1 Vision

The continued rapid evolution of computation and its impact on the world creates a new challenge for designing the learning experiences about computation at the university level. Deeply informed skills and knowledge about computation are needed in all STEM-H disciplines, and new requirements are emerging in non-traditional fields, like "digital humanities" [9, 5]. The growing awareness of "computational thinking" as a 21st century competency requires computation to be positioned in a university's general education curriculum [33]. Weaving together the curriculum, pedagogy, and tools that engage learners with starkly different dispositions and expectations about their learning of computation is a critical, on-going challenge [32, 8].

We propose to address this challenge by crafting *authentic experiences* that engage multi-disciplinary cohorts of learners with computation, both in and across contexts. Their efficacy will be evaluated by rigorous assessment. These authentic experiences are contextualized around "Big Data" streams that open exciting possibilities for learning through their importance and availability [24]. Techniques for handling Big Data have become crucial to science, business, and policy making at all levels of government [1]. Frequently in the news, Big Data provides a novel, interesting theme around which engaged learning can occur [20]. The characteristics of Big Data match the requirements of the desired learning experiences, underpinning authentic experiences because the data derives from real phenomenon (e.g., geophysical events or social media) [10, 31], is from definitive sources (e.g., US Geological Survey or Reddit), and is of genuine scale and complexity (not a "toy" version) [2]. Further, Big Data germane to a broad array of disciplines is available, making it relevant to the learner's concerns. However, technical difficulties have kept Big Data topics out of the introductory learning experiences [24]. This omission deprives most students of the chance to gain an appreciation for the possibilities and risks associated with Big Data, and removes a novel and interesting class of problems from the set of learning opportunities.

We seek EAGER funding to take advantage of an immediate and rare opportunity to infuse computational thinking into the general education curriculum at Virginia Tech [26]. [WHAT REMARKABLE OPPORTUNITY? YOU NEVER SAY WHAT IS GOING ON. RIGHT HERE IS WHERE WE NEED SEVERAL SENTENCES THAT EXPLAIN WHAT THE COURSE IS ABOUT, AND THE SITUATION AT VT THAT BRINGS THIS OPPORTUNITY ABOUT.]

This remarkable opportunity in Fall 2014 to develop and evaluate elements of curriculum, pedagogy, and technology: (1) serves as a model for other universities grappling with the challenge of providing computation for all STEM-H students or computational thinking for all students, (2) provides an on-ramp for developing minor courses of study in computer science, and (3) contributes to evolving discussions of theories on how students are best introduced to the computer science discipline and to topics specifically within Big Data. EAGER support for this effort will dramatically enhance the quality and degree of impact that we will be able to achieve.

We will positively impact Computer Science education by creating carefully scaffolded technology for manipulating web-available Big Data streams. This scaffolding builds on the success of the RealTimeWeb project, our framework for rapidly building real-time web-data centered assignments in introductory courses [3]. The RealTimeWeb tool chain allows students to work with challenging, but motivating, dynamic and/or large-scale data. Instructors can quickly and seamlessly incorporate real-time data streams into new learning experiences. This framework has been deployed in multiple courses at Virginia Tech and the University of Delaware [4]. Adding technical scaffolding to empower students to work with Big Data will leverage and enhance techniques that we have successfully applied to real-time data.

2 Approach

2.1 Pedagogy

Our approach is founded on authentic problem-based learning enabled by Big Data and a novel multi-disciplinary cohort model for courses serving a variety of majors. Authentic problem-based learning comes from Situated Learning Theory, originally proposed by Lave and Wenger, which argues that learning normally occurs as a function of the activity, context, and culture in which it is situated [23]. Therefore, tasks in the learning environment should parallel real-world tasks in order to maximize their authenticity. Authenticity has several interrelated meanings: (1) the problems are of realistic sizes, (2), the activity builds on students' individual interests and meaningfully relate to the real-world to promote cognitive engagement with the task at hand, and (3) the concepts and techniques are applicable to tasks the students will encounter subsequently in their field of study. Contextualization is a key factor in these settings, where learning is driven by the problem being solved, rather than the tools available [15]. Therefore, the problem being solved should lead directly to the concepts being taught and the tools employed. Our project builds heavily on this educational theory.

High levels of authenticity are made possible through a Big Data framework that is scaffolded – an artificially simplified design to support novices as they develop expertise. Scaffolding is used to remove certain complexities until a student is prepared to handle them, or in order to reduce the cognitive load for a learner. The scaffolding we propose is less necessary for high-level courses, but it is critical for the introductory level. As students progress through the curriculum, the scaffolding is *faded*, that is, parts are incrementally removed to give students a fuller understanding of the underlying systems [7].

We envision that courses would involve substantial projects chosen to reflect real-world concerns and involve issues of complexity and scale. Course material is delivered on-demand and in service of the students' project work. To support the projects, part of our work is to construct a rich catalog of resources that can be manipulated with the software tools used in the course. The student's project work focuses on "learning in context" because the student will choose projects that relate to his or her disciplinary area. The student engages in active learning because, as an individual, a student self-directs the selection, exploration, and completion of a project relevant to their major field of study. While minimum requirements will be set that all projects must meet, the student will have wide latitude to self-direct their project work.

Our approach also uses a novel *multi-disciplinary cohort* model for courses that are not strictly for CS majors. The multidisciplinary cohort is a form of peer learning [16]. For each project the students are organized into cohorts of 4-6 students so that each cohort has the greatest degree of diversity among the students' actual or intended major fields of study. The activities of the cohort support "learning across contexts" because the peer interaction will provide each student a perspective of how computational thinking applies in the other students' disciplines. As a member of the cohort a student is responsible for presentation, interaction, and support. These activities are:

- Presentation: describing to the other cohort members the significance of the project they have selected based on what the student has learned from their individual exploration.
- Interaction: asking questions and providing feedback about the projects of other cohort members. Through this interaction the student strives to gain some insight into the common

computational techniques that are used across disciplines.

- Support: helping other members of the cohort with the mechanics of the tools and frameworks that are common across projects.

Part of the planned assessment will evaluate the efficacy of the multi-disciplinary cohort model and give meaningful insight into intra-cohort dynamics.

2.2 Technology

Our approach builds on the success of the RealTimeWeb project (RTW) to create a generalized, adaptable framework that can be instantiated for any Big Data source. RTW contains a gallery of data streams drawn from publicly available web sources. For example, a current data source is large-scale weather data made available by the Weather Service. The essential service provided by RTW is to simplify what a student must code to access the large and complex data steams available on the web. The data streams may be accessed from a live feed in real time, or may be read from cached data files.

Central to RTW are client libraries of pre-built functions, one library for each data stream, which are used by the student's code. The client library includes both the code and the documentation. These libraries allow the student to focus on the algorithms that they must construct to process the data stream according to their project goals. While RTW has been used successfully in introductory computer science courses, more work must be done in preparation for the proposed non-major's course.

To rapidly convert a third-party, massive dataset into a practical educational resource, the proposed deliverable will use standardized Software Engineering methodologies and techniques. To that end, we will build on our existing extensible RTW software framework. Libraries and frameworks are key building blocks for software development – they provide predefined, extensible software components. Libraries and frameworks are similar, but subtly different in an important way. While both libraries and frameworks offer reusable elements of functionality, frameworks also define an architecture that developers can work within to build their software. A typical Object-Oriented software framework is composed of a collection of classes that define some central features. To create application-specific features, the developer can then add custom classes that extend abstract base classes. By developing our project as a framework, instructors who want to work with a new data source can simply extend our architecture. They are free to select among the features and functionalities that are offered through our system.

2.3 Curriculum

Our initial version of the non-major's course (in Fall 2014) will introduce students (via a scaffolded approach) to key programming concepts that allow them to complete significant Big Data projects. A combination of Blockly [25] and Python will be used. Blockly is used to introduce the fundamentals of algorithms. With Blockly the syntactic detail of programming languages are set aside in favor of a visual representation of the algorithmic structure. Thus, students can gain confidence (mastery experience) in their ability to construct algorithms before having to cope with the extra detail of the syntax of the programming language.

Blockly provides a collection of "blocks" which are shaped so that they can be assembled only in ways that "make sense" syntactically. Blocks have slots into which other blocks can be inserted to form complex algorithms. Algorithms can be executed to see their effects. Simple, limited ability for input/output is provided. Blockly is extensible so that blocks with application-specific semantics can be defined. An interesting aspect of Blockly is that the Python code for a Blockly algorithm can be seen. This makes the transition from Blockly to Python a more progressive step for learners.

To provide learning materials for both Blockly and Python we have been in conversation with Runestone regarding their interactive on-line Python book [21]. It is technically feasible to replace the interactive Python examples with equivalent examples using Blockly. Furthermore, with sufficient development support, it would be attractive to replace the existing interactive examples with ones which were specific to the data streams and Python structures being used in conjunction with RTW, and to develop new Blockly blocks specific to RTW data streams as additional scaffolding.

3 Proposed Work

In the short-term we propose to develop, offer (in Fall 2014), and rigorously evaluate a computational thinking course using the Big Data approach. We will also develop resources for using the Big Data approach in our introductory computer science course. IRB approvals will be secured as appropriate for our assessment activities. The work is divided into three primary tasks.

3.1 Develop curriculum resources

This development has a number of components. First, we will expand the available data sources and project ideas that are applicable for the broadest set of majors. Since this goal can only be partially realized in the short time available for development, we will focus our efforts on the disciplinary areas represented by students pre-enrolled in the Fall 2014 class. We will leverage Virginia Tech programs and rewards (e.g., summer programs, equipment credits, faculty recognitions) to incentivize the participation of faculty representing diverse majors in the identification or data streams and the development of project ideas. Second, we will specifically engage faculty in engineering education to identify data sources that are particularly appropriate for engineering students, and that may be leveraged by Virginia Tech Department of Engineering Education for use in the introductory courses that all engineering majors take. Third, we will develop projects for introductory computer science courses. The targetted courses are our CS1 and CS2 courses (Dr. Tilevich) and CS3 (Dr. Shaffer). We will be able in Fall 2014 to use and evaluate some of these projects. Fourth, we will prototype interactive learning materials. We will develop a variant of the Runestone interactive Python "book" to incorporate examples in Blockly and develop specialized Blockly elements for accessing RTW functionality. The goal is to allow for early experience with algorithms that manipulate "real" Big Data even before the syntax of Python has been mastered.

3.2 Expand and Enhance RTW

The technical work of providing the framework for the data sources identified in the work described above involves several key steps. First, we will develop the code and documentation of each newly identified data source. This work includes the development for each data source of the relevant components of the RTW framework (dataset selection, hardware attunement, data massage, sampling, network support). Testing of these components must be done to ensure reliable operation. Second, we will enhance the RTW framework to include linkages to visualization and statistical services. We anticipate that meaningful student projects will involve services typically encountered in big data applications. Two key services are those of visualization (display a topographic map of the location of the most severe earthquakes over the past year) and statistical analysis (is the occurrence of earthquakes correlated with some other phenomenon?).

3.3 Develop, Apply and Analyze Initial Assessments

Asessment will include three components: assessment of student motivation, assessment of mastery of learning objectives, and assessment of the inter-disciplinary cohort model. We will develop the assessment mechanisms during Summer 2014, and apply them to the initial offering of the Computational Thinking course in Fall 2014, with analysis as described next.

First, we will develop assessment of student motivation using the MUSIC model [17]. We hypothesize that a Big Data orientation will engage students in the projects and motivate them to work harder and learn skills more deeply. Relevancy and interest are known contributing factors for engaging students. In order to measure these and other factors of motivation, we will use the MUSIC model of academic motivation as our theoretical foundation. The MUSIC model is a well-supported theory that identifies five key constructs in motivating students:

- eMpowerment: The amount of control that a student feels that they have over an assignment.
- Usefulness: The expectation of the student that the material they are learning will be valuable to their short and long term goals.
- Success: The student's belief in their own ability to complete an assignment.
- Interest: The student's perception of how the assignment appeals to situational or long-term interests.
- Caring: The students perception of their professor's and classmates attitudes toward them.

The MUSIC model has compelling evidence supporting the validity of its associated instrument, the MUSIC Model of Academic Motivation Inventory [18]. This instrument will be deployed before, during, and after courses that use our resources, alongside qualitative questions that will isolate students' perceptions of our curriculum in light of these five elements. These data will indicate whether psychological constructs related to motivation can be affected through our assignments.

Second, we will leverage Virginia Tech resources to perform quantitative assessment related to achievement of course learning objectives. We have had initial discussions with Dr. Kate McConnell, the Assistant Director for the Office of Assessment and Evaluation. She has agreed to play a key role in developing and analyzing the assessments of how well students achieved key course learning objectives.

Third, we will conduct qualitative assessment of the inter-disciplinary cohort model. We hypothesize that a student cohort (i.e., one of our project groups) provides a valuable form of peer learning. Observations of cohort meetings will be used to gather data about the interactions within the cohort. These observations will be analyzed to isolate key factors across cohorts that appear to strengthen the peer learning effect. These analyses will help guide future uses of the cohort model. An Interim Protocol Request has been submitted for review to the Virginia Tech IRB.

4 Broader Impacts

By using Big Data to create engaging and authentic learning experiences, this project will have significant positive impacts on a broad population of students, especially populations that are traditionally more distant from computing or find computing less approachable. Our approach is informed by prior research showing that women are more likely to study and excel in Computer Science when content is contextualized and proven useful for solving real-world problems [11, 6]. Similarly, non-major students will benefit from the realistic assignments that directly relate to their intended line of work, further increasing their competence within computing, motivating additional study of computer science, and fostering a sense of identity within the computing community. Finally, appropriately redesigning programming projects to involve interesting, contextualized Big Datasets is likely to improve recruitment, retention, and engagement of students within Computer Science.

All theoretical and evidence-based conclusions researched during this project will be disseminated through publications and workshops at relevant international and national conferences. There are a large number of appropriate, recognized conferences that cover topics in Computer Science Education, including SIGCSE, ITiCSE, FIE, CCSC and ASEE. Relevant journals include ACM Transactions on Computing Education, Elsevier's Computers & Education: An International Journal, IEEE Transactions on Education, and Taylor & Francis' Computer Science Education. Tutorials, API references, and technical reports (regarding software analysis, implemented libraries, and infrastructure development) will all be made available through our website and open source repositories. In order to advertise and promote our toolchain, we will publish through highly visible communication channels within the CS Education community, such as the SIGCSE listserv.

Finally, this timely EAGER project will serve as the stepping stone to a full proposal directed at developing a deeper understanding on the impact of learning through a longitudinal study, the maturation of the scaffolding technologies, and the leadership in forming an international community of educators and researchers dedicated to developing and extending these Big Data pedagogical resources.

5 Results from Prior NSF Funding

NSF TUES Phase I Project (DUE-1139861) *Integrating the eTextbook: Truly Interactive Textbooks for Computer Science Education.* PIs: C.A. Shaffer, T. Simin Hall, T. Naps, R. Baraniuk. $200,000, 07/2012 – 06/2014. This award supported the initial phases of the OpenDSA eTextbook project, and an active collaboration involving Virginia Tech, U of Wisconsin–Oshkosh, and Aalto University (Helsinki), among others. Publications related to this work so far include [27, 29, 28, 12, 22, 19, 14, 13]. **Broader Impacts** include dissemination of AV artifacts and DSA courseware to a broad range of CS students, and made them available through the NSF NSDL.

NSF CE21 Planning Grant (CNS-1132227) *Planning Grant: Integrating Computational Thinking Into Middle School Curriculum* PI: D. Tatar. Co-PI: C. Corallo. Co-PI: D. Kafura Co-PI: M. Perez-Quinones Co-PI: S. Harrison $199,998 for 10/1/11 to 10/1/13. The planning grant developed a novel mechanisms for involving graduate students in the analysis, design, implementation, and testing of technology support for inserting computational thinking concepts into core middle school curriculum. A full proposal was developed. **Intellectual Merit** included the validation of the model for developing technology applicable to middle-school classrooms that had the potential to also convey computational thinking ideas. **Broader Impacts** include the prototyping of web tools related to learning fractions and analyzing texts that were subject ot initial testing in middle schools with high concentrations of students from under-served populations. Publications about the work are under review.

NSF DUE Project (DUE-1140318) *TUES-Type1:Transforming Introductory Computer Science Projects via Real-Time Web Data* PI: E. Tilevich. Co-PI: C. A. Shaffer. $200,000.00 for 07/2012 to 06/2015. This project creates an educational software infrastructure to support computer programming projects that use real-time web-based data to better engage and better train introductory computer science students. The project has led to research papers presented at SIGCSE [4] and SPLASH-E [3]. **Intellectual Merits** include validation of the theory that contextualization can provide more engaging introductory programming experiences that also improve student comprehension of real-time technology. **Broader Impacts** include a workshop offered at SIGCSE 2014 to introduce the developed technology to our peers in other institutions [30]. In addition, the curricula of CS1 and CS2 classes at Virginia Tech the University of Delaware were enhanced with the projects developed under the auspices of this project.

References

[1] Chris Anderson. The end of theory. *Wired magazine*, 16, 2008.

[2] Ruth E. Anderson, Michael D. Ernst, Robert Ordóñez, Paul Pham, and Steven A. Wolfman. Introductory programming meets the real world: Using real problems and data in cs1. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE '14, pages 465–466, New York, NY, USA, 2014. ACM.

[3] A. C. Bart, E. Tilevich, S. Hall, T. Allevato, and C. A. Shaffer. Using real-time web data to enrich introductory computer science projects. Oct 2013.

[4] A. C. Bart, E. Tilevich, S. Hall, T. Allevato, and C. A. Shaffer. Transforming introductory computer science projects via real-time web data. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE '14, pages 289–294, New York, NY, USA, 2014. ACM.

[5] A. Bundy. Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1(2):67–69, 2007.

[6] Lori Carter. Why students with an apparent aptitude for computer science don't choose to major in computer science. In *Proceedings of the 37th SIGCSE technical symposium on Computer science education*, SIGCSE '06, pages 27–31, New York, NY, USA, 2006. ACM.

[7] Jeong-Im Choi and Michael Hannafin. Situated cognition and learning environments: Roles, structures, and implications for design. *Educational Technology Research and Development*, 43(2):53–69, 1995.

[8] Diana I Cordova and Mark R Lepper. Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. *Journal of educational psychology*, 88:715–730, 1996.

[9] C. Day. Computational thinking is becoming one of the three R's. *Computing in Science and Engineering*, 13(1):88–88, 2011.

[10] Anne E. Egger. Engaging students in earthquakes via real-time data and decisions. *Science*, 336(6089):1654–1655, 2012.

[11] Allan Fisher, Jane Margolis, and Faye Miller. Undergraduate women in computer science: experience, motivation and culture. In *Proceedings of the 28the SIGCSE technical symposium on Computer science education*, SIGCSE '97, pages 106–110, New York, NY, USA, 1997. ACM.

[12] E. Fouh, M. Akbar, and C.A. Shaffer. The role of visualization in computer science education. *Computers in the Schools*, 29:95–117, 2012.

[13] E. Fouh, V. Karavirta, D.A. Breakiron, S. Hamouda, S. Hall, T.L. Naps, and C.A. Shaffer. Design and architecture of an interactive etextbook – the OpenDSA system. *Science of Computer Programming*, 88(1):22–40, August 2014.

[14] S. Hall, E. Fouh, D. Breakiron, M. Elshehaly, and C.A. Shaffer. Education innovation for data structures and algorithms courses. In *Proceedings of ASEE Annual Conference*, page Paper #5951, Atlanta GA, June 2013.

[15] C. Heeter. *Situated Learning for designers: Social, Cognitive and Situative Framework.* Michigan State University, 2005.

[16] Jan Herrington and Ron Oliver. An instructional design framework for authentic learning environments. *Educational technology research and development*, 48(3):23–48, 2000.

[17] B. D. Jones. Motivating students to engage in learning: The MUSIC model of academic motivation. *International Journal of Teaching and Learning in Higher Education*, 21(2):272–285, 2009.

[18] B. D. Jones and G. Skaggs. *Validation of the MUSIC Model of Academic Motivation Inventory: A measure of students' motivation in college courses.* Research presented at the International Conference on Motivation 2012, 2012.

[19] V. Karavirta and C.A. Shaffer. JSAV: The JavaScript Algorithm Visualization library. In *Proceedings of the 18th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2013)*, pages 159–164, Canterbury, UK, July 2013.

[20] Maria Knobelsdorf and Carsten Schulte. Computer science in context: pathways to computer science. In *Proceedings of the Seventh Baltic Sea Conference on Computing Education Research-Volume 88*, pages 65–76. Australian Computer Society, Inc., 2007.

[21] Ari Korhonen, Thomas Naps, Charles Boisvert, Pilu Crescenzi, Ville Karavirta, Linda Mannila, Bradley Miller, Briana Morrison, Susan H. Rodger, Rocky Ross, and Clifford A. Shaffer. Requirements and design strategies for open source interactive computer science ebooks. In *Proceedings of the ITiCSE Working Group Reports Conference on Innovation and Technology in Computer Science Education-working Group Reports*, ITiCSE -WGR '13, pages 53–72, New York, NY, USA, 2013. ACM.

[22] Ari Korhonen, Thomas Naps, Charles Boisvert, Pilu Crescenzi, Ville Karavirta, Linda Mannila, Bradley Miller, Briana Morrison, Susan H. Rodger, Rocky Ross, and Clifford A. Shaffer. Requirements and design strategies for open source interactive computer science ebooks. In *Proceedings of the ITiCSE Working Group Reports Conference on Innovation and Technology in Computer Science Education-working Group Reports*, ITiCSE -WGR '13, pages 53–72, New York, NY, USA, 2013. ACM.

[23] Jean Lave and Etienne Wenger. *Situated learning: Legitimate peripheral participation.* Cambridge university press, 1991.

[24] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela H Byers. Big data: The next frontier for innovation, competition, and productivity. 2011.

[25] Assaf Marron, 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 Abstractions*, AGERE! '12, pages 59–70, New York, NY, USA, 2012. ACM.

[26] Office of the Senior Vice President and Provost. Academic implementation strategy for a plan for a new horizon: Envisioning virginia tech 2013-2018, 2013.

[27] C.A. Shaffer, M. Akbar, A.J.D. Alon, M. Stewart, and S.H. Edwards. Getting algorithm visualizations into the classroom. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education (SIGCSE'11)*, pages 129–134, 2011.

[28] C.A. Shaffer, V. Karavirta, A. Korhonen, and T.L. Naps. OpenDSA: Beginning a community hypertextbook project. In *Proceedings of the Eleventh Koli Calling International Conference on Computing Education Research*, pages 112–117, Koli National Park, Finland, November 2011.

[29] C.A. Shaffer, T.L. Naps, and E. Fouh. Truly interactive textbooks for computer science education. In *Proceedings of the Sixth Program Visualization Workshop*, pages 97–103, Darmstadt, Germany, June 2011.

[30] Eli Tilevich, Clifford A. Shaffer, and Austin Cory Bart. Creating stimulating, relevant, and manageable introductory computer science projects that utilize real-time web-based data (abstract only). In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE '14, pages 743–743, New York, NY, USA, 2014. ACM.

[31] Marc Waldman. Keeping it real: utilizing NYC open data in an introduction to database systems course. *J. Comput. Sci. Coll.*, 28(6):156–161, June 2013.

[32] A. Weinberg. Computational thinking: An investigation of the existing scholarship and research. In *School of Education*. Fort Collins, Colorado State University, 2013.

[33] Jeannette M Wing. Computational thinking. *Communications of the ACM*, 49(3):33–35, 2006.