Using a Visual Programing Language to Interact with Visualizations of Electroencephalogram Signals

Chris S. Crawford, Marvin Andujar, France Jackson, Ivens Applyrs, Juan E. Gilbert
Computer and Information Science and Engineering Department, University of Florida

Abstract

As jobs begin to seek hybrid skill sets from employees more frequently, interdisciplinary learning will become more important. This paper presents a tool, which provides a novel interdisciplinary learning approach that merges math, computer science, and physiology to create engaging experiences for learners. The presented tool obtains neurophysiological measurements of electroencephalogram (EEG) signals using a Brain-Computer Interface (BCI). This data can be used to understand the cognitive and affective states such as fatigue, cognitive workload, engagement, and others. Using an interface equipped with a visual programing language (VPL) enables users to interact with and visualize EEG data obtained from the BCI. Interacting with a drag-and-drop VPL allows users to dynamically utilize data streams captured from the BCI. This approach enables the creation of creative visualizations that can be mapped to various mental states. Ultimately this work aims to investigate how BCI can be used with a VPL and data visualization tools to provide hands on educational activities.

Keywords

Brain-Computer Interface (BCI); Visual Programming Language (VPL); Electroencephalogram (EEG)

Introduction

Educators and researchers have used high-level programming languages for years to introduce K-12 students to programming[1]. Many of these platforms now utilize Visual Programming Languages (VPL) to teach children coding principles. VPLs have been used with many different types of technology ranging from robots[2] to mobile phone apps[3]. Although, researchers have explored integrating VPLs within various types of systems there has been limited investigation on using a VPL with a Brain-Computer Interface (BCI). BCIs are capable of measuring the central nervous system (CNS) activity that can be used as artificial output. This output is often used to replace, restore, enhance, supplement, or improve Natural CNS output[4]. This novel technology is often seen as exciting and has been featured in many science fiction movies. BCI could serve as an exciting technology for students, but there is currently no platform that utilizes the benefits of modern educational tools focused on integrating VPL with BCI. The integration of these technologies could result in educational concepts similar to the use of robotics to engage students with programming[5]. Current implementations that feature a VPL component with BCI are mostly geared towards traditional BCI researchers who perform work in the clinical domain. Although, they feature a visual programming component they can become complex and confusing causing it to be to challenging to work with. This issue presents an opportunity to create an application that utilizes a more accessible VPL with BCI. Currently many educational applications use the blockly block framework[6]. Previously this framework has served as an effective infrastructure for introducing coding to students, but it has yet to be implemented and tested with BCI.

Although, using robots has been shown to increase student's engagement[5] limitations could exist in certain scenarios[7]. One possible limitation is the lack of enough robots to support larger class sizes. To address these issues the integration of other novel technologies should be investigated. One potential technology that could be integrated is BCI. With BCI, student's neural activity can be stored and interacted with later without the physical device. Neural interfaces are also currently becoming more popular in science fiction movies.[8] Although this rise in popularity exists; there has been no research on approaches that use BCI with students actively in an educational setting. A system that integrates VPLs and BCIs could serve as a novel concept that engages students to become more interested in programming. This paper presents a system concept that utilizes offline BCI data, blockly, and three.js. This work explores how these components can be merged together to create dynamic brain-based virtual environments. After explaining the system setup this paper will discuss pros and cons of this type of system and provide potential next steps.

Brain-Computer Interface

Brain-Computer Interfaces have commonly been used in the medical setting. A recent emergence of consumer-grade BCI devices has prompted researchers to look into applications for BCI beyond the medical domain. Most research using BCI in an educational setting focuses on using BCI in a passive manner to collect affective information such as engagement while students complete assignments. Previously Andujar & Gilbert proposed a concept that utilizes EEG to enhance the reading experience[10]. This works mentions that dynamically introducing new content based on a user's affective state could enhance their reading experience. Andujar, et. al. also utilized BCI to compare levels of engagement between two learning techniques: video game and handout[13]. This consisted of measuring engagement levels of participants while they completed one of the two types of learning techniques. Results from this study showed a correlation between knowledge retention and engagement. Szafir and Mutlu also investigated a concept that used BCI in an educational setting[11]. This work used adaptive agents to monitor students' attention in real time using EEG. Using this approach the researchers were able to improve student's recall abilities. Szafir and Mutlu also researched the use of BCI with massive open online courses (MOOCs).[12] The concept presented in this work is based on adaptive content review, which determines topics students might benefit most on based on their attention levels while viewing lectures. This approach was shown to increase student's recall abilities over a baseline. Huang, et. al. presented an approach that provides contextual BCI training sessions. The system presented in this work consists of a dynamic projection of BCI training sessions when low levels of engagement are detected. Results from this study showed that the developed system could significantly improve engagement in terms of EEG based measurements and subjective measures from teachers. Although, there have been many studies that investigate engagement monitoring via BCI to enhance knowledge retention, there has been no work on using BCI actively with educational VPLs. Utilizing BCI with VPLs could lead to even higher levels of interest and retention of students computer science students.

Visual Programming Applications

Visual programming applications are commonly used to introduce children to programming. Alice, a 3D interactive animation environment features a VPL and has been used for years[25]. Using Alice, users can create virtual worlds and develop programs that animate objects in the virtual environment. It was previously reported to improve student's performance and retention in computer science[30]. Several researchers and educators have used Alice to teach coding concepts[26-29]. Over 137 million users have used the blockly[6] framework as part of code.org's *Hour of Code*[15]. This initiative is focused on expanding access to computer science and increasing participation. Currently over 5 million students are enrolled in courses that cover programming using this VPL. Over 11 million users have used a VPL via Scratch[16,17]. This application allows students to learn computer programming while developing projects that are personally meaningful. These projects often consist of animated stories or games created by students. Previous research has reported that scratch-based lessons were effective for teaching CS content and engaging students[23]. Scratch has also been used to address computer science retention issues[24]. One key goal of scratch is to encourage self-directed learning. MIT App Inventor features a VPL that allows users to create android apps[18]. Over 11 millions applications have been built using this application and it has almost 4 million registered users. Several high-school and college faculty have used successfully used App Inventor in their courses[19-22].

Although, many researchers and educators have used VPLs, no previous work exists that investigates the integration of BCI with a VPL. As the price of consumer-grade BCI devices continue to drop, the feasibility of using BCI as a novel tool in the classroom becomes more realistic. This work presents a conceptual system that uses the non-invasive Emotiv Epoc BCI device and the blockly framework to create cognitively controlled virtual environments.

Non-Invasive Emotiv BCI Apparatus

Figure 1: EEG Emotiv

The Emotiv EPOC non-invasive device (figure 1) is a wireless EEG data acquisition and processing device. It connects via Bluetooth to a computer. This device consists of 14 electrodes (AF3, AF4, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC8, F4, F8) and 2 references (P3/P4 locations) to obtain the EEG signals. These channels are based on the international 10-20 locations, which is the standard naming and positioning for EEG devices. The sampling method of the device is based on the sequential sampling with a sampling rate of 2048 Hz. This device was chosen among others for its portability and its adaptability as a wearable computing system. In addition, this device has been widely adapted for other research experiments.

Three.js

Table 1: List of some geometries supported in Three.js

Name	Description
Box Geometry	Quadrilateral primitive geometry class
Circle Geometry	Simple shape of Euclidean geometry
Cylinder Geometry	Class for generating cylinder geometries
Plane Geometry	Class for generating plane geometries
Sphere Geometry	Class for generating sphere geometries
Octahedron Geometry	Class for generating an octahedron geometry

Three.js is a JavaScript based 3D graphics library that provides a simple and convenient way to utilize Web-based Graphics library (WebGL). WebGL enables JavaScript to produce accelerated 3D graphics similar to OpenGL in the browser.[31] This is accomplished by leveraging the powers of HTML5 Canvas. With WebGL, Three.js can produce complex models via the HTML5 canvas. By abstracting away lower level WebGL API calls three.js makes WebGL development easier and more productive. The approach presented in this paper utilizes the box geometry class as explained in the following section. The Geometry base object included in Three.js provides the foundation for many models supporter in Three.js. It contains properties such as vertices, colors, and faces. Also included in this object is built-in functionality such as computing normal vectors and bounding boxes. Table 1 provides a list of a few objects that extend the base Geometry object. Each of these can be used to create various types of objects in the virtual environment.

Blockly

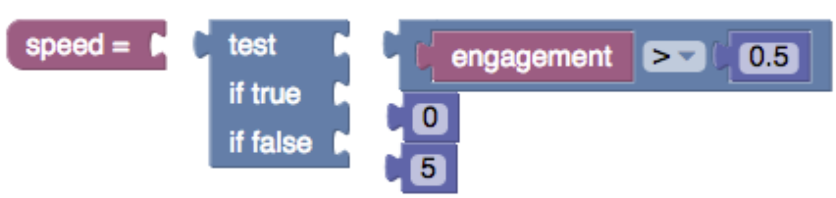

Figure 2: Blockly blocks

Blockly, developed by Google, is a graphical programming editor that uses blocks in a manner similar to scratch[17]. It is an open source visual block language development kit that can be used to create new block-based visual programming languages. Often it is used to address specific content created by researchers and educators. Due to its intuitive and easy to follow block interface blockly is usually more accessible to children than standard programming languages. It is written in JavaScript, which allows it to run in a web browser. With blockly, users can create scripts from block-shaped command templates. These templates are added to a script via a simple drag and drop interaction. Blocks featured in blockly have placeholders for variables as shown in figure 2. Scope is also accomplished in Blockly by nesting blocks physically inside an existing block. Programs created in Blockly can be exported as JavaScript, Python, or XML code. This feature is often used to help learners make connections between visual programming and the traditional programming languages. There has been much previous work that investigates how blockly can be used for various applications[32-35]. The visual nature of blockly adds to its overall usability which makes it a great framework for developing learning applications for novice programmers.

System Setup

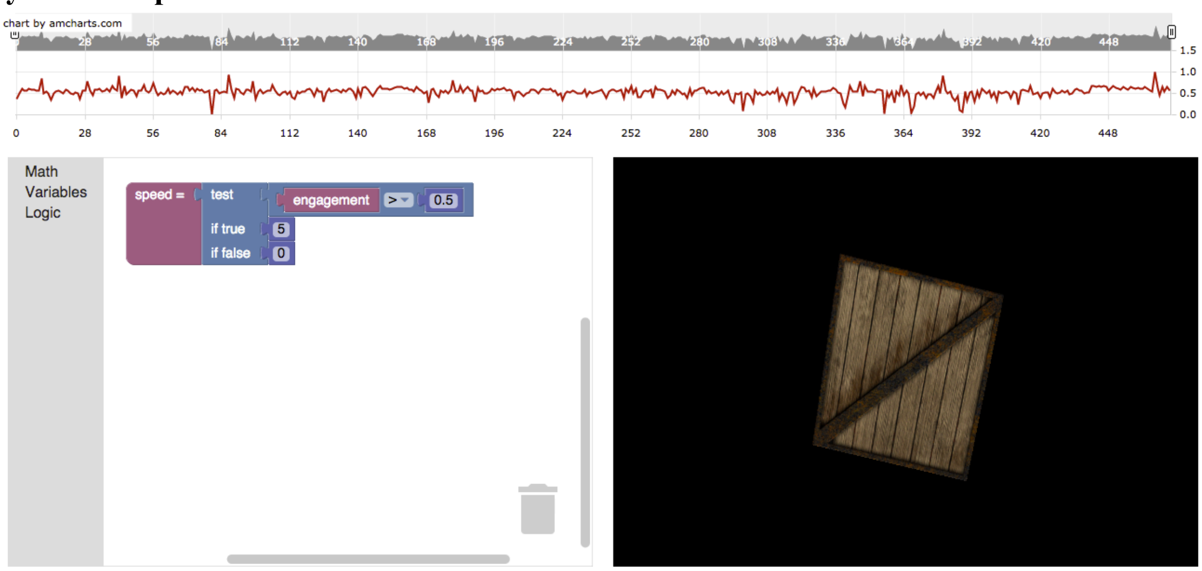

Figure 3: Web-based cognitively controlled virtual environment interface

$$E = \frac{\beta}{(\alpha + \theta)}$$

Figure 4: Engagement formula

Figure 3 shows a preliminary version of the presented programmable cognitive controlled virtual application. The main goal of this implementation is to show how engagement levels collected via a BCI device can be used in a programmable environment to interact with a virtual object. The top of the interface displays EEG data that reflects a user's engagement levels. The current version of the presented application uses prerecorded data, but it can be easily extended to record real-time EEG data. The pre-recorded data is stored in a comma separated value (CSV) file and loaded into the browser using JavaScript. This approach was taken initially to thoroughly test out the concept first prior to investigating the real-time implementation. The EEG data shown in figure 3 was captured via the non-invasive Emotiv Epoc BCI device. It was captured while a user was completing a cognitive manipulation task. The EEG data collection process begins by mounting the Emotiv on the user's head. Afterward the device must be adjusted until the best possible signal strength is acquired. Next users completed a baseline, which consisted of the user closing their eyes and clearing their minds. Users next were trained on how to move an object cognitively. A second shorter baseline was completed next. Finally, the user's EEG data was recorded while they completed a maze navigation task. Engagement was calculated using the formula shown in figure 4 which was first presented in work by Pope, et. al[36]. The engagement levels range from 0 to 1 indicated on the y-axis of the line graph at the top of figure 3. High levels of engagement fall closer to 1 and drops that fall closer to 0 indicate lower levels. The x-axis shown in the line graph section of figure 3 reflects time in seconds. It is important to note that even though this pre-recorded data was not directly connected to the manipulation of the virtual object, the fluctuation of engagement can still be used to control the virtual object programmatically.

As discussed earlier, most previous work has focused on using this kind of data specifically for analysis purposes only. The work presented in this paper uses engagement data as a form of input. These input values can be passed to blockly to create scripts that drive animations based on engagement levels. The script shown in figure 3 causes the wooden box on the right to rotate as levels of engagement increases. The speed blocks serves as the speed variable which is determine by the current level of engagement. The 'test' block shown in figure 3 provides logic that can be used to alter the value set to speed depending on the level of engagement. An additional logic block is used to perform a Boolean operation. This block, shown in figure 3, checks if the engagement level is greater than 0.5. Using this test logic block a value of 5 is assigned to the speed variable if this evaluates to true. In cases where the engagement level is less than 0.5 the speed variable is set to 0 causing the virtual object on the right to stop rotating.

Conclusion

This paper presents a novel concept of using a VPL and BCI to develop a dynamic brain-based virtual environment. Previous literature has shown that VPL have been successful in increasing knowledge retention and interest in STEM, but there has been no investigation of the influence of incorporating BCI. BCI has begun to become more commonly depicted in media. Also consumer-grade BCI devices are becoming more readily available. Although, this implementation presents a promising application future user studies are needed to evaluate how it influences student's performance. Also, it is important to analyze how students perceive this novel technology. Going forward it will be vital to investigate various ways to utilizing BCI signals, VPLs and visualizations within an educational context.

Future Work

The work presented in this paper serves as a preliminary investigation of utilizing a VPL with BCI data. Further work is needed to investigate additional forms of programmable interactions. As mentioned in the Three.js section, many geometry classes exist that can be used to develop new and more intriguing virtual environments. Although, engagement was used in this implementation, affective information such as frustration, meditation, boredom and others could also be investigated. Further work will also consist of implementing real-time EEG data collection. With this approach users could actively control objects in the virtual environment. Creating lessons with this system that integrates math and physiology educational content could also be added to provide interdisciplinary learning. User studies evaluating this tool as a way to encourage knowledge retention and interest in STEM will also be conducted as future work.

References

1 Kelleher, Caitlin and Randy Pauusch, "Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers", ACM Computing Surveys (CSUR), 2005, 83-137.

2 Sonya Alexandrova, Zachary Tatlock, and Maya Cakmak, "RoboFlow: A flow-based visual programming language for mobile manipulation tasks" Robotics and Automation (ICRA), IEEE International Conference on. IEEE, 2015, 5537-5544.

3 Spertus, Ellen, Mark Chang, Paul Gestwicki, and David Wolber, "Novel approaches to CS 0 with app inventor for android", In Proceedings of the 41st ACM technical symposium on Computer science education, 2010, 325-326.

4 Wolpaw, Jonathan and Elizabeth Wolpaw, Brain-Computer Interfaces: Something New Under the Sun, In: Brain-Computer Interfaces Principles and Practice, New York, New York: Oxford University Press, 2012, 3-12.

5 Tsang, Ethan, Collette Gavan, and Mark Anderson, "The practical application of LEGO® MINDSTORMS® robotics kits: does it enhance undergraduate computing students' engagement in learning the Java programming language?", Proceedings of the 15th Annual Conference on Information technology education. ACM, 2014. 121-126.

6 Fraser, Neil, Blockly website, https://code.google.com/p/blockly, accessed Nov 10, 2015.

7 Fagin, Barry S. and Laurence Merkle. "Quantitative analysis of the effects of robots on introductory Computer Science education." Journal on Educational Resources in Computing (JERIC) 2.4 (2002)

8 Schmitz, Michael, Christoph Endres, and Andreas Butz, "A survey of human-computer interaction design in science fiction movies", Proceedings of the 2nd international conference on Intelligent Technologies for interactive entertainment. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

9 Cabello, Ricardo, 2010. Three.js. http://threejs.org/.

10 Andujar, Marvin, and Juan E. Gilbert, "Let's learn!: enhancing user's engagement levels through passive brain- computer interfaces", CHI'13 Extended Abstracts on Human Factors in Computing Systems. ACM, 2013, 703-708.

11 Szafir, Daniel and Bilge Mutlu, "Pay attention!: designing adaptive agents that monitor and improve user engagement." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2012, 11-20.

12 Szafir, Daniel and Bilge Mutlu, "ARTFul: adaptive review technology for flipped learning", In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2013, 1001-1010.

13 Andujar, Marvin, Josh I. Ekandem, Juan E. Gilbert, and Patricia Morreale, "Evaluating engagement physiologically and knowledge retention subjectively through two different learning techniques", Human-Computer Interaction. Applications and Services, Springer Berlin Heidelberg. 335-342.

14 Huang, J., Chun Yu, Yuntao Wang, Yuhang Zhao, Siqi Liu, Chou Mo, Jie Liu, and Lie Zhang, "FOCUS: enhancing children's engagement in reading by using contextual BCI training sessions", In Proceedings of the 32nd annual ACM conference on Human factors in computing systems, 2014, 1905-1908.

15 Code.org, Hour of Code website, http://code.org/learn, accessed Nov 10, 2015.

16 Scratch project, MIT Lifelong Kindergarten Group, http://scratch.mit.edu/, accessed Nov 10, 2015.

17 Maloney, John, Mitchel Resnick, Natalie Rusk, Brian Silverman, and Evelyn Eastmond, "The Scratch programming language and environment," ACM Transactions on Computing Education, vol. 10, no. 4, Nov. 2010.

18 App Inventor website, http://appinventor.mit.edu, accessed Nov 10, 2015.

19 Gray, Jeff, Hal Abelson, David Wolber, and Michelle Friend. "Teaching CS principles with App Inventor", In Proceedings of the 50th Annual Southeast Regional Conference (ACM-SE '12). ACM, New York, NY, USA, 2012, 405-406.

20 Wolber, David, "App inventor and real-world motivation", In Proceedings of the 42nd ACM technical symposium on Computer science education (SIGCSE '11). ACM, New York, NY, USA, 2011, 601-606.

21 Ericson, Barbara and Tom McKlin. "Effective and sustainable computing summer camps", In Proceedings of the 43rd ACM technical symposium on Computer Science Education (SIGCSE '12). ACM, New York, NY, USA, 2012, 289-294.

22 Roy, Krishnendu, "App inventor for android: report from a summer camp", In Proceedings of the 43rd ACM technical symposium on Computer Science Education, ACM, 2012, 283-288.

23 Franklin, Diana, Phillip Conrad, Bryce Boe, Katy Nilsen, Charlotte Hill, Michelle Len, … Robert Waite, "Assessment of computer science learning in a scratch-based outreach program", In Proceeding of the 44th ACM technical symposium on Computer science education, 2013, 371-376.

24 Rizvi, Mona and Thorna Humphries, "A Scratch-based CS0 course for at-risk computer science majors", In Frontiers in Education Conference (FIE), 2012, 1-5.

25 Cooper, Stephen, Wanda Dann, and Randy Pausch, "Alice: a 3-D tool for introductory programming concepts. In Journal of Computing Sciences in Colleges", Consortium for Computing Sciences in Colleges. 2000, Vol. 15, No. 5, 107-116.

26 Wellman, Briana Lowe, Monica Anderson, and Susan V. Vrbsky, "PREOP as a tool to increase student retention in CS", Journal of Computing Sciences in Colleges, 25(2), 2009, 167-175.

27 Rodger, Susan H., Jenna Hayes, Gaetjens Lezin, Henry Qin, Deborah Nelson, Ruth Tucker, ... and Don Slater, "Engaging middle school teachers and students with alice in a diverse set of subjects", In ACM SIGCSE Bulletin Vol. 41, No. 1, 2009, 271-275.

28 Werner, Linda, Shannon Campe, and Jill Denner, "Children learning computer science concepts via Alice game-programming", In Proceedings of the 43rd ACM technical symposium on Computer Science Education, 2012, 427-432

29 Powers, Kris, Stacey Ecott, and Leanne Hirshfield M., "Through the looking glass: teaching CS0 with Alice", In ACM SIGCSE Bulletin Vol. 39, No. 1, 2007, 213-217.

30 Moskal, Barbara, Deborah Lurie, and Stephen Cooper, "Evaluating the effectiveness of a new instructional approach", ACM SIGCSE Bulletin, 36(1), 2004, 75-79.

31 Marrin, C. 2011. WebGL Specification. Khronos WebGL Working Group.

32 Trower, Jake, and Jeff Gray, "Blockly Language Creation and Applications: Visual Programming for Media Computation and Bluetooth Robotics Control", In Proceedings of the 46th ACM Technical Symposium on Computer Science Education, 2015, 5-5.

33 Bojda, Beáta, Katalin Bubnó, Benedek Nagy, and Viktor Takács, A graphical representation of Boolean logic. In Diagrammatic Representation and Inference, Springer Berlin Heidelberg. 2014, 228-230

34 Marron, Assaf, Gera Weiss, and Guy Wiener, "A decentralized approach for programming interactive applications with javascript and blockly", In Proceedings of the 2nd edition on Programming systems, languages and applications based on actors, agents, and decentralized control abstraction, 2012. 59-70.

35 Ashrov, Adiel, Assaf Marron, Gera Weiss, and Guy Wiener, "A use-case for behavioral programming: an architecture in JavaScript and Blockly for interactive applications with cross-cutting scenarios", Science of Computer Programming, 2015, 268-292.

36 Pope, Alan T., Edward Bogart H., and Debbie Bartolome S., "Biocybernetic system evaluates indices of operator engagement in automated task", Biological Psychology 40, 1-2, 1995, 187–195.

Chris S. Crawford

Chris Crawford is a PhD student in Human-Centered Computing at the University of Florida. His research focus is Brain-Robot Interaction, which consists of investigating ways electroencephalogram (EEG) can be used to enhance Human-Robot Interaction. He holds a B.S. in Computer Science.

Marvin Andujar

Marvin Andujar is a PhD student in Human-Centered Computing and a National Science Foundation Graduate Research Fellow (NSFGRF) at the University of Florida. His research of interests includes Affective Brain-Computer Interfaces, Personal Informatics/Quantified-Self, and Ubiquitous Computing. He holds a B.S. in Computer Science and a B.A. in Mathematical Sciences.

France Jackson

France Jackson is a PhD student in Human-Centered Computing at the University of Florida. She is a user experience researcher and designer. Her research interests center around connecting technology and popular culture such as Fashion and Technology. Currently, she focuses on wearable computing and Brain-Computer Interfaces. She holds a B.S. and M.S. in Industrial Engineering from Clemson University.

Ivens Applyrs

Ivens Telandcy Applyrs is a fourth year Computer Engineering Major at University Florida, Gainesville, Fl 32608. He is a former regional leader of the National Society of Black Engineers & former director of leadership in Student Government. He is a current undergraduate researcher with the Florida Georgia Louis Stokes Alliance for Minority Participtaion (FGLSAMP) program. His research interests include brain computer interactions and autonomous trafficking.

Juan E. Gilbert

Dr. Juan E. Gilbert is the Chair of the Computer & Information Science & Engineering Department at the University of Florida where he leads the Human Experience Research Lab. He is also a Fellow of the American Association of the Advancement of Science, National Associate of the National Research Council of the National Academies, an ACM Distinguished Scientist and a Senior Member of the IEEE. He received his B.S. in Systems Analysis from Miami University (OH) in 1991 and his M.S. and Ph.D. degrees from the University of Cincinnati in 1995, 2000, respectively.