEERING | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page

Table 5. Differences based on project leadership.

"The Corps …"	Members	Leaders	*p*
Increased my work with peers.	5 .3	5.0	0.02
Gave me more opportunities to work with people like me.	5.2	4.9	0.03
Exceeded my expectations, with regard to its benefits.	4.6	5.0	0.01
Improved my leadership skills.	4.9	5.3	<0.01
Was personally rewarding.	5.1	5.3	0.04
Gave me a project that was intellectually challenging and stimulating.	4.6	4.9	0.05
Gave me a project that I believe was meaningful.	5.1	5.4	0.02

Table 6. Differences in perceptions of Corps outcomes and process by time investment.

"The Corps …"	>3 hours	<3 hours	*p*
Increased my work with my peers.	5.4	5.2	0.01
Gave me more opportunities to work with people like me.	5.5	5.1	<0.01
Increased my commitment to a computing major.	5.3	5.0	0.01
Made me feel more committed to a career in computing.	5.2	4.9	0.03
Exceeded my expectations, with regard to its benefits.	5.2	4.7	<0.01
Improved my leadership skills.	5.4	5.1	<0.01
Was personally rewarding.	5.5	5.2	0.01
Helped improve my academic performance in computing.	4.8	4.3	<0.01
Gave me a project that was intellectually challenging and stimulating.	5.0	4.7	0.05
Gave me a project that enhanced my passion about computing.	5.1	4.8	0.02
Gave me a project that I believe was meaningful.	5.4	5.2	0.05

Celebration survey responses were first combined across 2012–2015, the years in which the items are consistent, to calculate comprehensive descriptive statistics for survey items asking students to rate their perceptions of the benefits of Celebration participation. Of the 266 students who completed the survey, 52 percent were female, and 55 percent were from underrepresented minority groups. As shown in Table 8, students reported an overall positive experience in attending the conference, stating that they were able to connect with others like themselves, learn about research, and understand the social relevance of computing.

The sense of community scale shows a moderate to strong community of practice and sense of belonging. Table 9 presents the means and standard deviations of the items pertaining to community on a four-point scale. Students rated their bond and attachment to the STARS community positively.

Next, we performed a gender comparison on each of the Celebration survey items reported earlier. *T*-test statistics were calculated to investigate possible differences between male ($N = 114$) and female ($N = 131$) students. Table 10's results indicate significant differences ($p < .05$) between males and females on sense of community, but no significant gender differences in overall conference experience or in how much men and women identify as being part of the STARS community. Females rated enjoyment and commitment to the Celebration community higher

Table 7. Correlations between perceived benefits and characteristics of Corps experience.

"The Corps …"	Support	Resources	Time	Realistic goals
Improved my leadership skills.	.54 (p = .001)	.57 (p = .001)	.49 (p = .001)	.08 (p = .066)
Was personally rewarding.	.53 (p = .001)	.56 (p = .001)	.45 (p = .001)	.09 (p = .067)
Improved my academic performance.	.44 (p = .001)	.42 (p = .001)	.40 (p = .001)	.23 (p = .001)
Enhanced my passion for computing.	.65 (p = .000)	.60 (p = .000)	.59 (p = .001)	.22 (p = .001)
Gave me a project that I believe was meaningful.	.72 (p = < .001)	.70 (p = < .001)	.60 (p = < .001)	.10 (p = .029)
Exceeded my expectations, with regard to its benefits.	.54 (p = .001)	.57 (p = .001)	.44 (p = .001)	.19 (p = .001)
Met project goals.	.63 (p = < .001)	.59 (p = .001)	.55 (p = .001)	.70 (p = < .001)

Table 8. Celebration satisfaction items (six-point Likert-type scale).

"At the Celebration …"	Percentage agree	Mean
I learned about current computing research.	90	4.9
I learned about undergraduate research opportunities.	82	4.7
I learned about graduate school programs and opportunities.	82	4.7
I learned about how to fund graduate school.	71	4.2
I was able to connect with peers.	94	5.2
I was able to connect with faculty.	88	4.8
I was provided with adequate opportunity to connect with students like me.	92	5.0
I was provided with adequate opportunity to learn about graduate school.	78	4.5
I was provided with adequate opportunity to learn about computing research.	90	5.0
It was emphasized how computing and IT professionals can use their skills to improve the collective quality of life in society.	95	5.2
I received training that discussed the disparity in representation of women in computing careers.	91	5.0
I received training that discussed the disparity in representation of people of color in computing careers.	89	4.9
I received training that discussed the disparity in representation of people with disabilities in computing careers.	79	4.5
Overall, I felt welcomed.	96	5.4

than males, yet rated trust and identification with community symbols lower than males. An interesting observation is that while women report higher ratings of their own influence in the community, they also report lower ratings of the community's ability to influence, an indication that women feel empowered in the STARS community but recognize the challenges STARS faces in making a global

Computing in Science & Engineering | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page | Qmags THE WORLD'S NEWSSTAND®

Table 9. Celebration community index items (four-point Likert-type scale).

Item	Percentage agree	Mean
I expect to be with this community until I graduate.	92	3.66
Being a member of this community makes me feel good.	91	3.56
I have influence over what this community is like.	61	2.89
It is very important to me to be a part of this community.	82	3.35
I am with other community members a lot and enjoy being with them.	73	3.16
I get important needs of mine met because I am part of this community.	73	3.03
This community has been successful in getting the needs of its members met.	82	3.09
When I have a problem, I can talk about it with members of this community.	77	3.14
I can trust people in this community.	83	3.26
I can recognize most of the members of this community.	68	2.99
This community has symbols and expressions of membership such as clothes, signs, art, architecture, logos, landmarks, and flags that people can recognize.	70	2.87
I put a lot of time and effort into being part of this community.	74	3.09
Fitting into this community is important to me.	75	3.05
This community can influence other communities.	91	3.40
I care about what other community members think of me.	73	3.01
Members of this community have shared important events together, such as holidays, celebrations, or disasters.	59	2.75
Community members and I value the same things.	86	3.23
People in this community have similar needs, priorities, and goals.	88	3.36
Most community members know me.	53	2.72
If there is a problem in this community, members can get it solved.	83	3.22
This community has good leaders.	91	3.43
Being a member of this community is part of my identity	63	3.48

impact. Because 52 percent of women responding to the Celebration surveys also reported minority ethnicity status, we believe these gender differences provide support for the complications inherent in being both female and from a minority ethnic group; this "double bind" has also resulted in women of color struggling for recognition and identification within STEM discipline communities,[23] further suggesting the need to consider the intersectionality of identities (such as female and minority) when building a community to broaden participation in computing.

Student comments about Celebration participation reflect an overwhelming sense of community and benefit. Emergent themes in open-ended comments demonstrate a strong peer bond formed at the conferences, and students continue to report the primary benefit as meeting friends and sharing ideas. As one student noted, "[The Celebration] connects us and allows us to learn and grow in a tangible way." Similarly, others have noted that the Celebration "fostered friendship and understanding" and provided an opportunity to "meet students

Table 10. Gender difference in community index (four-point Likert-type scale). All rows are significantly different by gender, except those that are italicized.

Item	Males	Females	*p*
I expect to be with this community until I graduate.	3.54	3.66	0.000
Being a member of this community makes me feel good.	3.45	3.56	0.000
I have influence over what this community is like.	2.77	2.89	0.000
It is very important to me to be a part of this community.	3.25	3.35	0.001
I am with other community members a lot and enjoy being with them.	3.04	3.16	0.004
I get important needs of mine met because I am part of this community.	3.04	3.03	0.000
This community has been successful in getting the needs of its members met.	3.23	3.09	0.003
When I have a problem, I can talk about it with members of this community.	3.18	3.14	0.010
I can trust people in this community.	3.37	3.26	0.014
I can recognize most of the members of this community.	3.03	2.99	0.023
This community has symbols and expressions of membership such as clothes, signs, art, architecture, logos, landmarks, and flags that people can recognize.	3.14	2.87	0.048
I put a lot of time and effort into being part of this community.	3.15	3.09	0.015
Fitting into this community is important to me.	3.15	3.05	0.000
This community can influence other communities.	3.47	3.40	0.000
I care about what other community members think of me.	3.05	3.01	0.000
Members of this community have shared important events together, such as holidays, celebrations, or disasters.	2.89	2.75	0.006
Community members and I value the same things.	*3.19*	*3.23*	*0.129*
People in this community have similar needs, priorities, and goals.	*3.33*	*3.36*	*0.208*
Most community members know me.	*2.67*	*2.72*	*0.059*
If there is a problem in this community, members can get it solved.	*3.30*	*3.22*	*0.305*
This community has good leaders.	3.52	3.43	0.058
Being a member of this community is part of my identity.	*2.86*	*2.89*	*0.476*

and faculty that understand the issues I face and providing [*sic*] possible solutions."

Another prevalent theme from student comments is feeling inspired by the Celebration and the community. One student noted, "I liked the speeches because they were inspirational." Another commented, "Members encourage each other and share information to support each other."

The STARS Computing Corps model follows the central tenets of quality service learning engagement, especially those that have demonstrated greater gains in student outcomes such as providing discipline-specific projects with formalized training, reflection, and discussion with peers and faculty.[24] Participation in STARS also emphasizes student membership in regional cohorts and a national community. Our study suggests that the use of student-led service learning and civic engagement projects as part of a national community can help achieve educational equity and potentially support the long-term goal to increase diversity in computing-related disciplines.

Our findings offer empirical support for the STARS model of engagement, a model well-grounded in theory and research, with flexible elements that are easily adaptable by others seeking to enhance student outcomes, particularly students who are traditionally underrepresented in computing. However, further work is needed to address some of the limitations in this initial study. To address the issue of self-report bias and lack of comparison group, the STARS research team plans to embark on a comparison of student outcomes at the local and national levels. Selected STARS institutions will be equipped to conduct institutional-level comparisons of student outcomes with propensity matching. STARS will continue comparing graduation outcomes to national samples. ■

Acknowledgments

This material is based on work supported by the US National Science Foundation under grant number CNS-1042468. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. J.S. Brown and P. Duguid, "Organizational Learning and Communities-of-Practice: Toward a Unified View of Working, Learning, and Innovation," *Organization Science*, vol. 2, no. 1, 1991, pp. 40–57.
2. F.A. Hrabowski III and K.I. Maton, "Enhancing the Success of African-American Students in the Sciences: Freshman Year Outcomes," *School Science and Mathematics*, vol. 95, no. 1, 1995, pp. 19–27.
3. J.M. Cohoon, Women and Information Technology: Research on Underrepresentation, MIT Press, 2005.
4. M. Hezemans and M. Ritzen, "Learning Environments and Responsibility: Three Types of Learning Environments," *Proc. IFIP 17th World Computer Congress*, 2002, pp. 19–27.
5. J. Blum and L. Frieze, "In a More Balanced Computer Science Environment, Similarity Is the Difference and Computer Science Is the Winner," *Computing Research News*, 17 May 2005; http://archive.cra.org/CRN/articles/may05/blum.frieze.html.
6. N. Thomas, "The Effects of an Ethnic-Based Mentoring Model on College Adjustment, Grade Point Average, and Retention among First Year African American College Students Attending a Predominantly White Institution," PhD dissertation, Dept. Psychology, Michigan State University, 2005.
7. R. Sellers, T. Chavous, and D. Cooke, "Racial Ideology and Racial Centrality as Predictors of African American College Students' Academic Performance," *J. Black Psychology*, vol. 24, no. 1, 1998, pp. 8–27.
8. J. Margolis et al., "Beyond Access: Broadening Participation in High School Computer Science," *ACM Inroads*, vol. 13, no. 4, 2012, pp. 72–78.
9. A.W. Astin et al., "How Service Learning Affects Students," Higher Education Research Inst., Univ. California, Los Angeles, 2000.
10. E.J. Coyle, L.H. Jamieson, and W.C. Oakes, "EPICS: Engineering Projects in Community Service," *Int'l J. Eng. Education*, vol. 21, no. 1 2005, 139–150.
11. G. Madey, C. Friedland, and P. Brenner, "A Service Learning Program for CSE Students," *Proc. ASEE/IEEE Frontiers in Education*, 2005, pp. F2F1–F2F6.
12. P. Sanderson and K. Vollmar, "A Primer for Applying Service Learning to Computer Science," *Proc. SIGCSE Tech. Symp. Computer Science Education*, 2000, pp. 222–226.
13. J. Carter Jr., "HBCU Digest Podcast Series: Morehouse Professor Kinnis Gosha," *HBCU Digest*, 25 Aug. 2015; http://thehbcudigest.tumblr.com/post/126757893466/hbcu-digest-podcast-series-morehouse-professor.
14. C. Tyson, "The 'Murky Middle,'" Inside Higher Education, 2014; https://www.insidehighered.com/news/2014/09/10/maximize-graduation-rates-colleges-should-focus-middle-range-students-research-shows
15. S. Zweben and B. Bizot, "2014 Taulbee Survey: Relentless Growth in Undergraduate CS Enrollment; Doctoral Degree Production Remains Strong, but No New Record," *Computing Research News*, vol. 27, no. 5, 2015, pp. 2–51.
16. E. Venit, "What Can We Learn from First Year GPA?," Education Advisory Board, 2014; https://www.eab.com/technology/student-success-collaborative/student-success-insights/2014/04/what-can-we-learn-from-first-year-gpa.
17. T. Dahlberg et al., "The STARS Alliance: Viable Strategies for Broadening Participation in Computing," *Trans. Computer Education*, vol. 11, no. 3, 2011, pp. 1–25.
18. C.I. Celio, J. Durlak, and A. Dymnicki, "A Meta-analysis of the Impact of Service-Learning on

Students," *J. Experiential Education*, vol. 34, no. 2, 2011, pp. 164–181.

19. G.D. Kuh and C.G. Schneider, "High-Impact Educational Practices: What They Are, Who Has Access to Them, and Why They Matter," Assoc. Am. Colleges and Universities, 2008.
20. M. Buckley et al., "Benefits of Using Socially-Relevant Projects in Computer Science and Engineering Education," *Proc. SIGCSE*, 2004, pp. 482–486.
21. M.A. Holliday and D.R. Luginbuhl, "Peer-Centered Service Learning," *Proc. ASEE/IEEE Frontiers in Education*, 2004, pp. F3E-1–F3E-6J.
22. B. Jacoby, *Service-Learning Essentials: Questions, Answers, and Lessons Learned*, Jossey-Bass, 2015.
23. M. Ong, L.L. Espinosa, and G. Orfield, "Inside the Double Bind: A Synthesis of Empirical Research on Undergraduate and Graduate Women of Color in Science, Technology, Engineering, and Mathematics," *Harvard Educational Rev.*, vol. 81, no. 2, 2011, pp. 172–208.
24. J. Payton et al., "The Effects of Integrating Service Learning into Computer Science: An Inter-institutional Longitudinal Study," *Computer Science Education*, vol. 25, no. 3, 2015, pp. 311–324.

Jamie Payton is an associate professor of computer science at the University of North Carolina at Charlotte. Her research interests include pervasive computing, smart and connected health, and broadening participation in computing. Payton received a D.Sc. in computer science from Washington University in St. Louis. Contact her at payton@uncc.edu.

Tiffany Barnes is an associate professor at North Carolina State University. Her research interests include artificial intelligence in education, broadening participation, and serious games. Barnes received a PhD in computer science from North Carolina State University. Contact her at tiffany.barnes@gmail.com.

Kim Buch is a professor at the University of North Carolina at Charlotte. Her research interests include work redesign, participative management, and scholarship of teaching and learning. Buch received a PhD in psychology from Iowa State University. Contact her at kkbuch@uncc.edu.

Audrey Rorrer is a research associate at the University of North Carolina at Charlotte. Her research interests include culturally responsive computing education, evaluation capacity development, and assessment. Rorrer received a PhD in counselor education from the University of North Carolina at Charlotte. Contact her at Audrey.rorrer@uncc.edu.

Huifang Zuo is a doctoral candidate studying educational research methodology at the University of North Carolina at Charlotte. Her research interests include neighborhood analysis in education, self-efficacy, and educational evaluation. Zuo received an MS in geography information sciences from the University of North Carolina at Charlotte. Contact her at hzuo@uncc.edu.

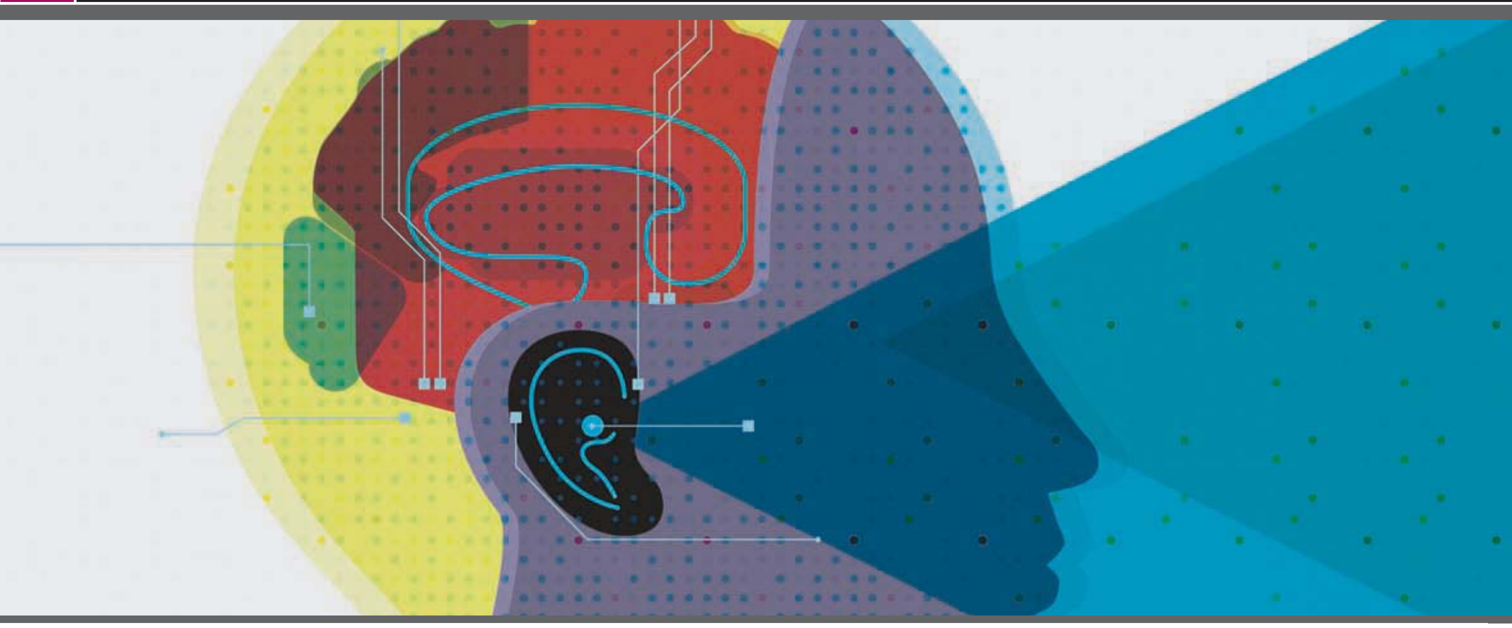
Kinnis Gosha is an assistant professor at Morehouse College. His research interests include conversational agents, broadening participation, and virtual mentoring. Gosha received a PhD in human-centered computing from Clemson University. Contact him at kinnisgosha@gmail.com.

Kristine Nagel is an associate professor of information technology at Georgia Gwinnett College. Her research interests include ubiquitous computing, technology outreach, and computing education. Nagel received a PhD in computer science from Georgia Institute of Technology. Contact her at knagel@ggc.edu.

Nannette Napier is an associate professor of management information systems and decision sciences at Georgia Gwinnett College. Her research interests include software development, project management, and requirements engineering. Napier received a PhD in computer information systems from Georgia State University. Contact her at nnapier@ggc.edu.

Ebrahim Randeree is an associate dean at Florida State University. His research interests include technology adoption, health informatics, and educational management. Randeree received an MBA from the University at Buffalo in MIS/health administration. Contact him at eranderee@cci.fsu.edu.

Lawrence Dennis is dean of the College of Communication & Information at Florida State University. His research interests are in experimental nuclear physics, scientific computing, online learning, information technology, and IT education and diversity. Dennis received a PhD in nuclear physics from the University of Virginia. Contact him at larry.dennis@cci.fsu.edu.



Enacting Agency:

The Strategies of Women of Color in Computing

Apriel K. Hodari | Eureka Scientific Inc.

Maria Ong, Lily T. Ko, and Janet M. Smith | Technical Education Research Centers Inc.

Research on marginalized groups in STEM fields commonly overlooks those who persist and succeed, characterizing groups such as women of color as passive victims instead of active agents in their own achievements. Focusing on women of color who are successfully staying in STEM helps us focus on the ways they enact that agency.

It's in the national interest to ensure that women who identify as Asian-American, Black, Latina, Native American, and mixed race/ethnicity—women of color—are well represented in science, technology, engineering, and mathematics (STEM), especially in computing. As US colleges already enroll a majority of women (57 percent) and an increasing population of minorities (44 percent), women of color live at the intersection of these growing populations (www.nsf.gov/statistics/srvyipeds/#overview). Yet, the promise of this growth remains unrealized, particularly in computer science. According to Edie Fraser, director of STEMconnector.org, 71 percent of new jobs will be related to computing (www.usnews.com/news/articles/2013/11/13/behind-americas-decline-in-math-science-and-technology); yet, in 2012, women of color earned fewer than 7 percent of BS degrees in computer sciences (www.nsf.gov/statistics/wmpd). In fact, the past decade has witnessed a general decline in the proportion of computer sciences degrees awarded to all women, especially to those who belong to underrepresented minority groups (Black, Latina, and Native American). Research shows that attrition of women of color from predominantly White, male STEM contexts such as computer science departments can be largely attributed to the feeling of isolation they experience.[1] However, research has also suggested that through *agentic*

 1521-9615/16/$33.00 © 2016 IEEE Copublished by the IEEE CS and the AIP

computing in SCIENCE & ENGINEERING | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page | Qmags THE WORLD'S NEWSSTAND®

The institutional culture of many STEM disciplines presents formidable barriers for women of color that can significantly contribute to their sense of isolation.

strategies, in which women of color take direct action via their own agency, some women of color have successfully demonstrated resilience in the face of institutional and social barriers.[2,3]

Previous studies have demonstrated how policy and practice that promote the success of underrepresented groups in STEM also promote the success of all STEM participants.[4,5] For example, Christianne Corbett and Catherine Hill point to Romila Singh and Nadya Fouad's study, which found that the primary difference between women who persisted in engineering and computing careers and those who did not was the workplace environment.[4] In fact, research on men in these fields showed that the same factors ("excessive and ill-defined work goals, and various kinds of incivility") motivated men to leave as well. Thus, Corbett and Hill concluded, improving the workplace environment for women could make the experience better for everyone. Similarly, our work—examining how individual and institutional strategies affect the retention and success of one of the most marginalized groups in computing, women of color—has the potential to increase awareness of what supportive, productive, and fully inclusive environments entail.

Conceptual Framework

The challenges facing women of color in STEM were first brought to national attention in *The Double Bind: The Problem of Being a Minority Woman in Science*.[6] Since that seminal report, research into the types of barriers women of color face pursuing STEM training and careers have pointed to environments in which the women felt isolated, engendered by factors such as "challenging interpersonal relationships with professors, supervisors, and peers, low expectations from others, lack of mentoring and support," which threaten their persistence and advancement in STEM disciplines.[2] Indeed, studies also found that many women of color in STEM education experienced feeling unwelcome and being treated differently from peers,[1] and in some cases were the only women of color in their departments.[3] In formal and informal settings in STEM departments at universities, women of color often faced subtle offenses (micro-aggressions) that were racist and/or sexist.[7,8] Additionally, sites from which women of color felt isolated or excluded included classrooms, laboratories, and informal after-work interactions where discussions about their field of study occurred.[2,9] In short, many women of color identified isolation as a common and significant barrier to their advancement in STEM fields.

The institutional culture of many STEM disciplines presents formidable barriers for women of color that can significantly contribute to their sense of isolation. Historically dominated by White males, STEM disciplines reflect at an institutional level an undervaluing of femaleness and non-Whiteness.[10] Indeed, the intersectionality[11] (the simultaneous experiencing) of gender and race/ethnicity is experienced by women of color in certain STEM disciplines as a double bind,[6] magnifying barriers and impeding their efforts and successes in ways not experienced by their White female and all male peers.[11] As a result, to persist in STEM fields, women of color must work to unravel the double bind, expending "time and energy that could have been spent doing science."[2]

The role of institutional culture in discouraging women of color in STEM can be obscured by discussions of the "leaky pipeline."[12] This metaphor describing the attrition of women of color during their pursuit of STEM education and careers places blame on the women themselves rather than focusing on specific practices of institutional culture that can affect their persistence or attrition. Moreover, the notion of a leaky pipeline can falsely depict women of color as being passive rather than demonstrating agency and making choices as they traverse their paths through STEM.[3,13]

In spite of the many barriers women of color face, many "persist and thrive in STEM."[2] Indeed, although the agency of women of color in STEM is a largely unexplored area, a few studies have documented ways in which they've actively developed and demonstrated strategies by which to cope with and navigate STEM environments.[14–16] For example, when faced with barriers in their STEM discipline, some women of color focused instead on identifying themselves as what Heidi Carlone and Angela Johnson call "altruistic scientists," women who use science to serve humanity. Laying claim to and prioritizing an identity that mattered to these

What agentic strategies do women of color use to persist and succeed in computing education and careers?

women allowed them to cope with and redefine the unsupportive, marginalizing identities that others in their discipline created for them.[16]

Carlone and Johnson[16] and Johnson and colleagues[3] have described the processes women of color use to create science identities in the face of others' projections. However, the success of authoring a science identity as a strategy for persisting in STEM careers depends on recognition of that identity by the people in power.[3] A strong science identity was found to be especially important for women of color who were students majoring in STEM fields. Their persistence in their studies was attributed to several agentic strategies, including exploring course content with their peers, being part of research activities, and participating in student organizations, all of which were critical to their STEM success.[9]

There presently exists scant empirical literature about women of color in computer science,[17] and even less exists about how they employ agentic strategies to persist and succeed in their chosen field. To help fill this gap, we present firsthand accounts of navigational strategies enacted by women of color who are succeeding and persisting in computing education and careers.

Data and Methods

Our data come from a US National Science Foundation-funded research study, "Computing beyond the Double Bind: Women of Color in Computing Education and Careers" (CBDB). This empirical research study has goals of advancing knowledge about women of color in computing by analyzing their navigation strategies through higher education and professional careers. CBDB has additional goals of studying programs that have successfully served women of color in computing and of creating a mentoring intervention for young women of color in computing based on our findings.

In this article, we focus on a single question: What agentic strategies do women of color use to persist and succeed in computing education and careers? Data for this article are a subset from the larger study described above; data sources include case study data and interviews that, together, help reflect the lived experiences of successful women of color in computing education and careers.

Data Sources

We gathered data on 17 women of color in computing through two primary sources: case study data and interviews.

Our analysis includes case study data from one national student support community, Coalition (a pseudonym). This community serves a diverse population (by race, gender, class, immigrant status, physical ability, and family educational background). Participants, including students and professionals in computing, come from many different educational and geographic settings. The programs, events, and other opportunities provided within this community promote computing excellence for students and increase participation of underrepresented students in cutting-edge computing organizations. We conducted a three-day site visit of a Coalition event in 2013. Our data collection methodology included ethnographic field notes in various contexts (community spaces, meetings, conferences, classrooms, and collaborative spaces). For this article, the ethnographic field notes helped contextualize the lived experiences of some of the women of color in our study.

We also interviewed Coalition members, gathering a limited but in-depth set of contemporary, discipline-specific experiences and viewpoints (see Table 1). Eleven interviews were conducted in person during the Coalition event described above, with six more handled later by phone. Interviews were conducted with eight Coalition students in higher education and three Coalition-affiliated faculty or other professionals. All interview participants self-identified as women of color. Additionally, we later selected five other non-Coalition women of color professionals in computing to interview. To achieve a wide range of perspectives in the professional realm, we selected the five additional participants by using a nonrepresentative stratified sampling technique[18] from a pool of candidates who were recommended to us through broad professional networks and via email solicitations to professional organizations serving women or minorities in STEM. (The study is ongoing; more interviews are planned.) All interviews were semistructured, open-ended, and between one to two hours in duration. Participants were asked a broad set of questions about their academic background and interests, a challenging time and how they overcame it, factors that promote their success, mentors and role models, and advice for

Table 1. Interview participants' characteristics.

Pseudonym	Career level	Race/ethnicity	Coalition affiliation?
Francesca	Undergraduate	Asian American	Yes
Sarah	Undergraduate	Black	Yes
Miranda	Undergraduate	Latina	Yes
Kathy	Undergraduate	Latina	Yes
Corinda	PhD student	Black	Yes
Sadie	PhD student	Native American	Yes
Jade	PhD student	Black	Yes
Daria	PhD student	Arab	Yes
Serena	Professional	Asian American	No
Hasina	Professional	Black	Yes
Christy	Professional	Arab	Yes
Georgette	Professional	Latina	Yes
Josie	Professional	Black	No
Julia	Professional	Latina	No
Karina	Professional	Asian American	No
Lucy	Professional	Black	No
Sierra	Professional	Black	No

institutions and individuals like themselves. Coalition-affiliated interviewees were also asked a set of questions regarding their experiences as part of the Coalition community. All interviews were audiorecorded and transcribed for coding and analysis.

Data Analysis

To analyze the narratives collected during the interviews, we employed narrative analysis. Catherine Riessman[19] states that narrative analysis, a method of seeing beyond individual life stories to broader patterns, is especially effective for understanding the experiences of those who are traditionally marginalized. For our study, narrative analysis required transparent processes of laying out stories, identifying codes, entering codes into a matrix, and then inductively creating conceptual groupings and orderings from the data. Building categories of themes from the bottom up,[20] we coded and discussed the data until we established a comprehensive set of themes. The data coding process included individual coding, paired consensus, and routine team calibrations to ensure that all members were applying established and emergent codes in similar manners. We then individually assigned a set of related codes to detect emerging themes. Team members exchanged their work with one another, and everyone then analyzed the themes to confirm or reconstruct the themes. By combining these systematic methods and utilizing constellations of multiple researchers, the project team created a form of triangulation to develop an understanding of the experiences of women of color.

Positionality and Limitations

To help clarify any bias that might influence our interpretation of the study results, we describe our own positions here. We're all women. Two of us have been researchers of, and advocates for, women of color in STEM for more than 10 years. We represent multiple races (Asian-American, Black, and White) and career stages (early and mid-career). None of us hold computing degrees, although one of us, a Black woman, has received advanced degrees in science and engineering; another, an Asian-American woman, currently serves as a social

We seek to identify policies and practices enacted by institutions and other individuals that could promote the persistence and success of women of color in computing.

science advisor to the National Center for Women and Information Technology; and another, a White woman, worked as an instructional designer for a large multinational computer hardware and software corporation. Inevitably, our backgrounds influence how and why we share participants' stories and our interpretation of these stories. We recognize our positionality and strive for trustworthiness with a goal of presenting the data to authentically represent the participants' lived experiences.

Note that the racial demographics in our study are not meant to reflect the current representation in US computer science. We aimed to have a purposive sample that included women of various races and ethnicities, as well as women in differing career stages. Our sample only included one Native American woman, in part because of the limitations of our team's recruiting abilities, and in part because there currently exists a dearth of Native American women in computing.[1] We will increase our recruiting efforts as the project progresses. Individuals and programs self-selected into the study and could differ from those who didn't volunteer to participate.

Findings

This article focuses on the navigational strategies that 17 women of color in our study took to advance in computing. They engaged these strategies in the context of isolating science environments such as those identified in other literature and earlier findings from our study.[1–3,7–9] In their narratives, many conveyed their sense of isolation relative to and entwined with the strategies they used to change it. For example, Serena (not her real name; we use pseudonyms throughout), an Asian-American postdoctoral scholar in computing education research, provided a vivid illustration of the abilities of women of color to rise above their isolation and succeed where so many around them do not[21]:

> In my computer science class, a lot of the projects were group [*work*], and so I found two ... [*minority*] groupmates, who were heaven-sent. And we stuck by each other and actually, after we found each other, planned all of our schedules in sync with each other, so we took the same classes in order to get through the undergraduate experience together. Because a part of being a minority is that people don't want to work with you. They don't look at you and sense that you are a smart person they want to work with. So finding people who believe in you and you believe in, and then sticking together, was really important.

Although we focus here on women of color's own agentic strategies, we seek to identify policies and practices enacted by institutions and other individuals that could promote the persistence and success of women of color in computing. Among the strategies that women of color employed to successfully navigate and persist in computing education and careers, some—confronting challenges and developing and using "soft" skills—were a response to the isolation the women felt in their learning and work environments. Alternatively, others looked within and pulled strength and strategies from the unique experiences of their own lives to persist in computing.

Being Motivated by Challenges

The women who participated in our study described strategies they employed to confront challenges to their merits, to their comfort (in a field with few gender or racial/ethnic peers), and to their progress (because of unhelpful teachers). Miranda, a Latina undergraduate computing student, described how she used encounters with peers who questioned her merits in racist and sexist ways to motivate her to "do better":

> I was kind of being accused of having my merits be because of my ethnicity and my gender. Really I don't let that stop me … but it does encourage me to do better for fear of not being competent compared to everybody else. It's a bad feeling but at the same time it yields good results.

Christy, an Arab professional in computing, and someone who admittedly gravitates toward things that make her nervous, used the challenge of small numbers of women in computing to persist in the field:

> So that was sort of like, "Okay, there are not many females in there, so I like it." So like I said, anything

Other women of color in computing drew from their personal histories and unique experiences to help them shape the route they took in computing.

> that makes me nervous, go for that: … "Not many females in this classroom. … I like the challenge."

While Miranda and Christy used challenges to motivate their own performance, Francesca, an Asian-American undergraduate student in computer science, was inspired to provide better instruction to others than she had received:

> I'm not saying I had horrible teachers back then, but [*I had*] teachers that weren't really helpful. [*So*] I wanted to be the type of instructor maybe to help those who were struggling or who think that they can't really do anything right, those kinds of stereotypes with students.

Georgette, a Latina professional in computing, reflected on the nature of failure in all of our lives and how she was able to take inspiration from stories of others' struggles to persist toward her own goals and keep the pressures of earning tenure in perspective, relative to her own definition of success:

> Everyone has run into failure. Everyone has had challenges that they have overcome and so when we learn more about the challenges that people have overcome, that inspires us because it's just like, "Well, if they did it, I can do it." [*When*] I was graduating with my PhD from [*institutional name*], I took … a very senior, incredibly well-respected member of the National Academy professor to lunch, and he told me, "You know, I didn't make tenure." I was like, "What? That person did not make tenure?" He was very quiet about it, and he didn't want to talk about it. He was kind of embarrassed about it. … It's important to know that you can have an amazing impact. And whether some people like you or don't or think you've done enough to warrant tenure or not, it has nothing to do with whether or not you'll be successful. But when this professor told me that, it gave me such courage because I thought, "If he didn't make tenure and he's been as successful as he can, then if I don't make tenure, I'm going to be okay. And if I don't make tenure, maybe it's because that wasn't the right place for me and the good Lord above wanted me to go elsewhere." And that was wonderful because I was like, "It's okay to fail." So that's really empowering.

In each of these cases, the women described situations or conditions that challenged their success, yet they demonstrated agency in confronting those challenges and turning them around to motivate success. Indeed, these strategies were powerful motivators in these women's successful navigations of their computing education and career experiences.

Drawing on Unique Experiences

Other women of color in computing drew from their personal histories and unique experiences to help them shape the route they took in computing. They applied insights from those experiences—some specifically about being a woman of color—to their research and practice of computing. Hasina, a Black postdoctoral scholar in computer engineering, described how her childhood experience motivated her choice to focus on energy consumption in low-income neighborhoods for her dissertation research:

> I don't know if it's my race that influenced my research or knowledge of injustice because of social economic status. My dad was really rich when I was younger and I didn't … really know the value of money. But real estate is one of those things that is cyclical. So when I was old enough to understand the value of money, he wasn't rich anymore. He was struggling. And I guess I realized how not having money could hold you back. And at that point, it wasn't about race. It was about access and I believe a part of that experience led me to looking at low income [*populations*]. I didn't say, "Well, [*I'll study*] energy consumption in African-American neighborhoods." I said, "[*I'll study*] energy consumption in low-income households." So I think it was social economic status and people not having a level playing field when it really came down to it. It wasn't about race anymore. I mean, if you look at the numbers, yeah…many African-Americans are

Computing in Science & Engineering | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page | Qmags THE WORLD'S NEWSSTAND®

> low income, but sometimes it's really about … money. … I think that drives my research in ways that I may not even be aware of.

Hasina went on to talk extensively about not only the computing tools she developed in her research, but what she learned about community cohesion and how to make technical tools work in communities not usually addressed in mainstream conversations about energy conservation.

In another example, Jade, a Black doctoral student in computing, describes how she based her dissertation project on both her origins in an urban Black community and her observations of her younger cousin's struggles to learn to read. In a very personal way, Jade has integrated her experiences of being a woman of color with seeking solutions to issues of esteem for girls of color, which she suggests inhibit their participation in computing education and careers. She was confident that insights from her own experiences provided a perspective on Black girls that a White man, for example, would not have. Jade suggested that her experiential perspective would lead to more effective technology-related solutions to self-esteem issues of Black girls:

> I want my dissertation work to be in the area of reading. I have a younger cousin who is in the fourth grade, and she cannot read. Not to say that there aren't White kids who can't read, but being a Black woman from an urban area, this is something that I see that is a problem and I think needs to be addressed. I think being a Black woman opens my mind to problems that technology can help solve. Somebody else who hasn't experienced that kind of thing a lot won't see that problem, and they'll go off and solve another problem, leaving this huge issue that is reading that affects everybody. They might not see it the way I see it. … Being a Black woman … I know that there are problems that affect a young Black girl's self-esteem: body image, the whole hair issue. Those kinds of things, the fact that Black girls don't go into computing, that's a problem. Those are problems that I can specifically solve because I have those experiences. ... I can put my experiences into the solution whereas a White man doesn't know how it feels to have those emotions. His solution is going to come from, "This is what I think will work for them." My solution [*comes from*], "This is what I know helped me so it'll help somebody else."

Jade identified possible barriers to the aspirations of girls of color relative to reading and computing education and careers and to the development of their self-esteem. Her experiences as a woman of color fed empathetic design solutions in a way not possible for others in computing who might have had similar technical expertise but who came from different gender and racial backgrounds. Jade's history gave her motivation and confidence to meet technical challenges, to know that she could overcome them, and to create innovation that will help others.

Kathy, a Latina undergraduate student in computer engineering, described a different kind of integration of her uniqueness, using her interest in psychology to help her decide the kinds of problems she wanted to tackle in her research:

> Psychology is my hobby away from computing. I want to integrate it into my computing later on, which is why rather than just being interested in, say, attacks and defenses for network security, I'm actually interested in the psychology of the people constructing attacks, their motivation and ways of identifying their personality, the way they code or the way they construct a piece of software. That sort of thing just interests me.

In each of these examples, our participants displayed how incorporating unique experiences and interests help them engender their own success by motivating them beyond a simple interest in technology for its own sake. These findings align well with our earlier work[22] that shows that women of color are often motivated by applications of technical expertise that help others or connect to a broader societal benefit.

Developing and Using "Soft" Skills

Several women of color in the study reported developing or using nontechnical ("soft") skills to address potential barriers and support their own success. Sarah, a Black undergraduate computer science student, applied her social skills to ameliorate the competitiveness she sees preventing good rapport and productive communication between herself and men in her computer science department:

> In my department, they are mostly males. … Being a female, I try to use that as a way to just be able to use my female instincts as far as just being more aware of personal things, you know, seeing how you're doing. I try to use that when I

Computing in SCIENCE & ENGINEERING | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page

> communicate with people or professors, … so they won't feel like I'm just coming to take something from them or I just need something. I want them to understand it's a two-way relationship. So I use that, not saying that males don't do that, but generally, females are more inclined to be more nurturing or things like that. So I try to use that to my advantage and build lasting relationships with people, men and women.

In contrast, Daria, an Arab doctoral student in information technology, focused on how she used her extroverted personality to introduce herself to experienced computing professionals, explaining that taking the initiative helped her in both corporate and academic settings:

> Everywhere I go, I pretty much look for mentors. When I was at [*corporation name*], I found a few that mentored me. … [*You*] go out and you're like, "Listen, I'm new here." And you just kind of put yourself out there, right?

Josie, a Black computing professional, explained how maintaining a good rapport with her White male manager, coupled with his valuation of her technical expertise, allowed her to have frank conversations about issues related to diversity and inclusion in their workplace. She described a conversation she had with her manager following a news organization's profile of their workplace in which all of the employees interviewed were White men, something she characterized as "a missed opportunity":

> I was actually brought into this particular position as the right hand, the second brain, if you will, to the [*manager*]. So there is certainly a rapport there, and we certainly respect each other and there is a level of rapport that I think is important because we work together so often. … So it's very much a close relationship and one that we have to trust each other, especially given that I bring a level of expertise and experience that he doesn't have. … But we have a really great relationship and it helps that he respects what I have to offer and what I bring to the table. … [*When explaining the news story*,] he couched his comments with, "You're not going to like this," right? And I said, "That's a missed opportunity." … So I think my comments and his wanting to share that with me came from a place where we both believe that there was more to be done about the current state of affairs [*regarding diversity and inclusion*].

Similarly, Sadie, a Native American doctoral student in computer science, explained how she worked for a full year on establishing good communication between herself and her advisor. She felt that it was worth the time to build her advisor's understanding of the different, and sometimes competing, perspectives she brought to their shared work:

> It's been this process of learning how to communicate most effectively with him. … That has taken effort to get him to consciously think about, as a Native American woman, this is how I would think about it, but as a graduate student, this is how I think about it. Then, as a cultural ambassador, this is how I think about it. We went through this whole year of that process where I was like, "Yo, dude, time out. We have to work on your Navajo etiquette."

In each case described above, the women explained how developing soft skills is key to any successful computing career. They felt this was a skillset often overlooked in the quest for better technical skills, and that neglecting these skills creates an unnecessary additional barrier to success.

Our findings inform an understanding of the ways in which the experiences of women of color are shaped by the intersection of race and gender and how this intersection influences their success in computing. We have described a range of agentic strategies women enacted to engender their own success, ones that confront challenges, draw on their unique experiences, and develop or use soft skills. When considering the implications to these findings, again we found wisdom in the voices of the women themselves.

While we can be inspired by the way women used challenges as motivation, including the challenges of outright failure, the energy they employed could be focused more productively if some of the challenges were removed, including those posed by the isolation of small numbers or perception that any failure is career-ending. Julia, a Latina computing professional involved in STEM education, advised that this could be helped by departments creating a more inclusive culture:

> [*Hidden*] biased training would be wonderful for work places, especially in [*an*] engineering department, because the work can be very heads down. … [*The*] more awareness you have

> of maybe hidden biases or stereotypes, the more inclusive a person can be at the workplace. And if everyone is inclusive then it's going to have an inclusive culture, which is great.

Georgette also advised an inclusive culture through working with her institutions to find the brilliance and uniqueness of students that don't fit the standard model:

> [*There*] is a student [*at my former institution*] who walks around with a cane and always has someone walking him to class, even with a cane. … I bent over backwards to help him succeed in the class, but he worked hard. He was so smart. He's smarter than any other student I've ever, ever met, which makes it all that more enjoyable to help a student like that flourish. Meanwhile, I get to [*my current employer*], and I understand that there is a blind student who is having all kinds of problems with accessibility, whereas [*my previous employer*] saw the student, realized how brilliant he was and … employed him to help him fix systems. And so, I'm hoping now I can do something like that for [*my current employer*].

While Georgette described here how an institution can learn from the seeing "brilliance" of students with physical disabilities, her advice could apply to many aspects of students' backgrounds. As the women in our study demonstrated, a rich source of motivation and inspiration came from the backgrounds of the women themselves. Efforts should be made to expand the kinds of problems computing expertise can contribute to solving, including those that have greater societal aims.

Hasina extended this advice when she suggested that employers allow students and professionals to see the impact of their work in other units or in the world beyond the organization:

> I feel like [*my previous company*] may have it down in terms of … any product that you help with … it's going to go to millions or billions of people. You already have an impact, right away. They have products to scale, and they pay their engineers well. When I was at [*different corporation name*], when I was a software engineer, I didn't see an impact. … I'm really close friends with a guy at [*university*], we're in the same [*subdiscipline*], who looks at different ways of interacting with devices. Like, you can scratch the table as a piece of input; … you know, he does this thing with skin, you can use your skin as an input to a device. I mean, I think that's really exciting. I think the companies that have the pipelines, where they're like visionary … I think that's great. But if you listen to what [*recruiters*] were telling students … they're like, "Here's what we need to solve the problem." They were saying, "We need people. We need people that can code." I just feel like that's shallow. … I feel like you have to do more than code to make a difference.

Hasina's advice was against keeping computing professionals in a closed coding box, but to make the impact of their work more visible. Our findings show that such information can be a strong source of motivation.

While women cited soft skills as tools they develop to overcome barriers, Josie suggested that institutions create spaces where subgroup members can support one another and have opportunities to develop both soft and technical skills:

> [*A*] lot of these organizations … release their numbers on diversity and they're dismal, and companies now are trying to better understand how can we attract more? Well, it's not about attracting [*people of color*], you have to keep them there. You have to ensure that there is upward mobility for people of color, that there are not just communities or subcommunities in your organization that support diverse groups. ... When I was at [*corporation name*] … there were these affinity groups like Hispanics in computing or like Black [*corporation name*]er's. ... So those are really great opportunities for connection and for camaraderie. But what's really important is that you're developing them, these communities technically—are you providing for, again, upward mobility? Are you providing or engaging in practices and developing policies that are inclusive? Those are really important. … You have to change business as usual.

This recommendation also pointed to the need for managers in computing organizations to develop their own abilities to engage regarding issues of diversity and inclusion. As Josie's story in the last section illustrated, the ability to simply have the conversation contributed to building effective rapport with colleagues.

By contributing to the knowledge base about success factors for retaining women of color in computing, we aim to support and enhance efforts of the STEM community to broaden overall participation rates of women of color in computing professions and transform understanding of the challenges they face as they work toward their educational and career objectives. ■

Acknowledgments

We're grateful to Rachel Kachchaf for her dedication in conducting many of these interviews and for her help in coding and organizing the data. We're also grateful to Angela Johnson, our external evaluator, who provided critical feedback in revisions of the interview protocol as well as our analyses of the data that form the basis of this article. This material is based on work supported by the National Science Foundation under grant numbers CNS-1240768 and CNS-1451341. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. A.C. Johnson, "Unintended Consequences: How Science Professors Discourage Women of Color," *Science Education*, vol. 91, no. 5, 2007, pp. 805–821.
2. L.T. Ko et al., "Agency of Women of Color in Physics and Astronomy: Strategies for Persistence and Success," *J. Women and Minorities in Science and Eng.*, vol. 20, no. 2, 2014, pp. 171–195.
3. A. Johnson et al., "Authoring Identity amidst the Treacherous Terrain of Science: A Multiracial Feminist Examination of the Journeys of Three Women of Color in Science," *J. Research in Science Teaching*, vol. 48, no. 4, 2011, pp. 339–366.
4. C. Corbett and C. Hill, *Solving the Equation: The Variables for Women's Success in Engineering and Computing,* Am. Assoc. for University Women, 2015.
5. K.I. Maton et al., "Enhancing the Number of African Americans Who Pursue STEM PhDs: Meyerhoff Scholarship Program Outcomes, Processes, and Individual Predictors," *J. Women and Minorities in Science and Eng.*, vol. 15, 2009, pp. 15–37.
6. S. Malcom, P. Hall, and J. Brown, *The Double Bind: The Price of Being a Minority Woman in Science*, Am. Assoc. for the Advancement of Science, 1976.
7. J. Joseph, "From One Culture to Another: Years One and Two of Graduate School for Black Women in the STEM Fields," *Int'l J. Doctoral Study*, vol. 7, 2012, pp. 125–142.
8. A.J. MacLachlan, "The Graduate Experience of Women in STEM and How It Could Be Improved," *Removing Barriers: Women in Academic Science, Technology, Engineering, and Mathematics*, J.M. Bystydzienski and S.R. Bird, eds., Indiana Univ. Press, 2006, pp. 237–253.
9. L. Espinosa, "Pipelines and Pathways: Women of Color in Undergraduate STEM Majors and the College Experiences that Contribute to Persistence," *Harvard Educational Rev.*, vol. 81, no. 2, 2011, pp. 209–241.
10. M. Ong, "Challenging Cultural Stereotypes of 'Scientific Ability,'" *Everyday Antiracism: Getting Real about Race in School,* M. Pollock, ed., The New Press, 2008, pp. 114–119.
11. K. Crenshaw, "Demarginalizing the Intersection of Race and Sex: A Black Feminist Critique of Antidiscrimination Doctrine, Feminist Theory and Antiracist Politics," *Feminist Legal Theory: Foundations,* D.K. Weisberg, ed., Temple Univ. Press, 1993, pp. 383–395.
12. J.C. Blickenstaff, "Women and Science Careers: Leaky Pipeline or Gender Filter?," *Gender and Education*, vol. 17, no. 4, 2005, pp. 369–386.
13. E.H. Branch, "A Road instead of a Pipeline? Incorporating Agency and Constraint into Our Understanding of Women in Science," presentation at the Eastern Sociological Society Annual Meeting, 2013; contact ebranch@soc.umass.edu.
14. M. Ong et al., "Inside the Double Bind: A Synthesis of Empirical Research on Undergraduate and Graduate Women of Color in Science, Technology, Engineering, and Mathematics," *Harvard Educational Rev.*, vol. 81, no. 2, 2011, pp. 181–196.
15. M. Ong, "Body Projects of Young Women of Color in Physics: Intersections of Gender, Race, and Science," *Social Problems*, vol. 52, no. 4, 2005, pp. 593–617.
16. H.B. Carlone and A. Johnson, "Understanding the Science Experiences of Successful Women of Color: Science Identity as an Analytic Lens," *J. Research in Science Teaching*, vol. 44, no. 8, 2007, pp. 1187–1218.
17. M. Ong and C. Bath, "Women of Color in Computer Science: A Summary of Empirical Literature Findings from Inside the Double Bind," invited talk presented at the National Center for Women and Information Technology Social Science Advisory Board Meeting, May 2009; contact maria_ong@terc.edu.
18. J. Trost, "Statistically Nonrepresentative Stratified Sampling: A Sampling Technique for Qualitative

Studies," *Qualitative Sociology*, vol. 9, no. 1, 1986, pp. 54–57.
19. C. Riessman, *Narrative Methods for the Human Sciences*, Sage Publications, 2007.
20. J.W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, 3rd ed., Sage Publications, 2009.
21. M. Ong, "The Status of Women of Color in Computer Science," *Comm. ACM*, vol. 54, no. 7, 2011, pp. 1–3.
22. A.K. Hodari et al., "New Enactments of Mentoring and Activism: US Women of Color in Computing Education and Careers," *Proc. 10th Annual Conf. Int'l Computing Education Research*, 2014, pp. 83–90.

Apriel K. Hodari is a principal investigator at Eureka Scientific Inc. and serves as co-leader (with Maria Ong) of the Beyond the Double Bind projects, investigating the lived experiences of women of color in science, technology, engineering, and mathematics (STEM). Hodari earned a PhD in physics from Hampton University. Contact her at AKHodari@EurekaSci.com.

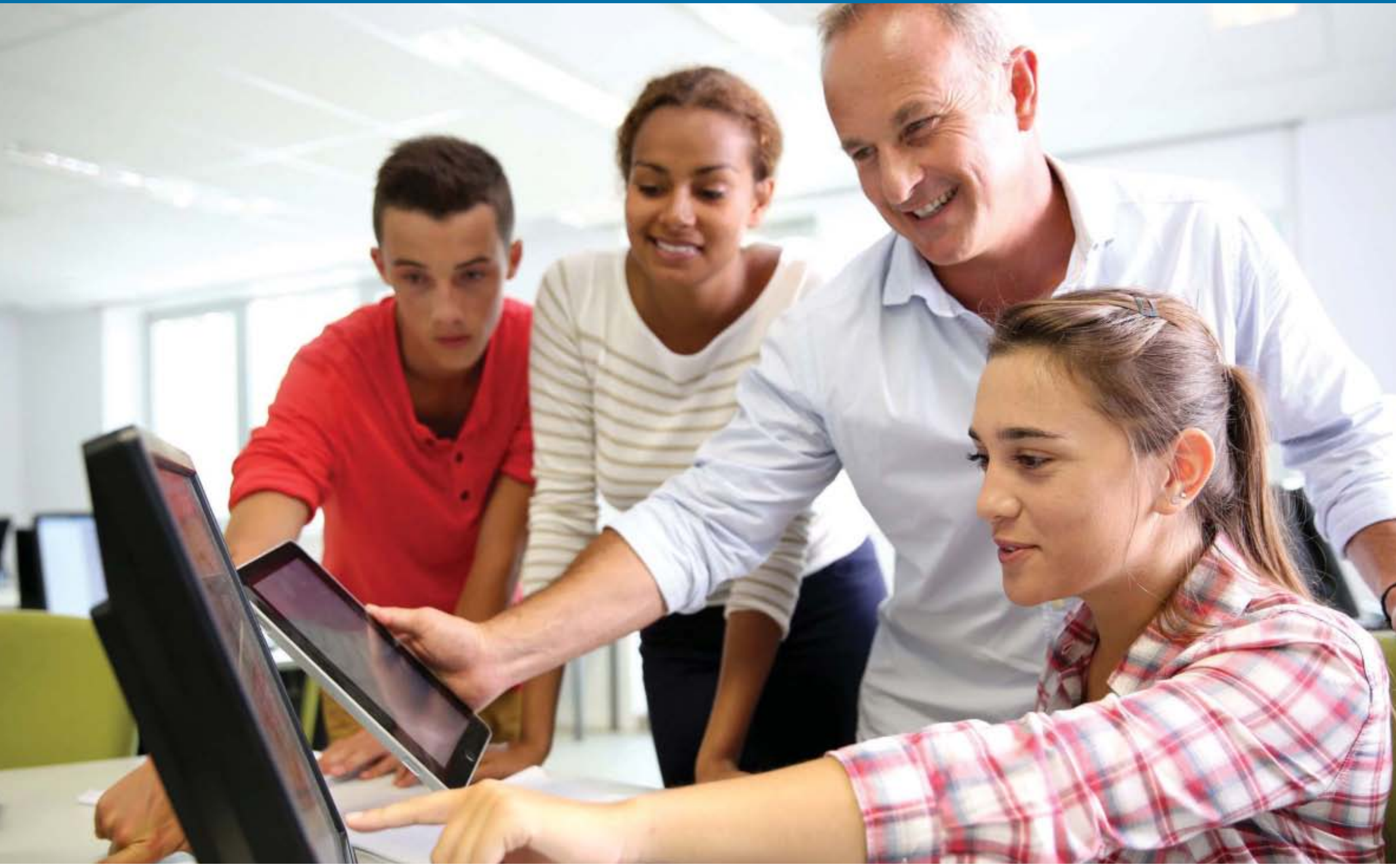
Maria Ong is a senior research scientist and evaluator in the Education Research Collaborative at the Technical Education Research Centers Inc. She holds a PhD in education from the University of California at Berkeley. Ong is a former member of three National Science Foundation advisory boards and currently serves on the Social Science Advisory Board for the National Center for Women in Information Technology. Contact her at maria_ong@terc.edu.

Lily T. Ko is a research associate at the Technical Education Research Centers Inc. Ko has a BS in marketing and management from Northeastern University. She currently works on two NSF-funded projects about women of color in computing and engineering. Contact her at lily_ko@terc.edu.

Janet M. Smith is a senior research associate at the Technical Education Research Centers Inc. Her research interests include issues of equity in STEM, STEAM, and life sciences education. Smith has a PhD in curriculum and instruction, teacher education, and special education from Boston College. Contact her at janet_smith@terc.edu.

CALL FOR NOMINEES

Education Awards Nominations



Taylor L. Booth Education Award

A bronze medal and US$5,000 honorarium are awarded for an outstanding record in computer science and engineering education. The individual must meet two or more of the following criteria in the computer science and engineering field:

- Achieving recognition as a teacher of renown.
- Writing an influential text.
- Leading, inspiring or providing significant education content during the creation of a curriculum in the field.
- Inspiring others to a career in computer science and engineering education.

Two endorsements are required for an award nomination.

See the award information at:
www.computer.org/web/awards/booth

Computer Science and Engineering Undergraduate Teaching Award

A plaque, certificate and a stipend of US$2,000 is awarded to recognize outstanding contributions to undergraduate education through both teaching and service and for helping to maintain interest, increase the visibility of the society, and making a statement about the importance with which we view undergraduate education.

The award nomination requires a **minimum of three endorsements**.

See the award details at:
www.computer.org/web/awards/cse-undergrad-teaching

Deadline: 15 October 2016
Nomination Site: awards.computer.org

IEEE | IEEE computer society

Editors: Barry I. Schneider, bis@nist.gov | Gabriel A. Wainer, gwainer@sce.carleton.ca

Discrete Event Modeling and Simulation-Driven Engineering for the ATLAS Data Acquisition Network

Matías Bonaventura, Daniel Foguelman, and Rodrigo Castro | Universidad de Buenos Aires, Argentina

Robust engineering methodologies offering product lifecycle control have proved to be a cornerstone in modern software development projects. Simultaneously, various modeling and simulation (M&S) techniques have become increasingly adopted in complex system design, particularly in scenarios in which it's difficult to predict system behavior as changes are introduced.

The DEVS (Discrete Event Systems Specification) framework is the most general formalism for modeling discrete event systems[1–3] and has been adopted in several disciplines for complex software and hardware system design and analysis.[4,5] In addition to providing an unambiguous mathematical formalism to define model behavior and structure, DEVS provides a clear framework for system analysis, experimental frame definition, model-to-simulator verification, and model-to-system validation.

We present a DEVS-based methodology for M&S-driven engineering projects that integrates software development best practices tailored to a large-scale networked data acquisition system in a physics experiment (specifically, the ATLAS particle detector[6] at CERN[7]). This project poses M&S challenges from several viewpoints, including system complexity, tight delivery times, the quality and flexibility of the developed models and tools, interdisciplinary communication of results to collaborators (mostly scientists), and big data-scale analysis.

 1521-9615/16/$33.00 © 2016 IEEE Copublished by the IEEE CS and the AIP May/June 2016

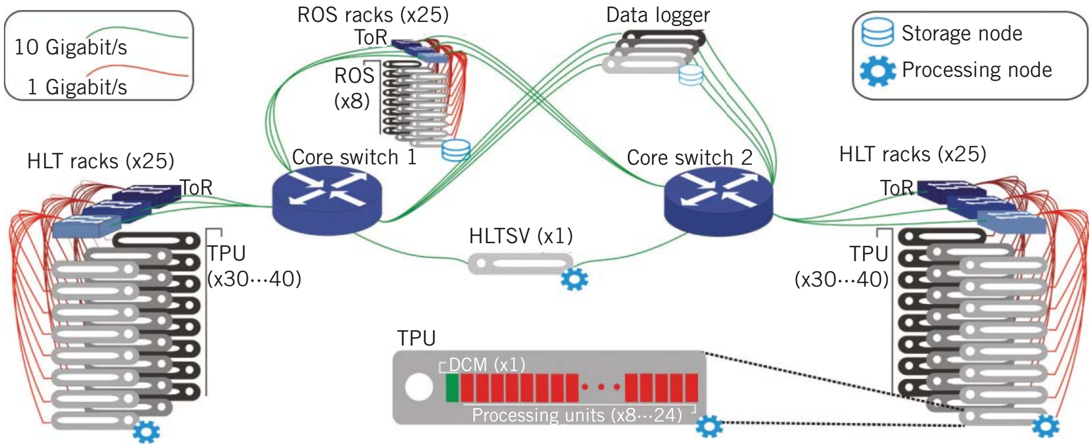


Figure 1. Topology and applications in the high-level trigger and data acquisition (TDAQ) farm. This intermediate configuration is from long shutdown 1 (LS1) in 2014.

The Data Acquisition Network at CERN's ATLAS Experiment

The Large Hadron Collider (LHC)[8] is the world's largest particle accelerator—27 kilometers in circumference—colliding bunches of particles (protons or ions) every 25 ns near large detectors, including ATLAS, CMS,[9] ALICE,[10] and LHCb.[11] In 2013, the Run1 detectors went offline for maintenance and upgrades (long shutdown 1, or LS1) until the Run2 restart in 2015. Collisions in the ATLAS detector generate very high energy, enabling the search of novel physical evidence such as Higgs boson, extra dimensions, and dark matter. Each particle bunch collision is called an *Event* (we use "Event" for high-energy physics and "event" for DEVS modeling) and consists of particle-induced signals registered in the detector and digitized for further analysis. The raw amount of information generated exceeds 60 Terabyte/s.

To assimilate this throughput, ATLAS uses a sophisticated layered filtering system (trigger and data acquisition, or TDAQ[12]) that decides in real time whether each Event should be permanently stored or safely discarded. The first-level trigger (L1) filters Events from an initial raw rate of 40 million Events/s down to a filtered rate of 100,000 Events/s. L1-accepted Events are temporarily stored in a readout system (ROS) in the form of data structures called *fragments* and then accessed by a second-level filter called the high-level trigger (HLT). At the HLT, physics algorithms reanalyze the fragments (this time with a different granularity), retaining only 1,000 "interesting" Events/s. The TDAQ system and its HLT-ROS data network is our system under study.

Applications and Data Network in the HLT

Figure 1 shows the interconnections among various applications in the HLT at the commencement of our case study. Upon selection by L1, Event data is transferred to the ROS, and the specialized application HLT supervisor (HLTSV) is notified. The HLTSV assigns Events to trigger processing unit (TPU) servers, which run an application called a data collection manager (DCM) to centralize communication between the TPU and the rest of the system. DCMs interface with instances of the application processing unit (PU)—one per available core, between 8 and 24 per host. Each Event is assigned to a single PU instance that analyzes it and decides whether it should be permanently stored or discarded. This system represents our starting point for the M&S process.

Applications communicate over an Ethernet network with link capacities of 1 and 10 Gbps. Two core routers and approximately 100 switches interconnect roughly 2,000 multicore servers using TCP/IP protocols. Figure 1 shows a diagram of the network. The farm is composed of 50 racks for TPU servers and 25 racks for ROS nodes. Each TPU rack contains from 30 to 40 servers (DCMs and PU applications), and each ROS rack contains 8 servers. Within each rack, servers are connected to a shared top-of-rack (ToR) switch via 1 Gbps links. The HLTSV node and the ToRs are connected to the core switches over 10 Gbps links.

DEVS for Data Network Modeling

DEVS is a mathematical formalism for M&S based on general systems theory—that is, it's independent of any specific application. DEVS lets us describe

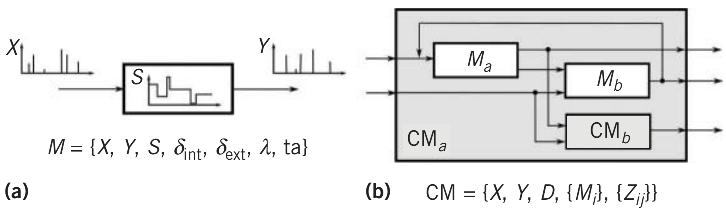


Figure 2. Basic Discrete Event Systems Specification (DEVS) (a) atomic models and (b) coupled models. Coupled models define the structure of the system (interconnections between coupled and atomic models). Atomic models define the dynamic behaviors.

exactly any discrete system and approximate numerically continuous systems with any degree of desired accuracy. The formal model specification provides tools for analytical manipulation and offers independence in choosing the programming language for implementation.[2] DEVS models are described as a hierarchical composition of atomic models (Ms) and coupled models (CMs) defined by mathematical tuples as shown in Figure 2.

CMs define system structure (interconnections between coupled and atomic models), whereas Ms define dynamic behaviors. For Ms, each possible model state $s \in S$ has an associated lifetime defined by the function $ta{:}S \rightarrow R_0^+$. When the model is in state $s = s_1$, at time $t_1 = ta(s_1)$ it autonomously undergoes an internal transition toward a new state $s_2 = \delta_{int}(s_1)$, where $\delta_{int}{:}S \rightarrow S$ is the *internal transition function*. An output event is simultaneously produced at t_1 with value $y_1 = \lambda(s_1)$, where $\lambda{:}S \rightarrow Y$ is the *output function*.

When a model receives an input event $x_1 \in X$, an external transition is triggered that instantly changes the model state to $s_4 = \delta_{ext}(s_3, e, x_1)$, where s_3 is the model state by the time it receives the input event, and e is the elapsed time since the last state transition (with $e < ta(s_3)$). The function $\delta_{ext}{:}S \times R_0^+ \times X \rightarrow S$ is the *external transition function*.

Vectorial DEVS. The DEVS simulation algorithm is universal, unambiguous, easy to implement, and independent of programming languages, with many of its extensions and specializations tackling different needs. We're particularly interested in vectorial DEVS (VDEVS),[13] which lets us model large-scale systems with a compact graphical representation. A vectorial model is an array of quasi-identical classic DEVS models that can differ in their initial parameters. Formally, the vector model's structure is defined by $VD = \{N, X_V, Y_V, P, M_i\}$, where N is the vector dimension, X_v is the set of input events vector, Y_v is a vector set of output events, P is the set of parameters, and each M_i is a classic DEVS model. For the interaction between vectorial and nonvectorial, we define scalar to/from mappings of vector models.

PowerDEVS. We developed a model for TDAQ using the PowerDEVS tool,[14] which provides a graphical interface to define DEVS models via block diagrams, a C ++ editor to code the four dynamic functions for the M tuple, and libraries with reusable models. PowerDEVS also has a native interface to Scilab (www.scilab.org), an open source alternative to Matlab for numerical computation purposes. We adopted a data networks library (queues, servers, traffic generators, a TCP implementation, and so on[15, 16]) and extended it for our case study.

Network-specific simulators strive to represent protocols and hardware nodes in great detail. They typically provide comprehensive and reusable libraries that allow for quick model prototyping—for example, OMNeT++ (www.omnetpp.org), NS2/3,[17] and OPNET[18] (an updated review[19] and a recent simulation study[20] of the TDAQ system using OMNeT++ appear elsewhere).

When adopting prebuilt network frameworks, it's difficult (or even impossible, depending on the software package) to freely choose the desired simulation abstraction level. Experience shows that once a question is defined, several protocol features (or even entire network layers) can become dispensable as they don't contribute significantly to increase result fidelity, but they do increase simulation costs.[21] This poses risks in M&S projects, particularly for large-scale networks.

By adopting a general-purpose discrete event formalism such as DEVS, we partially renounce some out-of-the-box detailed protocol features offered by network-specific packages, but we gain the freedom to decide what kind of representation and granularity suits a given stage of the project. Our strategy for modeling the TDAQ system is to flexibly select a sufficient level of abstraction to answer each particular question with an acceptable fidelity given time and computational resource constraints. Along these lines, we aim to perform hybrid simulations (discrete events mixed with continuous flows). This capability is readily available in DEVS[22] and implemented in advanced versions of PowerDEVS tailored for data networks.[16]

Context, Requirements, and Methodology

For any case study that might arise in TDAQ, crosscutting contexts and requirements call for a flexible yet robust development methodology.

The TDAQ HLT filtering farm is no exception. During LS1, it was subject to hardware and

Table 1. Elicited requirements.

Requirement	Goal
Evaluate candidate changes for the network and control algorithms before their commissioning	Perform early risk assessment
Define in advance the best set of tests to perform on the real system during scarce windows of availability	Harness the test window to focus on the most relevant questions
Enable flexibility for choosing the level of detail/accuracy with which the evaluations are obtained	Dynamically adapt to different and complex modifications that need to be assessed, and then schedule changes

control algorithm changes that affect network topology and throughput, yet predicting the impact these changes have isn't straightforward. Serious design and benchmark studies on system components give confidence, but they require access to the hardware in advance. In the end, testing the system as a whole happens only at the final integration phase.

The full TDAQ system was available for testing only about one out of every six weeks (during scheduled technical runs), which delays testing on new control algorithms that are continuously improved but can't be fully validated until the full system is available.

Table 1 lists the resulting requirements elicited during system analysis meetings. Moreover, these requirements are likely to change dynamically throughout a project's lifetime, with different experts having varying requirements on the same system component's analysis.

To implement an engineering strategy driven by modeling and simulation, we proposed the iterative process-based methodology illustrated in Figure 3.

DEVS Formal Framework

At the methodology's core, the system, model and simulator entities are strictly separated yet formally related by the DEVS framework. The real (or "source") system is experimented under a system experimental frame (EF_S), with questions encoded in the form of system parameters Θ_S that define experimental conditions. Experimental results relevant to the original questions are stored in a system behavior database λ_S.

As a specification of structures and behaviors, every new DEVS model is built for a pair {System, EF_S} according to a modeling relation and guided by selected homomorphisms/isomorphisms. A new model experimental frame (EF_M) also allows for questions about model attributes (using model parameters Θ_M for queries and a model database λ_M to store answers) related to coupling density, model topology, types of variables (discrete, continuous), and so on, with no access to the real system and independent of any simulation exercise.

A DEVS simulator reads a DEVS model and produces an output trajectory by obeying the model's dynamics (in short, a DEVS model is simulated). Its most common realization is a computer program, usually referred to simply as a simulator, which is constructed, adapted, and maintained to read and compute DEVS models efficiently within their EF_M. This establishes a simulation relation. The compute experimental frame (EF_C) defines new questions and parameters Θ_C for experimenting with (simulating) the computable model. It also hosts simulation results in a compute behavior database λ_C. The validation relationship lets us relate back to the original system to validate correctness (λ_S versus λ_C) or to perform scans over EF_S due to unexpected observations discovered in the EF_C.

Cycles and Phases

We organize the flow of tasks in three main cycles: *build* (the model) in blue, *hypothesis* (on the system) in orange, and *explore* (simulation results) in green. While each cycle's goal differs, in all cases the flow across the DEVS formal framework follows the system → model → simulation path. In turn, for each evolution through the cycle, two parallel and cooperative phases are defined: the *system study* phase drives progress according to questions about the system under study, and the *tools development* phase seeks to improve the supporting software algorithms and interfaces, leveraging modeling, simulation, and analysis capabilities.

The build cycle starts with observation and measurement of the system. Its objective is to provide quality models that, once simulated, will exhibit an adequate degree of validation against the original system. The hypothesis cycle exercises on the model several candidate changes to be applied onto the system. Its goal is to find improvement opportunities for the system when it's unavailable or when direct experimentation is too expensive. The explore cycle starts with analyzing

Computing in Science & Engineering | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page

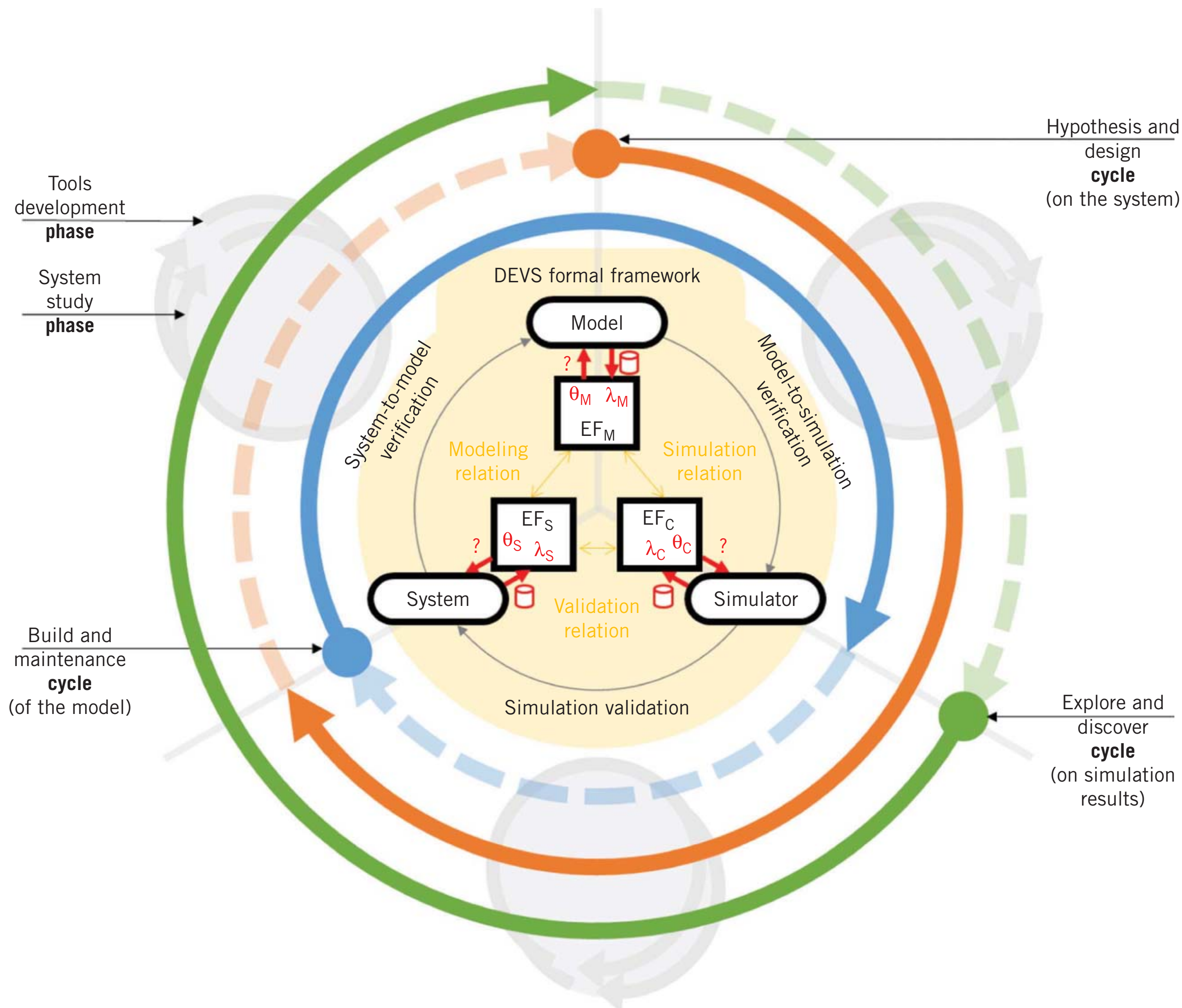


Figure 3. Modeling and simulation-driven engineering. The methodology diagram based on the DEVS formal framework shows iterative cycles and incremental phases.

the large amounts of information produced by simulations; its goal is to discover properties and correlations unthought of during the experimentation phases.

Cycles need not occur in any specific order (although a build cycle is usually required at the beginning of a project). This approach leads to a model that reproduces relevant behaviors of the real system within reasonable simulation times: less relevant dynamics are kept out of the model (such as intrinsics of the network physical layer). The methodology also offers a guideline for development phases of the underlying modeling and simulation software tools; new features are added to the tools at specific phases, responding to specific needs, framed within unambiguous cycle goals.

Existing Techniques and Methods

Software engineering processes and methodologies propose frameworks to control software projects' life cycles—some of the most popular are test-driven development, extreme programming, and the Rational Unified Process. Some of these foster practices such as pair programming or code reviews as part of this work, whereas others propose iterative and incremental cycles, with frequent deliveries focused on adding value quickly.

Our methodology shares some aspects with these approaches. However, none of the aforementioned methods include the formal M&S aspects provided by DEVS: strict separation between modeling formalism, abstract simulation mechanism, and code implementation (of both model behavior and simulation engines). This gives the advantage of independence between experimental frames for the real system, the model, and the simulator, straightforwardly propagating enhancements in any of these three areas to the others. In typical software-based projects, it's unusual to modify the base tools themselves to execute the project. However, in M&S-driven scientific projects, the base tools for modeling, simulation, and data analysis are crucial devices that call for their own requirements alongside requirements of the model itself. Our methodology naturally fills this need.

Large sets of simulation results can support data-driven hypothesis and predictive analytics.[23] A well-structured simulation database together with reusable data analysis libraries can systematize different layers of information aggregation, enabling stratified levels of analyses. Our methodology fosters this approach.

Case Study: Improving the TDAQ Flow and Data Network

Our real-life case study, in which we applied the above presented methodology, starts with two build cycles, observing the system, translating knowledge into an executable simulation model, and upgrading the model to represent important design changes in the system.

This step was followed by an explore cycle, in which we discovered hidden undesirable behaviors in load balancing mechanisms. Such behaviors were confirmed to exist in the real system and raised the need for improvements along with open questions about possible solutions.

To answer the new questions and provide for predictions, we used hypothesis cycles to test alternative scenarios that weren't rapidly exercisable on the real system. We then implemented into the real application a set of improvements that proved satisfactory in the simulated environment. Finally, we evaluated their true effectiveness in the real network by loading the system with emulated physics Events.

The model focuses on predicting HLT dataflow performance. We selected filtering latency as the main performance metric; it represents the time from when the HLTSV assigns an Event to a given PU until when the Event is either discarded or stored.

The sequence diagram in Figure 4 depicts the applications that take part in Event filtering. The PUs request information from the ROS in two stages: L2 filtering and Event building (EB). In L2, a small portion of the Event is first requested and then analyzed; this step can be repeated several times until EB takes place and all pending information is requested as a whole. For each requested portion of the Event, all involved ROS nodes send their replies to the same DCM almost simultaneously, creating traffic bursts from ROS → DCM that increase the filtering latency because of the queuing effect generated at the core and ToR switches.

TDAQ has high bandwidth and low latency in relation to TCP minimum retransmission time (200 ms). Together with the data flow described earlier, these conditions create a TCP throughput collapse known as the TCP Incast pathology.[24] The impact on TDAQ can be huge. Whenever a single TCP packet is discarded at the switches, a PU can't start processing the Event until that packet is retransmitted (after 200 ms at best), raising the perceived network latency of an Event request from a theoretical minimum of 19.2 ms (for 2,400 bytes) to more than 200 ms. To avoid the Incast effect, the DCM application restricts the number of simultaneous requests to the ROS using a credit-based traffic shaping control that limits "in flight" requests on the network.[25] Because responses can vary significantly in their size, traffic shaping doesn't completely prevent packet losses, so it's important to study the effects of queue saturation (and TCP retransmissions) and engineer the network and its algorithms to maximize performance and minimize high-latency risks. This is where our M&S-driven network engineering methodology comes into play.

First Iteration: Building the Model

We start the model implementation with a build cycle (blue cycle in Figure 1). We defined the system experimental frame EF_S for this cycle as a subset of the complete system: the HLTSV, all ROS nodes, and a single instance of the DCM and PU applications. To simplify timing calculations we assumed zero processing time at the PUs, and Events with fixed size (2.4 Mbytes). This EF_S is representative of the entire system with unlimited resources, as each PU independently processes a single Event at a time. Scaling this scenario shows emergent behaviors of resource sharing (DCM credits, network bandwidth, and so on).

Real system measurements. The build cycle begins with observation of the real system (experimentation and metrics acquisition), so we measured filtering latency in different scenarios. Experiments were defined using $\Theta_S = \{\text{number of initial DCM credits}\}$

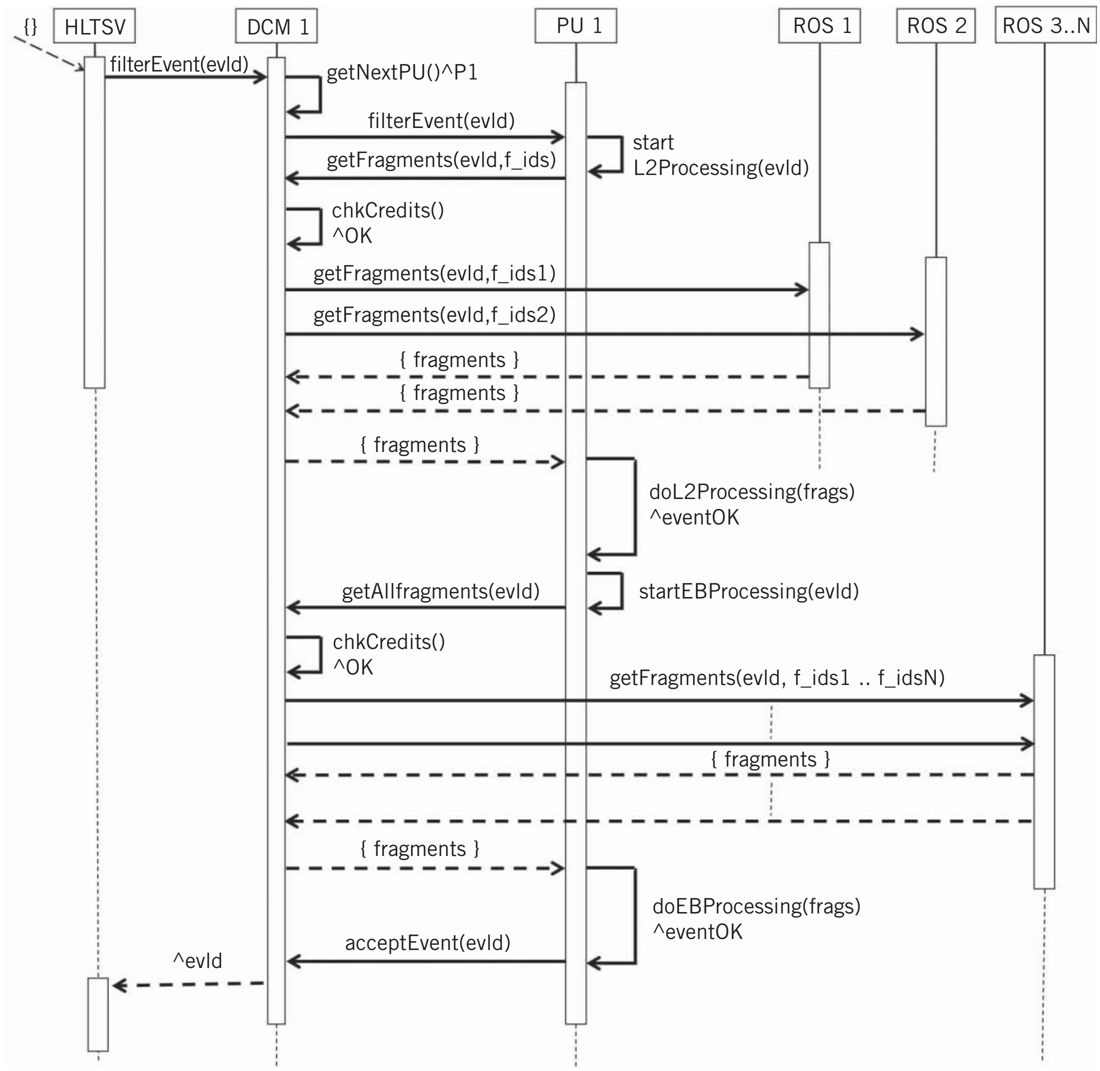


Figure 4. TDAQ application sequence diagram involved in filtering a single Event. The processing units (PUs) request information from the read-out system (ROS) in two stages: level-two (L2) filtering and Event building (EB).

and results stored in λ_S. In Figure 5a, we see an optimum configuration in which average latency stabilizes at 20 ms (close to the theoretical minimum) within a range of about 100 to 600 DCM credits. With fewer credits (12 to 100), latency increases (DCM can send fewer simultaneous requests, underutilizing network capacity). Using more than 600 credits, latency increases rapidly and stabilizes at around 500 ms. We observed packet discards on the ToR switches when more than 600 credits were used, thus confirming that the latency increase is due to network congestion and TCP retransmission (no packet loss was observed at core switches).

Model implementation. The build cycle continues with the creation of a DEVS model guided by the TDAQ architecture and data flow described for a single PU application. Figure 6 shows a PowerDEVS view of the implemented TDAQ model.

To preserve the real system's semantics, we built a hierarchical model complying with TDAQ naming and structure conventions. This greatly facilitated the extraction of control logic from the C++ algorithms in the real applications, thus maximizing the homomorphism with the system under study. The ROS and DCM coupled models implement the TCP flow and congestion control logic based on preexisting PowerDEVS libraries. TCPSender models TCP Cubic,[26] implementing only the TCP behavior relevant to the case study. Tests to validate the TCP model against the real system shifted our focus from the average latency (red curve in Figure 5) to

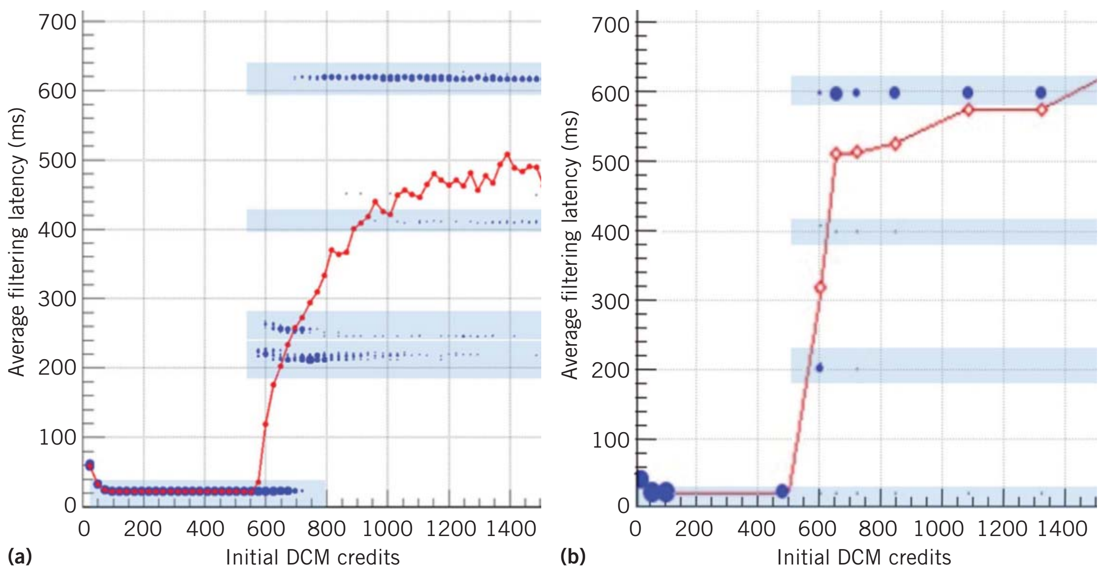


Figure 5. Filtering latency versus initial DCM credits: (a) real system measurements and (b) simulation results. The red curve shows average latency, and blue dots show individual latencies; larger dot clusters denote higher number of occurrences, which gather around discrete ranges (close to 15 ms, 200 ms, 400 ms, and 600 ms).

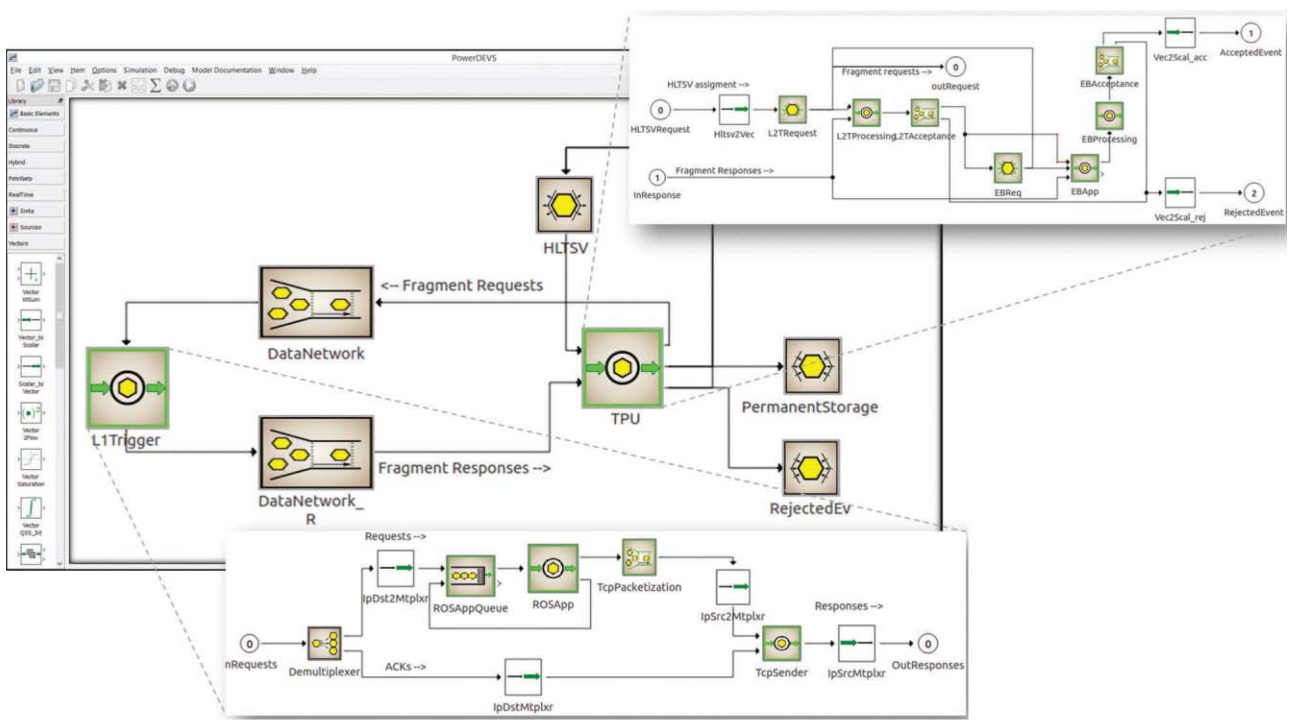


Figure 6. TDAQ simulation model implemented in PowerDEVS. Tests to validate the TCP model against the real system shifted the focus from studying averaged filtering latencies to analyzing clustered latency patterns (red curve vs. blue dots in Figure 5).

the clustered latencies pattern (blue dots). While the explanation for the occurrence of clustered latencies is outside this article's scope, it has a central role in the TCP Incast effect. Moreover, the modeling efforts led to the detection of a bug in the Linux SCL6 TCP implementation that's responsible for the (unexpected) cluster around 600 ms (https://bugzilla.redhat.com/show_bug.cgi?id=1203742).

Computing in Science & Engineering | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page

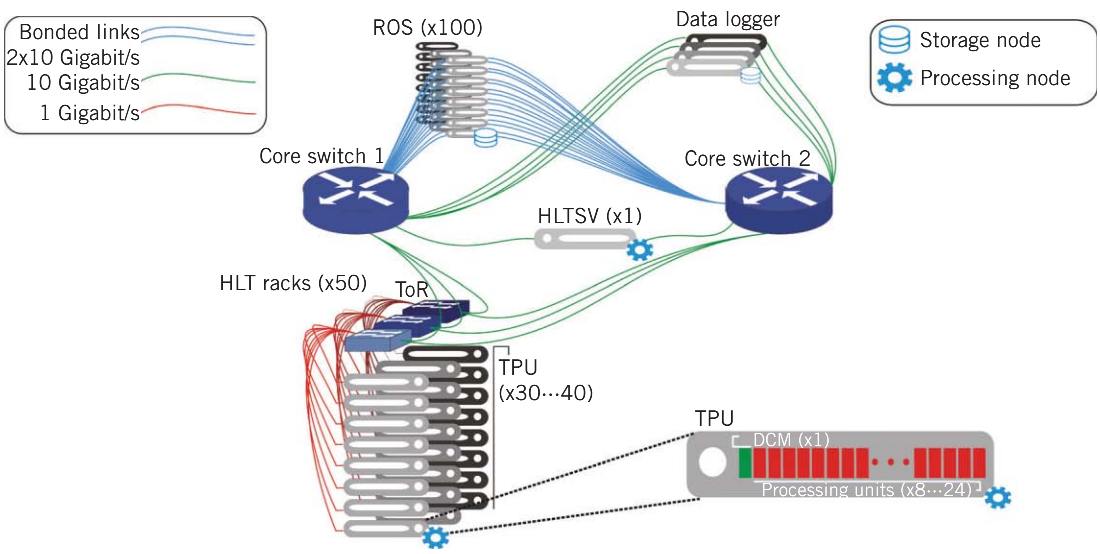


Figure 7. Topology and applications in the TDAQ HLT farm for Run2. This is an upgrade of the one in Figure 1.

Following the tools development approach, we implemented TCP atomic models (sender and receiver) and network elements (channels and switches) to be generic and reusable and incorporated them into the PowerDEVS network library. We also implemented new Scilab and ROOT[27] visualization mechanisms for latency post-analysis along with a new distributed simulation infrastructure, allowing us to execute multiple simulations for parameter sweeping purposes. These tools are meant to be reused in generalized simulation applications.

Simulation results and validation against the real system. The next step of the build cycle is model verification and simulation validation. We configured the simulation to follow the real system setup described earlier (controlled $\Theta_S \rightarrow \Theta_M \rightarrow \Theta_C$ translation), sweeping the number of initial DCM credits. Figure 5b shows the results. The simulation reproduces the individual filtering latencies (blue dots) following the same clustered patterns, validating the TCP dynamics (retransmissions and TCP Incast effect). The simulated average latency approximates real measured latencies ($\lambda_S \sim \lambda_C$), with 100 to 600 credits attaining minimum latency and fewer than 100 credits slightly increasing latency. For credits above 600, the simulation showed congestion and packet drops on the ToR switches, but the increase in average latency was much steeper compared to the real system. Another difference was the stabilization point under congestion: the real system latency stabilizes at 500 ms, whereas the simulated latency grows up to 700 ms. Although these differences require further study, the simulation reproduces very closely the intervals of major interest, underlining the constant tradeoff among degrees of model detail, simulation accuracy, and delivery times for a given engineering concern.

An important advantage of the simulated model is that it allows for fine-grained analysis (packet by packet if required). For example, link utilization and queue occupancies can be visualized and studied in detail in the simulation, but it's impossible to sample the instantaneous evolution of queue occupancies at network devices (for example, to pinpoint queuing bursts that are critical for TDAQ and occur in less than 8 ms).

Second Iteration: System Upgrade and Model Improvements

In the second iteration of the build cycle, we expand the system's experimental frame EF_S by increasing the number of TPUs and of PU applications on each TPU. During this cycle, the real system was upgraded, calling for changes in the model.

Changes in network topology. The TDAQ team commissioned several changes in the HLT network in preparation for ATLAS's Run2 phase, which doubles the maximum particle's collision energy. The ROS ToR switches were removed and the 200 ROS nodes replaced by 100 new computers with four 10 Gbps interfaces, each directly connected to both core switches. The ToR switches were expanded with additional 10 Gbps links to both core switches. The overall throughput supported at the network level increased by one order of magnitude (see Figure 7).[28]

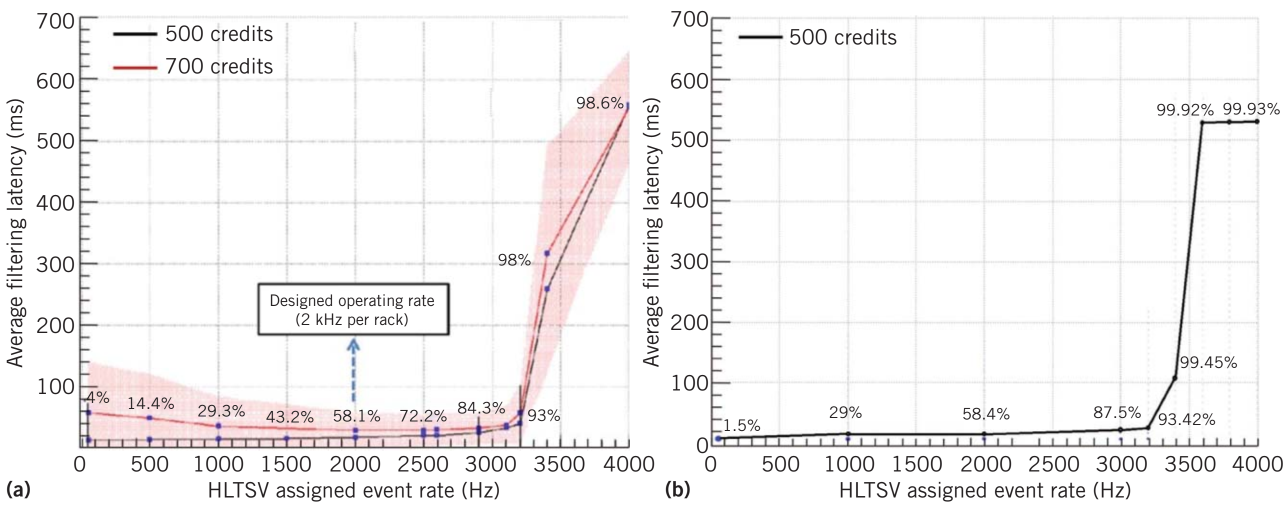


Figure 8. Average Event latency sweeping the HLTSV assignment rate (200 ROS, 1 TPU rack with 40 DCMs, 960 PUs): (a) real system measurements and (b) simulation results. Percentages represent network load, and red background shows standard deviation.

Real system measurements. Again, the first step in the build cycle is taking real metrics from the upgraded system (all new ROS nodes and a full rack of TPUs), where the network traffic is largely determined via HLTSV assignment rate. With a 100 kHz rate for the HLTSV and 50 TPU racks (full farm), each rack should handle Events at 2 kHz. Thus, new experiments must sweep this parameter ($\Theta_S = \{$HLTSV rate$\}$), ranging from 50 Hz (nonsharing of resources; Events are processed faster than 20 ms) up to 4 kHz (network saturation point). To simplify the analysis, we used a synthetic configuration: PUs accept Events 50 percent of the time, Event size is 1.3 Mbytes, and the DCM uses 500 and 700 credits.

Figure 8a shows the average Event latency for increasing the HLTSV assignment rate. When the HLTSV assigns Events at 50 Hz, latency is minimal (13 ms) because the network is completely free when applications start filtering Events. For increasing assignment rates, latency rises as several PUs simultaneously request Events competing for finite network resources and DCM credits. For rates above around 3.2 kHz, latency increases exponentially as the network approaches its maximum capacity (93 percent utilization).

Model implementation. Model changes related to topology upgrades were minimal: the ROS ToR switch models were easily removed, thanks to the modularity fostered by DEVS, and the channel's configuration changed to match the new link capacities. This shows the model's flexibility and the advantage of having a one-to-one mapping between components of the real system and the simulation model. At this stage, we developed a complete HLTSV implementation, reusing directly some chunks of C++ code from the real HLTSV application for greater reliability. To increase the number of model instances, we used VDEVS, developing 16 new vectorized DEVS models and 10 new multiplexer models to represent packet routing.

For the tools development phases, we implemented three generic solutions to address the scalability requirement of increasing the number of simulated instances 50 times. VDEVS's original proposal was extended, allowing for C++11 SmartPointers in vector DEVS messages. SmartPointers were also included directly in the PowerDEVS simulation base engine to allow for automatic and transparent memory management in any atomic DEVS model. This approach dramatically reduced the simulator's memory footprint, pushing its scalability to the next order of magnitude. We also developed a new general framework for PowerDEVS to automatically launch simultaneous simulations on distributed nodes, reducing simulation times for parameter sweeping experiments linearly with respect to the number of nodes used (simulations are completely independent).

Simulation results and validation with the real system. To complete the build cycle, we validated the simulations against the real system, replicating previously conducted experiments to sweep the HLTSV assignment rate parameter. We executed nine experiments, each simulating 60 seconds (180,000 filtered Events in the most stringent case) in three different nodes, completing all simulations in 120 minutes. As Figure 8b shows,

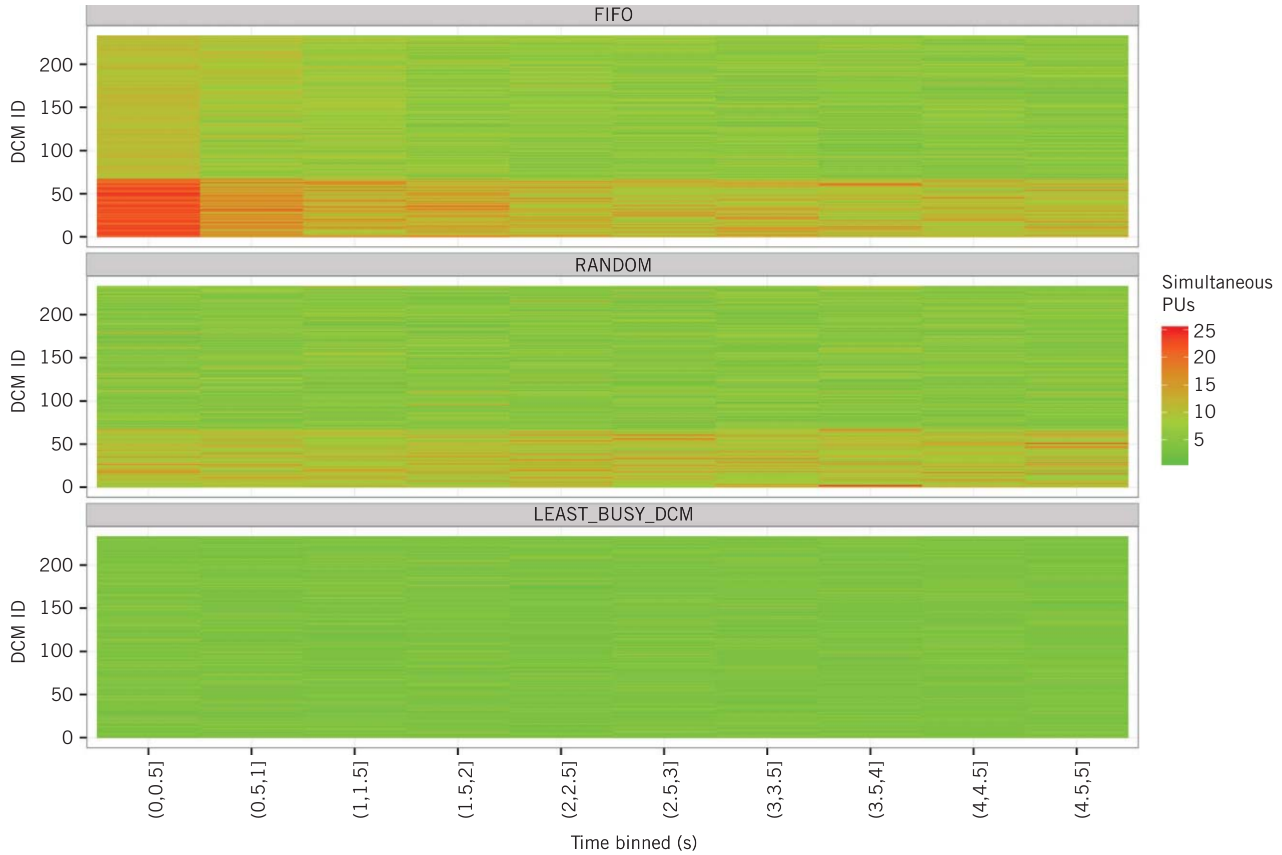


Figure 9. Heatmap of the load in the HLT farm for different HLTSV assignment policies. Tile color represents the maximum amount of PUs simultaneously processed in each DCM (230 DCM IDs in the vertical axis) in 0.5 s (5 s binned in the horizontal axis).

the simulation results closely reproduce the latency curve measured in the real system. The absolute latency values and network load on the simulation differ from reality within an acceptable range (less than 5 percent difference), showing a good degree of validation.

Third Iteration: Exploring and Discovering System-Level Behaviors

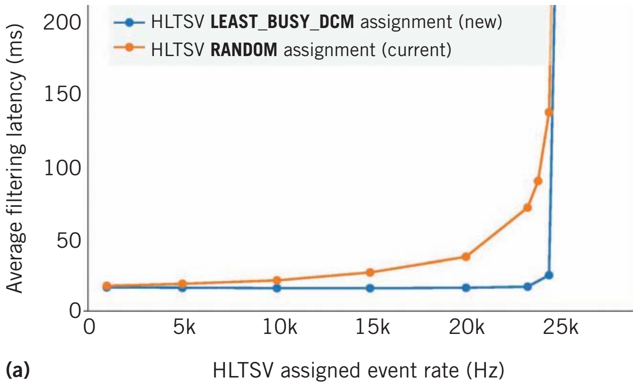
The simulation accuracy obtained in previous cycles provides us with a sufficient confidence level on the simulated data that justifies running an explore cycle (green cycle in Figure 3) to find potential emergent behaviors.

Full system simulations generate huge volumes of information for the 6 million Events filtered in a single minute (processing times, filtering latencies, queues occupancy, link usages, farm utilization, and so on). Some of this information isn't available in the real system or is too difficult to gather uniformly for post-analysis goals.

Figure 9 is an example data analysis performed on the simulation results for λ_C. It shows how Events are distributed across the farm in different time slots using various load-balancing algorithms: first in first out (FIFO) is the default policy implemented for selecting the TPU node that will filter the next Event.

In the FIFO policy, the reddish area at the bottom explains the fact that 30 percent of the DCMs had double the amount of PUs available for processing, thus explaining their higher load. All DCMs are heavily assigned in the first time bins; after a few seconds of execution, load becomes similar to the RANDOM algorithm (each Event's filtering time differs). Another detected system-level behavior is that individual DCMs differ significantly in the number of Events they process—the color intensities vary noticeably along any single row and along any single column. These observations led us to infer that a potentially uneven load-balancing mechanism might be the cause of overall higher filtering latencies.

For the tools development phase, we developed a set of reusable R libraries for data analysis and visualization of the large volumes of logging information produced by PowerDEVS. The new graphical



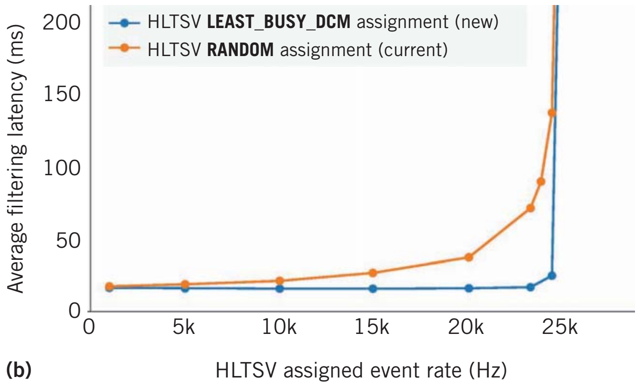


Figure 10. Comparison of assignment policies (RANDOM versus LEAST_BUSY_DCM) for (a) real system measurements and (b) simulation results. The RANDOM algorithm exhibits the same behavior as the FIFO algorithm, while the new algorithm maintains average Event latency close to a minimum (16 ms) for all frequencies below 24 kHz.

information generated through the R platform (such as the heatmaps in Figure 9) became standard means of communication with the TDAQ team.

Fourth Iteration: Real System Improvement Proposal

The revealed behavior discovered during the explore cycle moved us forward to the hypotheses cycle (orange cycle in Figure 3) to test new load-balancing algorithms in a simulated domain in search of improved performance.

Testing the hypothesis on the model. The latency's linear increase in Figure 8 is the effect of several PUs competing for the same resources (reddish tiles in Figure 9). However, in our synthetic experiment for the designed frequency (2 kHz per rack), each DCM receives on average one assignment every 50 to 60 ms, while the minimum latency for filtering an Event is roughly one third of this period (with an unloaded network). Under these conditions and an optimal assignment policy, it isn't necessary for two PUs to process simultaneously on a single DCM. However, such assignments currently behave as random (uniformly distributed), so sometimes 10 to 25 PUs of the same DCM process simultaneously while other DCMs are almost idle (DCM load in Figure 9).

We modeled three HLTSV assignment policies: FIFO, used by the real system; RANDOM, in which the HLTSV selects a random idle PU; and the new LEAST_BUSY_DCM, in which the HLTSV selects an idle PU within the DCM with fewer busy PUs. The main idea behind LEAST_BUSY_DCM is to revert the uneven load detected in the explore cycle by assigning Events according to the load on each DCM.

Simulation results. After implementing new alternatives in the model, we performed simulations to compare the RANDOM algorithm with the proposed LEAST_BUSY_DCM algorithm (FIFO is omitted because it eventually becomes equivalent to RANDOM as shown in Figure 9). To compare, we simulated the same experiment as in the second iteration (sweeping the HLTSV rate) but configured nine TPU racks (267 DCMs and 6,408 PUs). Figure 9 shows that LEAST_BUSY_DCM effectively balances the load of all DCMs in the farm, reducing the amount of simultaneous PUs processing in each DCM (tile colors present more similarity along rows and columns). Figure 10b shows simulation results comparing both algorithms. The RANDOM algorithm exhibits the same behavior as the FIFO algorithm, while the new algorithm maintains average Event latency close to a minimum (16 ms) for all frequencies below 24 kHz. For higher frequencies, the latency grows exponentially due to network congestion. These results suggest that the new algorithm could reduce latency between two to four times for this specific configuration (design rate of 15 kHz with a network saturation point of 23 kHz). New tests are under way with more realistic data flow to increase validation confidence.

Implementation and validation in the real system. Once we test the hypothesis in the simulation, the next step in the hypothesis cycle is to implement changes to validate against the real system. It was possible to reuse some C++ code developed for models in the simulation, with minor adaptations to attain close-to-real-time performance (the 100 kHz rate requirement for HLTSV is a stringent one). We

Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page

performed the same experiment in the real system, with HLTSV rate sweeping using nine TPUs racks. Figure 10a shows the result of comparing RANDOM and LEAST_BUSY_DCM algorithms in the real system. With the new algorithm and rates under 24 kHz, the average latency is kept to a minimum and shows improvements of two to four times compared to the current FIFO algorithm, as predicted in the simulation. The simulation is thus validated, showing that the model is capable of reproducing known behaviors, representing a valuable tool to predict the impact of changes in the real system.

We're currently implementing our model with a variety of TDAQ scenarios in which we study different candidate traffic control techniques in search of further performance improvements (in particular, looking for quick recovery times in the face of system failures). We also plan to apply our methodology and tools to assess candidate ATLAS upgrades (planned for 2018), comparing performance and modeling techniques with other simulation frameworks. Ongoing research aims to automate parameterization-simulation-validation cycles by retrieving real run parameters and metrics recorded in the ATLAS Information Service database. ■

References

1. B.P. Zeigler, *Theory of Modeling and Simulation*, John Wiley & Sons, 1976.
2. A.C.H. Chow and B.P. Zeigler, "Parallel DEVS: A Parallel, Hierarchical, Modular, Modeling Formalism," *Proc. 26th Conf. Winter Simulation*, 1994, pp. 716–722.
3. B.P. Zeigler, H. Praehofer, and T.G. Kim, *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems,* Academic Press, 2000.
4. G. Wainer, *Discrete-Event Modeling and Simulation: A Practitioner's Approach,* CRC Press, 2009.
5. G. Wainer and J. Mosterman, *Discrete-Event Modeling and Simulation: Theory and Applications,* CRC Press, 2010.
6. ATLAS Collaboration, "The ATLAS Experiment at the CERN Large Hadron Collider," *J. Instrumentation*, vol. 3, no. 8, 2008, p. S08003.
7. D. Pestre, "L'organisation Européenne pour la Recherche Nucléaire (CERN): A Succès et Politique Scientifique," *Vingtieme Siecle, Revue d'Histoire*, JSTOR, 1984, pp. 65–76.
8. L. Evans and P. Bryant, eds., "LHC Machine," *J. Instrumentation*, vol. 3, no. 8, 2008, p. S08001.
9. CMS Collaboration, "The CMS Experiment at the CERN LHC," *J. Instrumentation*, vol. 3, no. 4, 2008, pp. 1748–0221.
10. K. Aamodt et al. (ALICE Collaboration), "The ALICE Experiment at the CERN LHC," *J. Instrumentation*, vol. 3, no. 8, 2008, p. S08002.
11. A.A. Alves Jr. et al. (LHCb Collaboration), "The LHCb Detector at the LHC," *J. Instrumentation*, vol. 3, no. 8, 2008, pp. 1–205.
12. ATLAS Collaboration, *ATLAS High-Level Trigger, Data-Acquisition and Controls,* tech. report, CERN-LHCC-2003-022, ATLAS-TDR-016, CERN, 2003.
13. F. Bergero and E. Kofman, "A Vectorial DEVS Extension for Large Scale Parallel System Modeling and Simulation," *Simulation*, vol. 90, no. 5, 2014, pp. 522–546.
14. F. Bergero and E. Kofman, "PowerDEVS: A Tool for Hybrid System Real-Time Modeling and Simulation," *Simulation*, vol. 87, nos. 1 and 2, 2011, pp. 113–132.
15. R. Castro, "Integrative Tools for Modeling, Simulation and Control of Data Networks" (in Spanish, extended summary in English), PhD dissertation, Control Dept., Nat'l Univ. Rosario, Argentina, 2010.
16. R. Castro and E. Kofman, "An Integrative Approach for Hybrid Modeling, Simulation and Control of Data Networks Based on the DEVS Formalism," *Modeling and Simulation of Computer Networks and Systems: Methodologies and Applications,* M.S. Obaidat, Z. Faouzi, and P. Nicopolitidis, eds., Morgan Kaufmann, 2015, chapter 18.
17. T. Issariyakul and E. Hossain, *Introduction to Network Simulator NS2*, Springer, 2008.
18. X. Chang, "Network with OPNET Simulations," *Proc. 31st Conf. Winter Simulation*, 1999, pp. 307–314.
19. J. Suárez et al., "Computer Networks Performance Modeling and Simulation," *Modeling and Simulation of Computer Networks and Systems: Methodologies and Applications*, M.S. Obaidat et al., eds., Morgan Kaufmann, 2015, ch. 7.
20. T. Colombo et al., "Modeling a Large Data-Acquisition Network in a Simulation Framework," *Proc. Cluster Computing Conf.*, 2015, pp. 809–816.
21. J.L. Burbank, W. Kasch, and J. Ward, *An Introduction to Network Modeling and Simulation for the Practicing Engineer*, John Wiley & Sons, 2011.
22. F. Cellier and E. Kofman, *Continuous System Simulation*, Springer Science & Business Media, 2006.

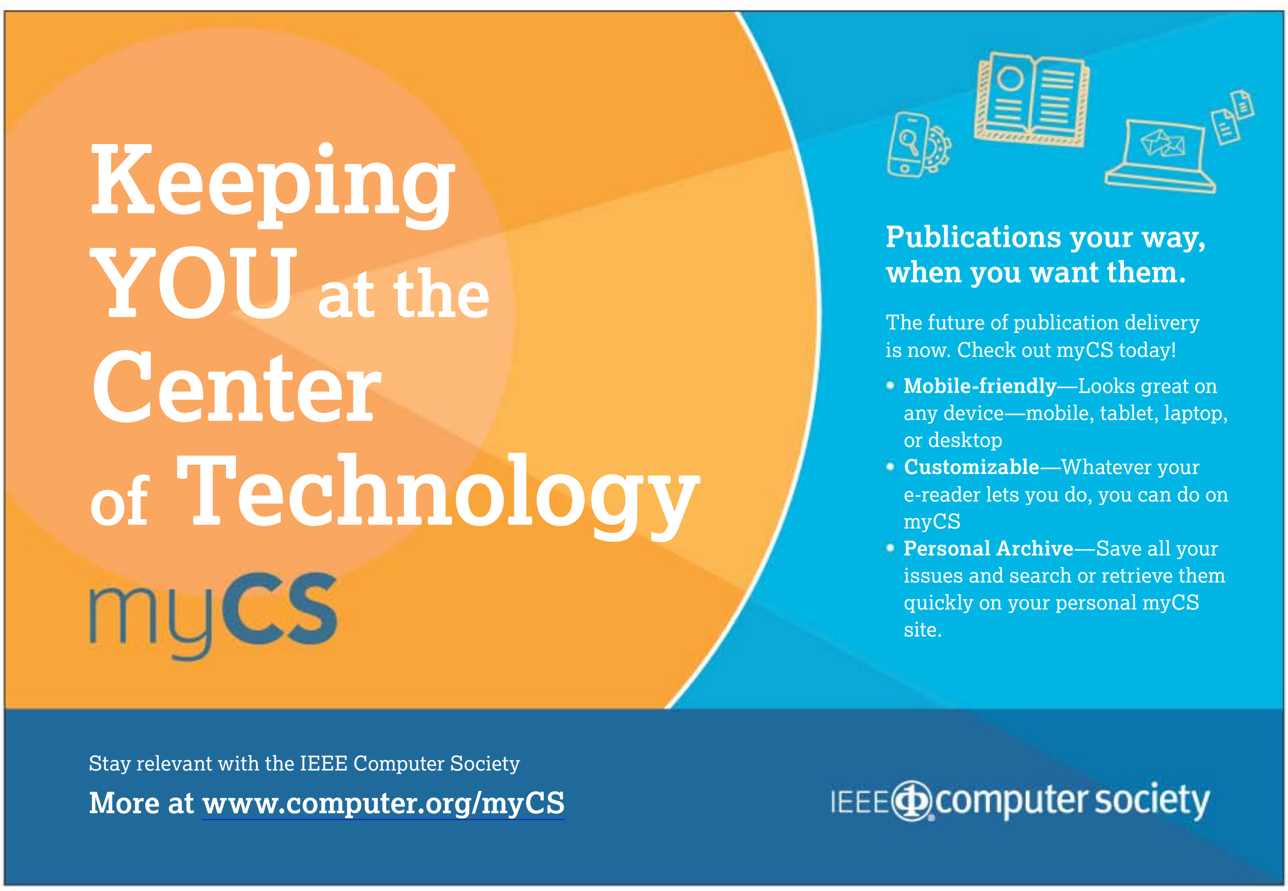
23. B. Gonçalves and F. Porto, "Managing Scientific Hypotheses as Data with Support for Predictive Analytics," *Computing in Science & Eng.*, vol. 17, no. 5, 2015, pp. 35–43.
24. S. Kulkarni and P. Agrawal, *Analysis of TCP Performance in Data Center Networks,* Springer, 2014.
25. T. Colombo (on behalf of the ATLAS Collaboration), "Data-Flow Performance Optimisation on Unreliable Networks: The ATLAS Data-Acquisition Case," *J. Physics: Conf. Series*, vol. 608, no. 1, 2015, p. 012005.
26. S. Ha, I. Rhee, and L. Xu, "CUBIC: A New TCP-Friendly High-Speed TCP Variant," *ACM SIGOPS Operating Systems Rev.*, vol. 42, no. 5, 2008, pp. 64–74.
27. I. Antcheva et al., "ROOT: A C++ Framework for Petabyte Data Storage, Statistical Analysis and Visualization," *Computer Physics Comm.*, vol. 182, no. 12, 2011, pp. 2499–2512.
28. M.E. Pozo Astigarraga (on behalf of the ATLAS Collaboration), "Evolution of the ATLAS Trigger and Data Acquisition System," *J. Physics: Conf. Series*, vol. 608, no. 1, 2015, p. 012006.

Matías Bonaventura is a MASc in computer science and a PhD student in the Departamento de Computación, Facultad de Ciencias Exactas y Naturales, at the Universidad de Buenos Aires, Argentina. Contact him at mbonaventura@dc.uba.ar.

Daniel Foguelman is a MASc in computer science and a PhD student in the Departamento de Computación, Facultad de Ciencias Exactas y Naturales, at the Universidad de Buenos Aires, Argentina. Contact him at dfoguelman@dc.uba.ar.

Rodrigo Castro is a PhD in electrical engineering (Universidad Nacional de Rosario, Argentina) and professor in the Departamento de Computación, Facultad de Ciencias Exactas y Naturales, at the Universidad de Buenos Aires, Argentina. Contact him at rcastro@dc.uba.ar.

cn *Selected articles and columns from IEEE Computer Society publications are also available for free at http://ComputingNow.computer.org.*



Computing in SCIENCE & ENGINEERING | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page | Qmags THE WORLD'S NEWSSTAND®

Expanding the Scope of High-Performance Computing Facilities

Thomas D. Uram | Argonne National Laboratory
Michael E. Papka | Argonne National Laboratory and Northern Illinois University

Participants in the race to build exascale supercomputing face familiar challenges: How do you manage billion-way concurrency? How do you achieve high performance in the face of new accelerator technologies? Can we code for some reasonable performance portability? Such worthwhile questions, focused on the construction and deployment of exascale resources, will be answered by the computing and computational science communities in the years to come.

We want to emphasize a complementary role for our own high-performance computing (HPC) facility and ask an additional question: How could our center evolve to become a highly usable service facility, deeply integrated into science projects and serving as a hub for our users' science communities? Computational science facilities should introduce services that integrate with user environments and improve the usability and utilization of HPC centers, particularly among users of experimental and observational facilities; these services could support programmatic job submission and monitoring interfaces, gateways, flexible scheduling, and analysis and visualization.

Simulations running in HPC facilities routinely produce multi-terabytes to petabytes of data that must be made available to all project team members. Teams working in large-scale science facilities face the same challenge and often support project communities with hundreds or thousands of members. While the use of certain HPC technologies might be prevalent among these users today, such as workstations with HPC chips, those technologies are deployed at a much smaller scale than what HPC computing facilities often provide. Facilities such as the

 1521-9615/16/$33.00 © 2016 IEEE Copublished by the IEEE CS and the AIP

Flexible reservation systems could support jobs with varying needs, ranging from real-time computing to supporting immediate feedback of experimental results to less time-intensive jobs.

Large Hadron Collider, the Large Synoptic Survey Telescope, the Deep Underground Neutrino Experiment, and the nation's light sources will generate similarly massive data volumes in the next decade that will require storage and compute resources on a scale typically available only at HPC facilities.

Gateways

Many projects establish data management environments to support movement, postprocessing, and dissemination of raw and derived data products, with computing needs spanning from data collection to analysis of highly processed data. Linking these environments to the large HPC centers would allow experimental facilities to combine their own resources with large-scale compute and storage resources. This integration would support a services-based, large-scale, distributed data management and analysis solution of tremendous complexity, presented to researchers in a consistent, familiar application. Analysis and visualization tools common among researchers in a particular domain could be provided in this application and, by virtue of the services approach, be updated for all users as the tools improve over time, without requiring users to upgrade to new versions of software or install software on new computing systems. These combined capabilities are often delivered to users through a Web interface, commonly known as a gateway.

HPC facilities routinely store petabytes of simulation results for their users, who typically conduct further analyses on these data over the course of their multiyear allocations, either using the supercomputer or accompanying data analysis clusters. The same can be said for many large science facilities that have or will need gateway-like infrastructure for managing their data and analyses. The challenge for these facilities is the development and maintenance of such infrastructure. Fortunately, HPC facilities are a natural place to host such a gateway. A lot of work has been done by the community to address some of these issues, which can be seen in the success of the science gateway efforts of the TeraGrid and XSEDE programs.[1] Still, more work needs to be done to fully support the idea in areas of job submission, monitoring, and scheduling.

Programmatic Job Submission and Monitoring Capabilities

As part of the gateway ecosystem, service gateways are needed from which users can schedule workflows—moving data to HPC systems, submitting simulation jobs on HPC systems and subsequent analysis jobs on the accompanying analysis/visualization systems, and moving data back out—to produce a more usable environment that accommodates the process of scientific exploration. Facilities currently support moving data into and out of the facility using services; namely, Globus Online.[2] By extending this programmatic capability into job scheduling and monitoring, the full operations of the HPC facilities would be available as a service for integration with applications such as data acquisition workflows and gateways. Flexible reservation systems could support jobs with varying needs, ranging from real-time computing to supporting immediate feedback of experimental results to less time-intensive jobs. While the need for real-time supercomputing is evident in multiple disciplines (for example, hurricane tracking or light source calibration), enabling it would involve many technical and social innovations at the HPC centers. Also, increasingly prevalent iPython-like notebooks used by individual researchers could integrate this functionality.

Flexible Scheduling

Large HPC resources, like those found in leadership computing facilities, are batch scheduled and must balance the trade-off of utilization and job sizes. These resources can approach utilization as high as 90 percent, even when focused on long-running jobs—a testament to the quality of their scheduling algorithms. But the remaining 10 percent of computational cycles that go unused would, for many scientists, transform their ability to achieve computational science results. These facilities aim to enable computational science at the largest possible scale—that is, they focus on projects that require a significant amount of the resource for a significant amount of time. In reality, job sizes vary from the full machine down to the smallest allocable unit. This mixture of job sizes inevitably requires the scheduler to pack jobs onto the machine as efficiently as possible under the constraints of job

Extending the scheduler to better accommodate these requests would enable more projects to take advantage of the available resources.

size, job duration, and queuing priority; its ability to do this task determines the utilization of the machine. In its efforts to make room for a job to run, a scheduler often aggregates blocks of compute nodes as they finish until it reaches the required block of nodes for the next job. Because jobs don't finish at the same time, some portions of the required block become available before others. The scheduler fills these blocks with jobs that fit in the available size/duration window.

Extending the scheduler to better accommodate these requests would enable more projects to take advantage of the available resources. By allowing jobs to specify multiple size/duration combinations, the scheduler could choose the size that optimizes job packing, yielding more compute hours available for science. For example, 10 percent of today's typical leadership-class computing resource nominally represents hundreds of millions of core hours per year, a significant amount of computing. This number will only grow with exascale systems, which will be 100 times bigger. By extending the scheduler's capabilities, initiating policy changes at the facility, and expanding the facility's workload, we could enable more science by taking advantage of these otherwise lost resources. A simple example within the leadership-class resource space would be to inject small, short-running jobs onto the compute nodes while they wait for the next big job.

Additionally, more flexible scheduling is needed to support next-generation science facilities. These facilities are expected to generate data that far exceeds their local compute capabilities and will need computational help from elsewhere. Progress needs to be made in batch scheduling to support real-time use, in the preemption of jobs to enable real-time use, and in automated and reliable data movement, management, and staging. Much work has been done, and is being done, in the data space. The real open research areas are real-time scheduling and preemption, both of which will require shifting policy within the facilities to accept smaller project workloads, as well as addressing several technical challenges.

Proactive Supercomputing

By integrating job submission into scientific exploration, the process of defining jobs could be parameterized such that when users identify regions of missing data or results, the jobs required to produce them could be automatically devised and submitted (after approval by the scientist). This interface would vary between science domains, from simply taking the form of spreadsheets or plots all the way up to large-scale 3D visualization. We could even imagine integrating recent advances in machine learning to produce job recommendations in the same way that Amazon makes product recommendations based on recent purchase history. Such an effort would require that we capture sufficient metadata about both the simulation and the job, as well as some understanding of the domain itself. Metadata capture is, in many projects, still largely managed in an ad hoc manner and manifested in any number of ways, ranging from formal project-specific systems to file-naming conventions to the notes of the individual researchers conducting the experiments. With a uniform metadata solution, outfitted with application-specific adapters, researchers could leverage a common platform for analyzing their job and simulation data in terms of performance metrics and science metrics in a Web-based simulation dashboard.[3] The data would be useful for individual projects as a deeply detailed view into their simulations as well as a long view over the duration of their allocations, particularly if augmented with domain-specific analyses. Having common metadata across projects would also be a substantial benefit to the HPC centers in analyzing performance across applications.

Automated Analysis and Visualization

Scientific visualization has been tremendously successful in improving our understanding of simulation data, but it often requires large resources and significant expertise to achieve useful results. Open source scientific visualization software such as ParaView and VisIt allows researchers to visualize their data and have commoditized visualization access in many fields. In many respects, however, reaping the benefits of visualization requires a deeper understanding of how to apply these tools to reveal relationships in the data, and this is especially true at large scales, where data sizes surpass the memory of a researcher's desktop machine and demand instead the power of a data analysis cluster. Data analysis clusters in leadership facilities typically have direct access to simulation data, and on this basis, at least some part of the visualization process could be automated. Given a dataset, a data-type-aware visualization infrastructure could determine the required execution configuration for a tool to produce imagery and run the jobs, delivering the results to the scientist. Delivering this capability as a service would avail all researchers in the supported domains to visualize their data in an automated fashion—for example, common strategies for visualizing large-scale crystallography data could be offered to users of light sources generally. Over time, the common services could embody more advanced features, such as domain-specific feature extraction methods to prioritize regions of interest in simulation results.

Leadership-class HPC facilities enable science at the furthest reaches of today's computing technology. Future systems will push computing to the exascale and spur computational science to correspondingly greater discovery. To foster these advances among current projects and encourage new disciplines to leverage these resources, facilities will continue efforts to eliminate barriers to productively using their systems. As simulations become more precise and detailed, and supercomputers become more complex, it is imperative that we work actively with our users to achieve high performance with as little work as possible and integrate with their workflows to produce seamless, usable environments. We've outlined a handful of areas that could enable these advances. This will be no easy task, but the work is unquestionably justified by the promise of scientific results at the exascale. ■

Acknowledgments

The Argonne Leadership Computing Facility is a DOE Office of Science User Facility supported under contract DE-AC02-06CH11357. We acknowledge the assistance of Laura Wolf, Argonne National Laboratory, in editing the article.

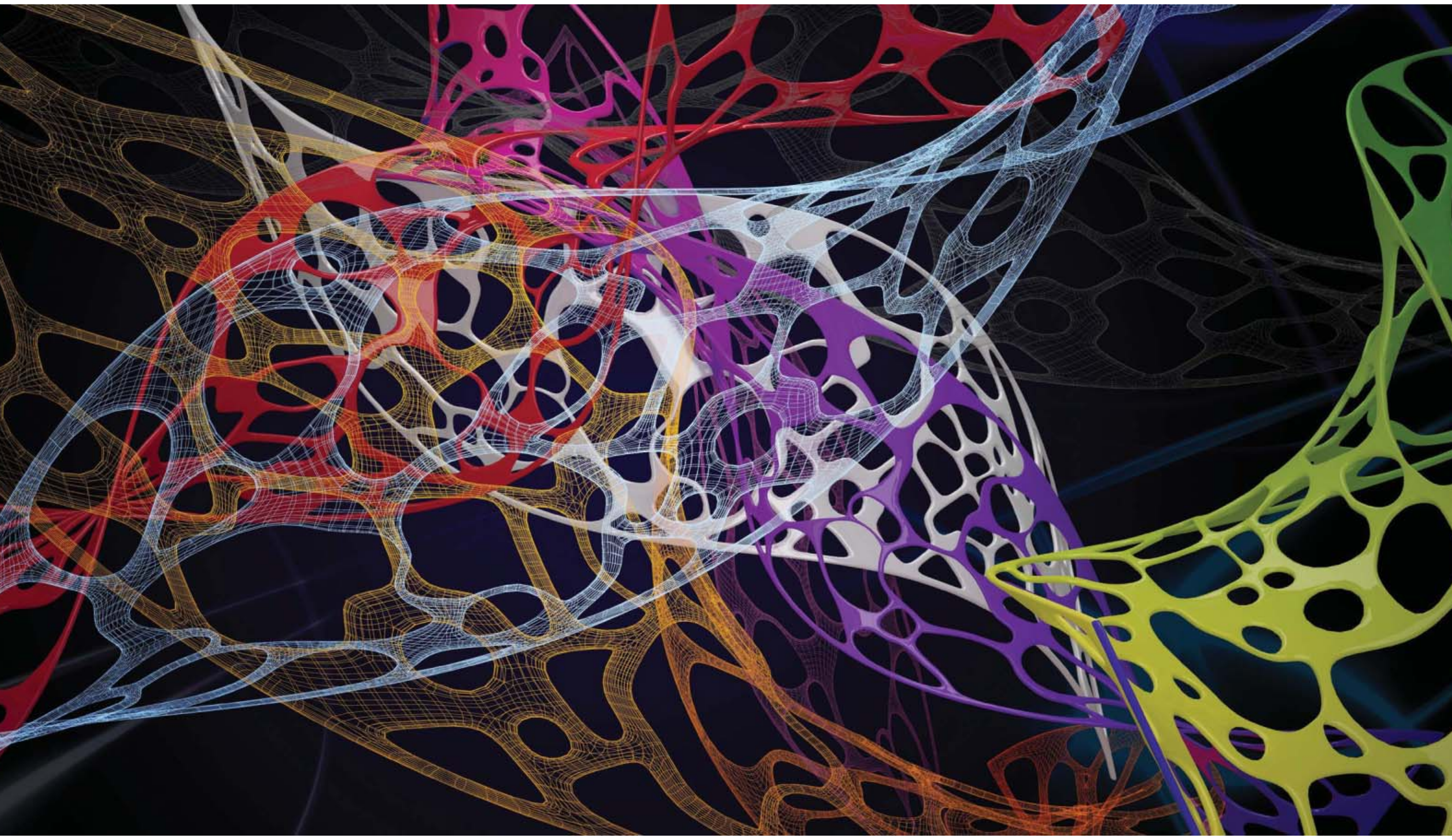
References

1. N. Wilkins-Diehr et al., "TeraGrid Science Gateways and Their Impact on Science," *Computer*, vol. 41, no. 11, 2008, pp. 32–41.
2. I. Foster, "Globus Online: Accelerating and Democratizing Science through Cloud-Based Services," *IEEE Internet Computing*, vol. 15, no. 3, 2011, pp. 70–73.
3. R. Hudson et al., "Experiences Using SMAASH to Manage Data-Intensive Simulations," *Proc. 20th Int'l Symp. High Performance Distributed Computing*, 2011, pp. 205–216.

Thomas D. Uram is a scientific applications developer at the Argonne Leadership Computing Facility at Argonne National Laboratory and a research staff member in the University of Chicago/Argonne Computation Institute. His interests include large-scale parallel programming in support of science, most recently in the context of particle physics. Contact him at turam@anl.gov.

Michael E. Papka directs the Argonne Leadership Computing Facility at Argonne National Laboratory, where he also serves a Deputy Associate Laboratory Director for Computing, Environment and Life Sciences. His research interests include the visualization and analysis of large data from simulation and experimental sources. Papka has a PhD in computer science from the University of Chicago. He's a senior fellow in the University of Chicago/Argonne Computation Institute and an associate professor of computer science at Northern Illinois University. Contact him at papka@anl.gov.

Editors: Cláudio T. Silva, csilva@nyu.edu | Daniel Weiskopf, weiskopf@visus.uni-stuttgart.de



Parallel Voronoi Computation for Physics-Based Simulations

Julio Toss and João Comba
Universidade Federal do Rio Grande do Sul (UFRGS)
Bruno Raffin
Université Grenoble Alpes, INRIA

Voronoi diagrams are fundamental data structures in computational geometry, with applications in such areas as physics-based simulations. For non-Euclidean distances, the Voronoi diagram must be performed over a grid-graph, where the edges encode the required distance information. The major bottleneck in this case is a shortest path algorithm that must be computed multiple times during the simulation.

We present a GPU algorithm for solving the shortest path problem from multiple sources using a generalized distance function. Our algorithm was designed to leverage the grid-based nature of the underlying graph that represents the deformable objects. Experimental results report speed-ups up to 65× over a current reference sequential method.

Voronoi Diagrams

The Voronoi diagram is a classical partitioning of a space into closest-point regions (see Figure 1). It has a vast application domain, usually employed for answering proximity queries such as finding nearest site, facility location, motion planning, and coverage in sensor networks.

The Voronoi diagram is also a key for the Sibson's natural neighbor interpolation (NNI) method.[1] NNI is a well-known method used for interpolating irregularly spaced

 1521-9615/16/$33.00 © 2016 IEEE Copublished by the IEEE CS and the AIP

Computing in SCIENCE & ENGINEERING Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page Qmags THE WORLD'S NEWSSTAND®

data, with applications in several different fields such as medical imaging, meteorological or geological modeling,[2] flow map reconstruction,[3] and scattered data visualization.[4] Natural neighbor-based interpolations have also been applied to the field of solid mechanics via the natural element method (NEM), which uses Sibson and non-Sibsonian (Laplace) interpolators to perform crack simulations.[5]

Voronoi-based interpolations have also been applied to meshless simulation of complex deformable bodies.[6] In that recent study, researchers computed a Voronoi tessellation on a non-Euclidean space by using a discrete distance map that encodes material-aware distances biased according to the local rigidity inside the simulated body (see Figure 2). Although the idea of a Voronoi partitioning of the space remains the same as in other classical applications, its computation here is fundamentally different. The distances aren't defined on the Euclidean space—instead, they rely on shortest paths computation over an implicit grid-graph. Computing this variant of Voronoi diagrams is therefore significantly more costly than computing the classical discrete Voronoi case. Using Sibson's NNI method on a graph space actually requires computing a shortest path tree for each interpolated value queried.

A lot of effort has already been dedicated to improving NNI's performance, particularly when interpolating on a discrete grid in the Euclidean space (discrete Sibson interpolation). A popular approach relies on GPU parallelization of the discrete Voronoi diagram, such as in the DEM construction.[2]

Early studies on parallelizing the discrete Voronoi diagram already exploited GPUs' parallel processing capabilities.[7] With the popularization of these architectures and the evolution of programming tools such as CUDA and OpenCL, other algorithms for Voronoi computation emerged to allow a better utilization of their computing power.[8,9]

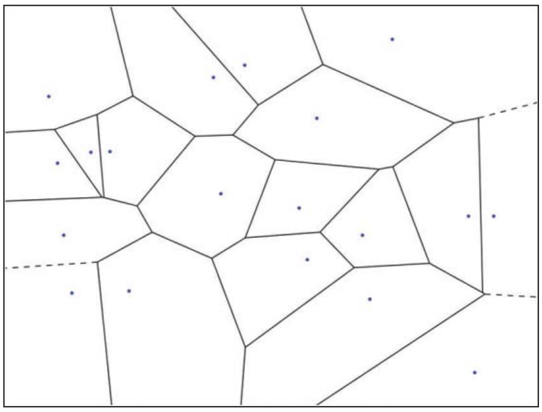
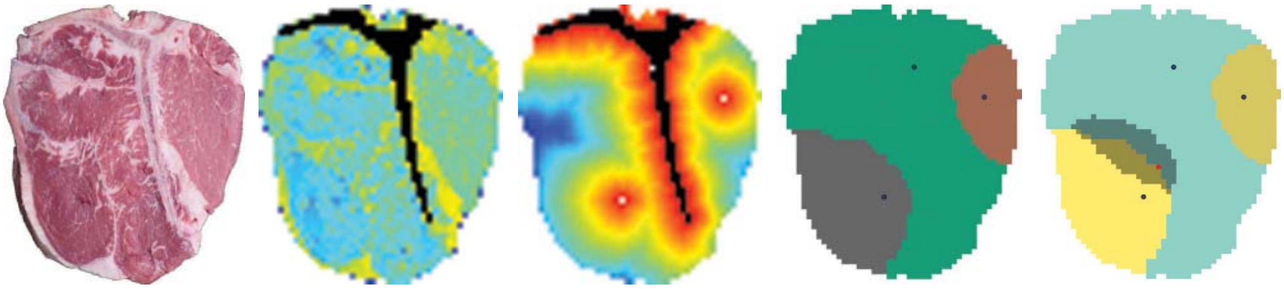


Figure 1. Voronoi diagram of a set of data points (in blue) in 2D Euclidean space. The space is partitioned into closest-point regions called Voronoi cells.

The graph variant of the Voronoi diagram (called the *graph Voronoi diagram*)[10] defines a vertex partitioning in a connected graph $G(V, E)$. Given a subset $S \subset V$ of source vertices, each vertex $v \in V$ is assigned to the partition P_i of the source vertex $s_i \in S$ with the shortest path distance. Applications of this kind of Voronoi diagram arise in several network problems and social data analysis, such as in community detection algorithms.[11]

Parallel solutions for computing the graph Voronoi have to deal with the shortest path



(a) (b) (c) (d) (e)

Figure 2. Use case example. (a) A T-bone steak contains a mixture of flexible meat, softer grease, and a rigid bone. As input, we take (b) the voxelized material map of stiffness values and the coordinates of the simulation nodes. (c) Distances to the nodes inside the object are biased according to the stiffness values and used to compute (d) the Voronoi diagram of the nodes. This diagram will be used to compute (e) the natural neighbors interpolation on the other voxels of the domain.

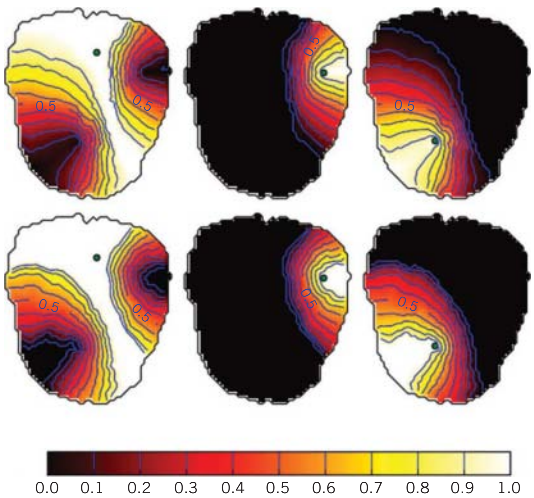


Figure 3. Shape functions' weights for three simulation nodes (from Figure 2) computed with two different interpolation methods: Sibson (top) and distance ratio (bottom). Weights are normalized starting at 1 in the node locations (Voronoi cell center) and decrease until vanishing outside of the support.

problem. Dijkstra's algorithm implemented with a priority queue provides an efficient solution to this problem but is inherently sequential with lots of synchronizations. For graphs with grid topology, our previous work created a first parallel algorithm to compute the Voronoi diagram.[12] The algorithm creates a distance map over a discrete image by computing the shortest paths from each pixel to the closest Voronoi seed (Figure 2d). The implementation is done on GPUs and takes advantage of grid connectivity to leverage parallelism.

Discrete Voronoi Diagrams in SOFA

The discrete Voronoi diagram is used in a real-time simulation framework called SOFA (www.sofa-framework.org), a modular and extendable architecture that allows researchers of different fields related to physics-based simulation to implement and compare their own algorithms.

Deformation Using Shape Functions

Researchers have proposed a method for simulating complex objects that are composed of mixed types of materials with different stiffness.[6] This method relies on meshless models using sparse samples to capture the displacements at the simulation nodes, which are then interpolated within the object using a novel *material-aware shape function*. This shape function uses a special distance metric scaled according to the local material rigidity of the simulated object (Figures 2b and 2c), a technique that lets us easily take into account material heterogeneity during simulation. The object's material properties, such as stiffness, are usually represented as a volumetric image of voxels containing property values. To determine the material-aware distance, the shortest path is computed over this 3D grid of voxels, where each voxel represents a vertex of a grid-graph with 26 neighbors, and the edges' weights are defined as a function of the stiffness of the adjacent voxels.

To understand the role of shape functions in numerical simulations consider the following steps:

1. The displacement of a deformable object is sampled at discrete locations called degrees of freedom (DoFs). Each DoF has a shape function associated that defines where and how it will influence other points in the object. The area of influence is often referred to as the *support* of the shape function (Figure 3).
2. The goal is then to interpolate these sampled displacements within the rest of the object.
3. The displacement at a given point is interpolated as a weighted sum of the node's displacements, where the weights are the values of the shape function for each DoF influencing this point.

Finally, we have the problem of defining how weights should be computed, which informs how shape functions from different DoFs will be blended in the rest of the domain. Voronoi diagrams have been used in the so-called *Voronoi shape functions* to compute these weights.

Voronoi-Based Interpolations

Consider the problem of finding neighbors in a set of nonuniform distributed data points. The Voronoi tessellation generated from this point provides us with the notion of *natural neighbors*, which are those data points whose Voronoi cells share a common frontier. Two interpolation schemes are based on this notion of neighborhood. The first one, the Sibson interpolation, is defined as a ratio of areas in 2D (volumes in 3D). It is computed by inserting the query point q in the initial Voronoi tessellation of sample points. The interpolating weight of each data point is then given by the ratio between the area stolen from the neighbor Voronoi cell and the area of the newly inserted Voronoi cell (Figure 2e). The second method, the Laplace (or non-Sibsonian) interpolation,[5,13] uses the same notion of natural neighbor, but it computes the ratio between segments in 2D (areas in 3D). Instead of taking the area of the neighbor cells, it uses the ratio between the length of the Voronoi frontier (line in 2D, facet in 3D) and the distance from q to its natural neighbor nodes.

Both of these NNI methods require the computation of a new Voronoi diagram for each query point $q \in Q$ added to the input diagram of the data samples. This results in $|Q|$ executions of the Voronoi diagram, where Q is the interpolation resolution desired.

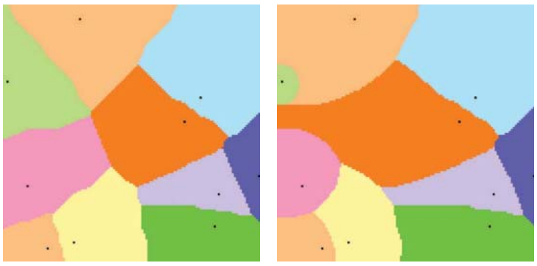
To reduce the number of Voronoi diagrams computed, an alternative interpolation method, called distance ratio,[6] only needs to compute a constant number of Voronoi diagrams per data sample $s \in S$, where $|S| \ll |Q|$. This method applies a particular scheme that computes the ratio between the distance from the point to the Voronoi border and the distance to the node (center of the Voronoi cell). Although less formal guarantees on the properties were presented for this interpolant, this algorithm is implemented on SOFA and shows good practical results, with the advantage of being more computationally efficient.

Parallel Voronoi Diagram Computation

As shown previously, many other algorithms, particularly those in physics-based simulations, build on Voronoi diagrams. In the case of real-time and interactive simulations, the computation of the Voronoi shape functions must meet strict performance requirements. Parallel processing is one popular strategy to meet this goal.

Parallel Voronoi computation on Euclidean distance has been extensively studied in previous work.[7–9] However, such approaches can't be directly applied in the physics simulation use cases described here, where distance measures are actually shortest paths computed on a graph.

Geodesic distance is closely related to the well-studied single-source shortest path (SSSP) problem from the graph theory domain. Parallel algorithms for solving the SSSP problem on general graphs have been proposed,[14–17] usually based either on the Dijkstra or Bellman-Ford algorithms. These algorithms assume a general graph without having any prior knowledge about its structure. They rely on a generic graph representation (such as adjacency matrix or list) where a vertex can have any number of edges with no



(a) (b)

Figure 4. Comparison of Voronoi diagrams generated with the same set of seeds on two different material maps: (a) uniform stiffness and (b) a stiffness gradient.

structured neighborhood. In physics-based simulation, the Voronoi-shape functions are built on top of a much more regular-structured graph, which allows us to use a less general but more optimized algorithm. Targeting efficiency, we presented[12] a parallel algorithm using GPUs to compute the graph Voronoi diagram on a voxelized 3D grid. This solution is well suited for the physics simulations used in SOFA as it exploits the grid topology that implicitly represents edges. The solution is based on parallel wavefront expansions starting at each Voronoi seed. By using the massive amount of parallel threads in a GPU, we can compute each Voronoi cell concurrently.

We present here some experimental results of speed-up of the parallel Voronoi computation. We conducted experiments on an Nvidia GPU GTX480 with 1.5 Gbytes of global memory and 15 multiprocessors with 32 cores each, totaling 480 CUDA cores. The speed-up presented is related to the base sequential version from SOFA and executed on an Intel Core i7 CPU model 930 with 4 cores running at 2.89 GHz and 12 Gbytes memory.

The benchmark consists of computing the Voronoi diagram on a 3D volume for a given set of randomly distributed data points. The dataset used varies in volume dimensions, distance map distribution, and number of seeds. Figure 4 shows an example of two Voronoi diagrams computed with the same dimensions and seeds but using different distance distribution. Figure 4b was computed from an image with a gradient of stiffness increasing from right to left, whereas Figure 4a has constant stiffness. These two distance distributions are referred to as gradient and constant on the bar plots of Figure 5. Note that the distance between

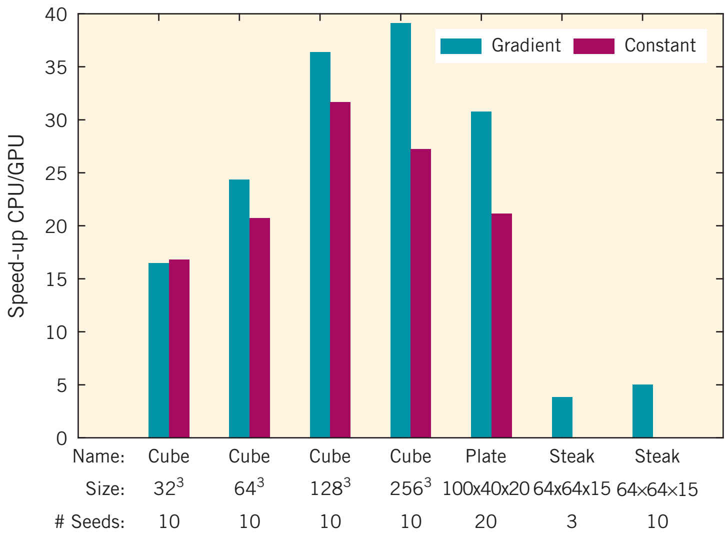


Figure 5. Speed-up for different input sizes. Gradient and constant topologies are presented for synthetic benchmarks only. The steak's topology corresponds to the dataset shown in the use case of Figure 2.

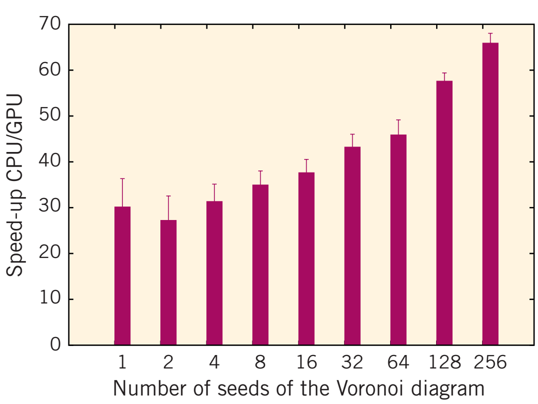


Figure 6. Average speed-up when increasing the number of seeds of the Voronoi diagram. The instance used has a volume size of 128^3 and a gradient material map.

two neighbor voxels is given by a function of the stiffness between them. As seen in Figure 4 the shape of the Voronoi diagram greatly depends on the stiffness properties of the material map of the object simulated.

In the parallel algorithm implemented, each CUDA thread processes a single voxel. With larger input volumes, more parallelism is exposed; therefore we note an increase in speed-up (Figure 5). At 256^3, the speed-up for the constant topology slightly decreases. We attribute this to an overhead of scheduling an excessive number of idle threads. This happens because with a volume of 256^3 and only 10 seeds, the Voronoi diagram becomes overly sparse.

The amount of parallel work available also increases with the number of seeds in the Voronoi diagram (Figure 6). When many Voronoi cells are being computed concurrently, more threads are active at the same time.

Parallel Natural Neighbor Interpolations

In the classical NNI, each queried point is inserted, one at a time, into the Voronoi diagram of initial data samples. For each seed added, the initial diagram is updated to generate the new Voronoi cell, which will then be used to compute the interpolation.

As seen in the experimental results, the number of seeds computed in the Voronoi diagram has a big impact on the amount of parallelism that will be exposed. Computing a single Voronoi cell in parallel reduces the possibilities of parallelization. This situation is even worse when we consider updating an existing Voronoi diagram with the addition of a new seed (the *query seed*). In this case, only a limited region (around the query seed) would be recomputed (Figure 2e). Performing NNI simply as a sequence of (parallel) Voronoi computations on the GPU doesn't pay off the overhead of memory transfer and thread scheduling inherent to this architecture.

One strategy to generate interpolated values over a grid would be to perform multiple queries concurrently in parallel. This approach has already been used[2] to generate an interpolation of a regular grid using the assumption that every sample has a limited radius of influence, thereby allowing the decomposition of the domain in independent blocks where queries can be answered in parallel batches. One study[4] proposed a more efficient implementation of Sibson's interpolation on raster images. The method avoids the explicit construction of a new Voronoi diagram for each query point, favoring instead a *Kd-tree* structure to find the closest seed to the current query point and using this distance as a radius of influence to increment the interpolation weight. Again, these techniques make assumptions that are valid for Euclidean space but not trivially generalized for geodesic (graph) distances.

Parallelization can still be implemented for NNI over graph spaces if we duplicate some data structures. More precisely, for each NNI query, we can copy the input Voronoi diagram of the data

Table 1. Computation time for parallel NNI queries. Average Sibson query corresponds to time spent for querying a whole batch.

Batch size	Average Sibson query (ms)	Total time (ms)
CPU 1 (seq)	1.003	81,057.226
GPU 1	9.745	779,657.756
GPU 10	12.413	99,309.125
GPU 100	29.537	23,629.870
GPU 150	34.807	18,587.130

sample and update it with the addition of a new seed at the coordinates of this query.

Table 1 shows the computation time spent for the parallel Sibson algorithm on an Nvidia Tesla K40 GPU. The interpolation is performed over a uniform grid of dimensions 100 × 40 × 20 (80,000 voxels) and 20 data points. The GPU algorithm performs batches of parallel NNI queries in sequence iteratively until the entire grid is computed. We show the average amount of time spent by each batch of parallel queries and the total time for interpolating the whole grid. Note that the parallel version manages to amortize the overhead when more than 10 NNI queries are done in parallel.

The parallel Voronoi implementation presented here has a valuable application in soft object simulation methods. It provides a performance solution to those in the physics-based simulation community who want to employ Voronoi shape functions in their meshless simulations.

Recent work has proposed using Voronoi shape functions on grids with extended connectivity, called *non-manifold grids*,[18] which would let us represent objects with more complex topologies in meshless frameworks. Possible extensions of this work will consider the application of our parallel algorithm in these new domains. ■

Acknowledgments

We thank the projects Capes/Cofecub 764/13 and CNPq 308851/2015-3 for financial support.

References

1. R. Sibson, "A Brief Description of Natural Neighbor Interpolation," *Interpreting Multivariate Data*, vol. 21, 1981, pp. 21–36.
2. A. Beutel, T. Mølhave, and P.K. Agarwal, "Natural Neighbor Interpolation Based Grid DEM Construction Using a GPU," *Proc. 18th Int'l Conf. Advances in Geographic Information Systems*, 2010, p. 172.
3. S.S. Barakat and X. Tricoche, "Adaptive Refinement of the Flow Map Using Sparse Samples," *IEEE Trans. Visualization and Computer Graphics*, vol. 19, no. 12, 2013, pp. 2753–2762.
4. S.W. Park et al., "Discrete Sibson Interpolation," *IEEE Trans. Visualization and Computer Graphics*, vol. 12, no. 2, 2006, pp. 243–252.
5. N. Sukumar, "Voronoi Cell Finite Difference Method for the Diffusion Operator on Arbitrary Unstructured Grids," *Int'l J. Numerical Methods in Eng.*, vol. 57, no. 1, 2003, pp. 1–34.
6. F. Faure et al., "Sparse Meshless Models of Complex Deformable Solids," *Proc. ACM SIGGRAPH*, 2011, p. 1.
7. K.E. Hoff III et al., "Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware," *Proc. 26th Ann. Conf. Computer Graphics and Interactive Techniques*, 1999, pp. 277–286.
8. O. Weber et al., "Parallel Algorithms for Approximation of Distance Maps on Parametric Surfaces," *ACM Trans. Graphics*, vol. 27, no. 4, 2008, pp. 1–16.
9. G. Rong and T.S. Tan, "Jump Flooding in GPU with Applications to Voronoi Diagram and Distance Transform," *Proc. Symp. Interactive 3D*, 2006, p. 109.
10. M. Erwig, "The Graph Voronoi Diagram with Applications," *Networks*, vol. 36, no. 3, 2000, pp. 156–163.
11. D. Deritei et al., "Community Detection by Graph Voronoi Diagrams," *New J. Physics*, vol. 16, no. 6, 2014, article no. 063007.
12. J. Toss, J.L.D. Comba, and B. Raffin, "Parallel Shortest Path Algorithm for Voronoi Diagrams with Generalized Distance Functions," *Proc. 27th Conf. Graphics, Patterns and Images*, 2014, pp. 212–219.
13. V.V. Belikov et al., "The Non-Sibsonian Interpolation: A New Method of Interpolation of the Values of a Function on an Arbitrary Set of Points," *Computational Mathematics and Mathematical Physics*, vol. 37, no. 1, 1997, pp. 9–15.
14. S. Kumar, A. Misra, and R.S. Tomar, "A Modified Parallel Approach to Single Source Shortest

Path Problem for Massively Dense Graphs Using CUDA," *Proc. 2nd Int'l Conf. Computer and Communication Tech.*, 2011, pp. 635–639.

15. P. Harish, V. Vineet, and P.J. Narayanan, "Large Graph Algorithms for Massively Multithreaded Architectures," tech. report IIIT/TR/2009/74, Int'l Inst. Information Tech. Hyderabad, 2009.
16. K. Madduri et al., "Parallel Shortest Path Algorithms for Solving Large-Scale Instances," *Proc. 9th DIMACS Implementation Challenge: The Shortest Path Problem*, 2006, pp. 1–39.
17. H. Ortega-Arranz et al., "A New GPU-Based Approach to the Shortest Path Problem," *Proc. Int'l Conf. High Performance Computing & Simulation*, 2013, pp. 505–511.
18. P.-L. Manteaux et al., "Interactive Detailed Cutting of Thin Sheets," *Proc. ACM SIGGRAPH Motion in Games*, 2015, pp. 1–8.

Julio Toss is a joint PhD student at Universidade Federal do Rio Grande do Sul and Université Grenoble Alpes. Contact him at jtoss@inf.ufrgs.br.

João Comba is an associate professor at Universidade Federal do Rio Grande do Sul. Contact him at comba@inf.ufrgs.br.

Bruno Raffin is a director of research at INRIA and the leader of the DataMove joint research team between INRIA and the Université Grenoble Alpes. Contact him at bruno.raffin@inria.fr.

What You Really Wanted

by Charles Day
American Institute of Physics
cday@aip.org

The annual number of people admitted to Britain's cinemas reached its peak of 1.6 million in 1946. By 1984, after three decades of near-monotonic decline, just 100,000 cinema tickets were sold in the country, despite there being 10 million more Britons and despite the release that year of an unusually high number of broadly appealing movies, among them *Ghostbusters*, *Indiana Jones and the Temple of Doom*, *Beverly Hills Cop*, *The Karate Kid*, *The NeverEnding Story*, and *The Terminator*.

Television caused the slump in movie attendance, but behind that simple attribution lies a subtlety. Movies weren't more popular in the 1940s than they were in the 1980s. Rather, if given the choice, people tend to prefer to watch diverting entertainment in the comfort and convenience of their own living rooms. Before the advent of TV as a mass medium in the 1950s, they lacked that choice.

Such hidden preferences are also found in our digital lives. In the 1990s, my wife, Jan, maintained a blog, *Life's a Banquet*, where she posted whimsical observations from her life for friends and family. Her sister, Sue, also had a blog, *Loads of Pink*, where she recorded the travails of mothering and laundering the clothes of her three young girls. Both Jan and Sue stopped blogging when they joined Facebook, which made posting and sharing updates easier than blogging.

Likewise, the decline of the domestic desktop computer and the concomitant rise of the tablet reflect the fact that when it comes to finding information and consuming entertainment, a physical keyboard is an unnecessary amenity—and always was.

What hidden preferences will future technologies reveal? Google, Tesla Motors, and other developers of driverless cars are betting that our love of personal vehicles arises from our wish to travel wherever and whenever we please, not from a love of controlling the speed and direction of a moving vehicle. The growing use of e-readers is freeing authors and publishers to produce books whose length is no longer constrained by the limits, upper and lower, of producing perfect-bound paper books.

But technology is limited in its ability to open choices. Advances in TV displays have narrowed the gap between watching, say, *Mission Impossible—Rogue Nation* at a cinema versus watching it at home. But those same advances fail to reproduce the full experience of attending a live performance of, say, *The Crucible*—and likely never will. Ticket sales of Broadway plays and musicals remain robust.

And as mobile phones, cars, and TVs become ever more sophisticated, some people, Jan included, are returning to old crafts. On the second-story porch of our house in Washington's Capitol Hill neighborhood live two colonies of honeybees, each inhabiting a hive of a type patented in 1852 by one Lorenzo Lorraine Langstroth. Meanwhile, her first carboy of homemade mead is fermenting in the basement. ■

Charles Day is *Physics Today*'s editor-in-chief. The views in this column are his own and not necessarily those of either *Physics Today* or its publisher, the American Institute of Physics.

 1521-9615/16/$33.00 © 2016 IEEE Copublished by the IEEE CS and the AIP May/June 2016

Computing in SCIENCE & ENGINEERING | Previous Page | Contents | Zoom in | Zoom out | Front Cover | Search Issue | Next Page