Keywords: Big Data, Education, programming environments

Introduction: Leveraging the promise of Big Data offers unprecedented opportunities for improved economic growth and enhanced innovation. Unfortunately, computer scientists in the workforce are woefully unequipped for this shifting paradigm. Indeed, a report by MGI and McKinsey's Business Technology Offices declares that "… by 2018, the United States alone could face a shortage of 140,000 to 190,000 people with deep analytical skills as well as 1.5 million managers and analysts with the know-how to use the analysis of Big Data to make effective decisions"[7]. In order to close this gap, we need to do more than just improve the education of existing students; we must draw from under-represented and non-traditional sectors, including K-12 students, non-CS majors, and continuing education.

There are two main obstacles on the way of effectively educating students on Big Data. First, its representation, manipulation, and expression is challenging, with modern curriculums and programming environments being inadequate. In fact, the definition of Big Data is quantities of information that cannot be handled with traditional methods[7]. Second, postponing these topics until late in the curriculum, as is commonly done[6], fails to prepare students to solve problems in this domain. Moreover, learning how to manage this complexity is central to interdisciplinary work in everything from agriculture to medicine to law, and everything in between[3]. We cannot expect collaborators to progress too far through our own subject, so therefore work must be done to cover this material in introductory courses, as early and often as possible. But how do we make this complicated and difficult topic accessible?

Hypothesis: Visual, beginner-friendly programming environments, such as Scratch, have proven to be successful in introducing novices to programming. Scratch has been used with great effect to teach under-represented computing groups everything from variable assignment to iteration. **I now propose to build on the success of Scratch to create a programming environment that will expose students to Big Data concepts, without the expected cognitive overhead and overload.** A promising approach is to leverage the facilities offered by popular modifications of the platform (e.g., Snap!) to add new programming blocks that enable users to manipulate Big Data. The key principle of this idea is to apportion complexity in a scaffolded manner: start small, guiding students through iteratively larger datasets until they finally reach Big Data scales.

Research Plan: This will be a design-based research project, emphasizing formative evaluations and iterative development while being guided by relevant educational theories. Specifically, by using research on how students learn to grapple with Big Data, we will drive the development of the programming abstractions[4][5]. Tackling the technological challenge of apportioning Big Data processing's complexity will require expertise in Software Engineering and High Performance Computing. As these components are developed, they will be tested on users in increasingly larger samples to determine their effectiveness. By gathering both quantitative and qualitative data, we can eventually create a product that can be subjected to summative analysis in authentic classroom, learning environments. Ultimately, we will measure both student performance and engagement in order to gauge the effectiveness of the new components and their effect on student motivation. The audiences used throughout this study will include K-12, Continuing Education, and non-major students interested in Computational Thinking.

New, exciting assignments will become available thanks to these new Big Data Processing components. For example, younger students could be tasked with analyzing social

networking data such as their Facebook news feed or Twitter channels. Most students are already used to this information, and Constructivist theories predict that the context will be a powerful pedagogical aid. Moreover, many kinds of cross-curricular questions can be posed around this data, including statistical queries ("How often do your friends post?") or sentiment analysis ("Do your friends tend to use positive or negative words?"). Finally, this Situated Learning style project raises important social questions on the nature of privacy in an organic setting: e.g. the potential for cyber-bullying requires that privacy be treated as a first-class concern.

Investigator and Location Strengths: I bring Software Engineering and Educational Theory expertise from a related NSF-funded project called RealTimeWeb, which introduces real-time data into introductory courses by apportioning complexity[1]. This system has already seen adoption at two universities and is being used by roughly 400 students[2]. By building off my unique experience in building software to help students manage real-time data streams, I feel I am uniquely positioned to develop this project. Furthermore, Virginia Tech is an ideal site for this project, with strong High Performance Computing and Data Analysis labs, a rich history of projects involving the Scratch platform, and impressive resources for collaborations with Learning Scientists and potential deployment sites.

Intellectual Merit: This work will require significant amounts of technical proficiency to address the problem of introducing Big Data into classrooms. Developing the proposed programming environments will discover principles that can enable multiple diverse sets of users to take advantage of Big Data. Additionally, this research will build off existing theory to yield new insights into how students solve problems involving Big Data; these insights will have implications in Data Processing, Visual Analytics, and Computer Science Education.

Broader Impacts: The classrooms using this technology should see measurable improvements in students' understanding of Big Data methods. These improvements will also broaden participation of under-represented and non-traditional groups, including women and non-majors. Finally, introductory courses will be contextualized for an increased pool of students, also flattening the learning curve for continuing-education professionals.

[1] **A. C. Bart**, E. Tilevich, C. A. Shaffer, T. Allevato, S. Hall, *Using Real-Time Web Data to Enrich Introductory Computer Science Projects*, SPLASH-E '13, Indianapolis, Indiana. October 26-31, 2013.

[2] **A. C. Bart**, E. Tilevich, C. A. Shaffer, T. Allevato, S. Hall, *Transforming Introductory Computer Science Projects via Real-Time Web Data*, SIGCSE '14, Atlanta, Georgia. March 5-9, 2014.

[3] Anderson, Chris. *"The end of theory."* Wired Magazine 16 (2008).

[4] Katy Börner. 2012. *Visual analytics in support of education.* In Proceedings of the 2nd International Conference on Learning Analytics and Knowledge (LAK '12), Simon Buckingham Shum, Dragan Gasevic, and Rebecca Ferguson (Eds.). ACM, New York, NY, USA, 2-3. DOI=10.1145/2330601.2330604

[5] Roger H. L. Chiang, Paulo Goes, and Edward A. Stohr. 2012. *Business Intelligence and Analytics Education, and Program Development: A Unique Opportunity for the Information Systems Discipline.* ACM Trans. Manage. Inf. Syst. 3, 3, Article 12 (October 2012), 13 pages. DOI=10.1145/2361256.2361257

[6] Sahami, Mehran and Roach, Steve and Cuadros-Vargas, Ernesto and Reed, David. *Computer science curriculum 2013: reviewing the strawman report from the ACM/IEEE-CS task force*. ACM. p. 3—4, 2012.

[7] The McKinsey Global Institute. 2011. *Big data: The next frontier for innovation, competition, and productivity*. McKinsey & Company.