TITLE: Enhancing Computer Science and Computational Thinking Pedagogy through a Scaffolded Data-Centric Approach

Project Summary

This proposal addresses important issues in Engaged Student Learning (Design and Development I) by: (1) stimulating student motivation through pedagogically rich data interrelated with social impacts, and (2) improving feedback to learners by increasing the immediacy of feedback on student developed algorithms through static code analysis and by interactive visualizations for big data, a principal form of data used in our work. A multi-methods assessment focuses on motivation, self-efficacy and cognitive gain made by students in diverse majors, both STEM and non-STEM in a Computational Thinking class and STEM majors in two CS classes.

Pedagogically rich big data raises the level of student motivation through authentic experiences using data of genuine scale and complexity about real phenomenon from authoritative sources. The data is characterized as “big data” or “real-time data”. The challenges in this work are to achieve a scaffolding of complexity, provision of study questions, and connections to social impacts. We will also develop a taxonomy to assist instructors in locating streams with target characteristics. Our previous work developed a big data framework. The challenges that we address now are: (1) how to access this framework from a block-based programming language and a visual programming environment, and (2) creating a scaffolded environment in a Computational Thinking course.

While the immediacy and interactivity of feedback is important to engaged learning, feedback on algorithms is usually done manually by instructors with the resulting loss of immediacy. We propose to develop static program analysis to provide immediate feedback for targeted programming exercises. An additional important challenge is providing an effective instructor authoring tool. We have developed a significant collection of visualizations for improved interactivity with traditional algorithms and data structures. A challenge is how to extend these visualizations for use with big data. The developed resources will be delivered by and seamlessly integrated with other course materials in the OpenEDX web-based framework.

Broader Impacts

The exposure of students to big data provides the "data literacy" and concern for social impacts called for in an NRC report. While advantageous for all students, these characteristics are especially engaging for students in under-represented populations. Creating access to big data streams from a block-based programming language and a visual programming environment adds value to all instructors using these approaches, allowing them to reach beyond "interesting" assignments to ones which are also "useful". Our work makes two broader contributions to instructors, especially the community using the OpenEDX framework: (1) the integration of the powerful algorithm and data visualization capabilities developed in our earlier work, and (2) the provision of static program analysis and a related authoring tool. The data streams and the visualization tools developed for programming big data are useful not only for those using big data in introductory courses but are also useful to instructors in emerging data science courses. Finally, through our dissemination through a High-School Teachers Workshop and a Girls in Computing Day we will impact pre-university learners.

PROJECT DESCRIPTION

1. Introduction (Dennis)

In response to the Engaged Student Learning (Design and Development I) solicitation we propose to improve the experience of learning about computing by diverse students in diverse majors in two ways. First we will develop and assess resources that increase student motivation through pedagogically enriched data that includes the exploration of social impacts based on a model of social impacts that we have prototyped and will extend. The pedagogically enriched data will be accessible through a carefully scaffolded framework, the lowest layer of which we have created in earlier work. Second, we will increase the immediacy of feedback through static code analysis of student exercises and increase the interactivity of feedback through manipulable visualizations of the pedagogically rich (big) data. This later work builds on our expertise in interactive visualizations for traditional data and algorithms. All of these resources will be delivered through and seamlessly integrated with other course materials in a single web-based framework.

The students whose learning experience we are seeking to improve are from three very distinct groups. The students in one group are in a newly created computational thinking class that is part of the general education curriculum at Virginia Tech. This class is open to all majors and we are actively seeking as broad a participation as possible from diverse fields of study. These students have little or no prior exposure to computer science and many are likely to have little confidence in their quantitative skills. Students in a second group are computer science majors in their first programming class. Some of these students have only a tentative commitment to the field of study, most have little awareness of the social and ethical concerns related to computing, and many have difficulty relating the technology of computing that they experience on a daily basis with the concepts of computing that they are confronting in their studies. The third group of students are more advanced computer science majors confronting the conceptual and practical intricacies of algorithms and data structures. These students are relatively committed to the field but need help in seeing the application of the techniques they are learning to real-world situations and need better help coping with the more challenging cognitive dimensions of the material they are learning.

In our work, pedagogically rich big data streams raise the level of student motivation by engaging students in authentic experiences. The experience is authentic because the data derives from real phenomenon (e.g., geophysical events or social media) [Eager-10, Eager-31], is from definitive sources (e.g., US Geological Survey or Reddit), and is of genuine scale and complexity (not a "toy" version) [Eager-2]. Further, big data is germane to a broad array of disciplines and interests, making it relevant to the learner's concerns. Students will generally perceive the authenticity of problems using big data because big data is frequently in the news and is increasingly crucial to science, business, and policy making at all levels of government [Eager-1]. Big data is used by others [refs], but the challenge we address is how to create "pedagogically rich" data stream. By pedagogically rich we mean that each stream will be accessible at different level of complexity, include connections between the data and its possible social impacts, provide questions that can be answered with the data, and be described in terms of a taxonomy of data characteristics that assist instructors in deciding whether the data stream is a candidate for an assignment they have in mind. Though we are using big data our goal is not to teach “data science” where big data is itself the object of study. Rather, our goal is to use big data to ground the instruction where computing itself is the object of study.

The big data streams and associated tools will be accessible by carefully scaffolded technology. This scaffolding builds on the success of the RealTimeWeb project, our framework for rapidly building real-time web-data centered assignments in introductory courses [Eager-3]. The RealTimeWeb tool chain allows students to work with challenging, but motivating, dynamic and/or large-scale data. Instructors can incorporate big data streams into new learning experiences. This framework has been deployed in multiple courses at Virginia Tech and the University of Delaware [Eager-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. The challenge that we address is how to incorporate the existing framework into a block-based programming language (Blockly) and the visual programming environment (Greenfoot) that are used in two of our courses. Scaffolding will also be used to address the challenge of providing an environment for working with big data streams in the computational thinking course. This scaffolded environment will enable students to follow the use-modify-create sequence with successively more technical and challenging aspects of computation being exposed at each step (i.e., moving from a point-and-click interface to block-based programming to textual programming).

Exploring the social impacts of computing also improves student motivation by exposing the value-oriented dimension of computer science. We have anecdotal evidence from our first offering of a Computational Thinking class that engagement with social impacts is motivating. Addressing the social dimension is particularly relevant to encouraging women and students from other under-represented populations to see computer science as an appealing discipline. to We are especially interested in providing this exposure to students in introductory courses when their sense of professional "identity" is in its early stages of development. Exploring social impacts also connects the learning to contemporary issues (e.g., electronic surveillance, net neutrality). Though social impacts is included in the CS Principles [ref] we address the challenge of providing a model of social impacts and ethical behavior that is accessible to beginning students. We develop such a model in this research.

The immediacy and interactivity of feedback is important to improved learning. Many eBook platforms, our own included, provide immediate feedback on questions with highly structured answers(true/false, multiple choice) or on program output (by comparison to hidden correct answers). However, feedback on algorithm design is usually done manually by instructors with the resulting loss of immediacy and degradation of the learning opportunity. We propose to incorporate static program analysis into an eBook platform so that some forms of immediate feedback can be provided. Our intention is to provide a mechanism for analyzing relatively small pieces of code that are developed by students as an answer to posed questions (e.g., "Write an algorithms that..."). An important challenge in this work is to develop an authoring o that allows salient features of the code to be described and related to varying forms of feedback. The provided feedback should be accessible both to the student and to instructors for additional comments. We will make this capability available in both the OpenEDX framework and the OpenDSA framework.

Interactive visualizations allow a student to gain insight into the dynamics of algorithms and the manipulations of data structures. In conjunction with our OpenDSA platform a significant collection of such visualizations have been developed. We propose to develop corresponding visualizations that are appropriate for big data. For example, a visualization that shows operations on a list of ten elements is not sufficient for a big data stream with thousands of elements. In addition to this challenge we also address the challenge of incorporating visualization capabilities into the block-based language (Blockly) and visual environment (Greenfoot) that we use in two of our classes. With the exception of the stand-alone visual environment these interactive visualizations will be implemented in both the OpenEDX and OpenDSA frameworks.

The assessment of our work will involve a variety of qualitative and quantitative methods. All of the researchers have experience with human subject research assessment and our team includes experts in educational assessment. Assessments will be conducted on motivation and self-efficacy as well as on course-specific cognitive gains. A particular assessment goal is to determine the relative importance of motivation and self-efficacy in introductory versus more advanced courses.

Broader Impacts

The exposure of students to big data provides the "data literacy" described in the National Research Council Workshop Report on "Training Student to Extract Value from Big Data". This form of literacy informs both future computer scientists and also future domain specialists. The early introduction of social impacts addresses the issue also identified in the NRC workshop that noted: "Students often do not recognize that big data techniques can be used to solve problems that address societal good, such as those in education, health, and public policy". We believe that the use of real world data and its related social impacts, while advantageous for all students, are especially engaging for students in populations currently under-represented in the computing community. Commenting on a number of studies [Barker, Carter , Disalvo&Buickmna, Fischer&Margolis], Goldweber et.al. write that “there is some evidence to suggest that success in broadening participation may be improved when computing is shown to connect with students’ values rather than their more superficial interests.” [Goldweber]. Through our dissemination efforts we will inform high school teachers though a High Schol Teachers Workshop sponsored annually by our department. We will also use tailored big-data activities in a Girls in Computing Day sponsored annually by our department to help inspire young girls toward computing and STEM study and career choices.

Another impact comes from the access to big data streams through a block-based programming language and a visual programming environment. This work adds value to all instructors using these approaches, allowing them to incorporate more realistic and motivating assignment and projects.

Our technology work also makes two broader contributions, especially to the community of authors in the OpenEDX community. First, the integration into OpenEDX of the powerful algorithm and data visualization capabilities developed in OpenDSA adds a new tool for constructing dynamic and engaging content. Second, the provision of static program analysis and a related authoring tool adds a powerful new tool for instructors to develop better instructional resources. Finally, the data streams and the visualization tools developed for programming big data are useful not only for those using big data in introductory courses but are also useful to instructors in data science courses.

2. Background – Related and Preliminary Work

a. Socio-Constructivism Theory (Cory)

In order to design effective pedagogy and technology, work must be grounded in well-researched educational theories. In this proposal, we deal with both cognitive and motivational concerns. We leverage popular Socio-Constructivism theories of knowledge for the former, but the latter is more complicated. We hypothesize that introductory students begin with holistic motivational problems and end with more specific self-regulation problems - as students progress through a discipline, they become more naturally engaged with the material, but still haven't fully developed the metacognitive tools needed to succeed with it. Therefore, we use two motivational theories: the MUSIC Model of Academic Motivation and Self-Regulation theory.

Socio-Constructivism is an evolution of the popular Constructivist learning theory that emphasizes the role of context in learning. Constructivism, which has already seen some popular application within Computer Science Education [Ben-Ari ref, Guzdial ref], posits that knowledge is actively and recursively constructed from prior knowledge, rather than being passively absorbed through direct instruction and textbook readings. Metaphorically, the PC is to the internet as Constructivism is to Socio-Constructivism, conveying the distribution of cognition under the enhanced framework. Although both theories suggest the use of Active Learning techniques with rapid feedback and enhanced agency of the student, Socio-Constructivism emphasizes the value of culture within the learning process. This culture can come from the instructor (as both a guiding presence and a source of direct instruction), the classmates (who share the learners inexperience but bring their own skills, history, and understanding to the table), and society at large (with its generations of resources, impetuses, and conventions). One of the ways that this culture is made concrete within the learning environment is Anchored Instruction, an approach where a problem is embedded within a frame that provides valuable details. Instead of decontextualized, abstract experiences, students must think critically within realistic scenarios that are easier to construct their knowledge upon. Socio-Constructivism is applied within this proposal to suggest the value of Social Impacts and strongly influences the technology to be developed.

Motivation is a major issue for early introductory students, but then Self-Regulation becomes a bigger problem later on. We use the MUSIC Model of Academic Motivation as a general lens to explore why people choose to participate and excel in Computer Science. The MUSIC Model is a tool specifically designed to explain engagement in education, setting it apart from more domain-unspecific motivational frameworks. Derived from a meta-analysis of other motivational theories, the model is a tool meant for both design and evaluation that has been extensively validated and utilized in other educational domains, making it a reliable device[TODO: Jones validity paper citation]. The MUSIC model identifies five key constructs in motivating students [TODO: Jones description paper citation]:

- Empowerment: The amount of control that a student feels that they have over their learning -- e.g., course assignments, lecture topics, etc..
- Usefulness: The expectation of the student that the material they are learning will be valuable to their short and long term goals. There is no clear delineation of the time-scale for these goals, but there is nonetheless a distinction between strategic skills that students need to be successful in careers and personal interests and the tactical skills they need to complete present-day tasks.

- Success: The student's belief in their own ability to complete assignments, projects, and other elements of a course with the investment of a reasonable, fulfilling amount of work.
- Interest: The student's perception of how the assignment appeals to situational or long-term interests. The former covers the aspects of a course related to attention, while the latter covers topics related to the fully-identified areas of focuses of the student.
- Caring: The students perception of other stakeholders' attitudes toward them. These stakeholders primarily include their instructor and classmates, but also can be extended to consider other members of their learning experience (e.g., administration, external experts, etc.).

Students are motivated when one or more of these constructs is sufficiently activated.
They are not all required to achieve maximal levels, and in fact that is not always desired -- it is possible, for instance, for a student to feel too empowered, and become overwhelmed by possibilities. Students' subjective perception of these constructs is a defining requirement and is more important than objective reality. The MUSIC Model of Academic Motivation Inventory (MMAMI), a well-validated instrument, is used to measure engagement through the five aspects.

Self-Regulated Learning was actually one of the sources of the MUSIC Model, expressing itself in the Success aspect. In the SRL model, learners succeed by actively practicing and developing their learning, willingly taking on challenging tasks, and reflecting on their strategies that lead to success.
In Computer Science, students must develop their understanding of how long programming tasks take, understand where their conceptual knowledge is weaker and believe that they can improve (as opposed to a fixed view), and a host of other metacognitive skills. Evidence collected by PI Shaffer suggests that ... [SHAFFER]. In order to assess students' self-regulation, we rely on the well-validated Motivated Strategies for Learning Questionairre (MSLQ) developed by Pintrich.

b. Issues in social impacts/ethics (Dennis)

Understanding the impacts of computing and information technology on society is widely recognized as important in the education of computer scientists and engineers. As noted in [Lambrinidou] the Accreditation Board for Engineering and Technology (ABET) criteria require students to have "an understanding of professional and ethical responsibility," and have acquired "the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context." [ABET] To educate computing professional the new CS Principles Course [CollegeBoard] includes a specific task and performance assessment described as follows:" Computing innovations have had considerable impact on the social, economic and cultural areas of our lives. To focus your work on this task, select a computing innovation that h as significant impact, or the potential for significant impact on our society, economy, or culture, and that possesses the potential for both beneficial and harmful effects." The ACM Code of Ethics [ACM] states that "An essential aim of computing professionals is to minimize negative consequences of computing systems, including threats to health and safety. When designing or implementing systems, computing professionals must attempt to ensure that the products of their efforts will be used in socially responsible ways, will meet social needs, and will avoid harmful effects to health and welfare." Similarly, the Software Engineering Code of Ethics and Professional Practice [ACM-SE] notes that software engineers: "Because of their roles in developing software systems, software engineers have significant opportunities to do good or cause harm, to enable others to do good or cause harm, or to influence others to do good or cause harm."

However, the exploration of social impacts and the acculturation of students to ethical professional behavior is often missing early in the curriculum and/or fosters a "culture of disengagement" [CECH] that distances the student, and later professional, from genuine engagement with the impact of their practice on society. In our own curriculum, for example, the topic of ethics and social impacts does occur until the student has reached the upper division (junior year) in the curriculum. The sense of "disengagement" is described in a critique of the ABET standards: "By not specifying whose definitions of “desired needs,” “realistic constraints,” “engineering problems,” or “contemporary issues” are to be used and how these definitions are to be identified, ABET seems to imply that somehow students on their own can know the technical and social complexities associated with their work simply by virtue of their training. At the same time, student ability to elicit and take into account non-dominant perspectives is absent from the list." [Lambrinidou] Similarly, in framing the NAEs Grand Challenges for Engineering, [Lambrinidou] observes that "Moreover, it does not seem to represent “people’s” own views on what engineering challenges compromise their ability to “thrive” and how engineers can help address these challenges." The challenge then is to provide an early exposure to issues of social impact and ethical professional behavior early in the education of future computing professionals and in a way that allows students to begin forming an awareness of non-technical perspectives on the impacts of their work.

With the use of big data in introductory classes as envisioned in this proposal it will be possible to engage students both early and in a meaningful way. The strands of proposed work with big data and with social impacts are, therefore, synergistically connected because the big data provides an important motivating context for the exploration of social impacts.

c. Background on automatic feedback and interactive algorithm visualization (Cliff)

Dynamic process, such has the behavior of an algorithm, is difficult to convey using static presentation media such as text and images in a textbook. During lecture, instructors typically draw on the board, trying to illustrate dynamic processes through words and constant changes to the diagrams. Many students have a hard time understanding these explanations at a detailed level or cannot reproduce the intermediate steps to get to the final result. Another difficulty is lack of practice with problems and exercises. Since the best types of problems for such courses are hard to grade by hand, students normally experience only a small number of homework and test problems, whose results come only long after the student gives an answer. The dearth of feedback to students regarding whether they understand the material compounds the difficulty of teaching and learning DSA.

For this project, we will build modules using OpenDSA technology, which addresses the issues listed above. OpenDSA modules combine content in the form of text, visualizations, and simulations with a rich variety of exercises and assessment questions. Since OpenDSA modules are complete units of instruction, they are easy for instructors to use as replacements for their existing coverage of topics (similar to adopting a new textbook) rather than including an AV on top of their existing presentation. Since OpenDSA's exercises are immediately assessed, with problem instances generated at random, students gain far more practice than is possible with normal paper textbooks. Since the content is highly visual and interactive, students not only get to see the dynamic aspects of the processes under study, they also get to manipulate these dynamic aspects themselves. Emphasizing student engagement with the material conforms to the best practices as developed through more than a decade of research by the AV research community [63, 56, 42].

Each module includes mechanisms for students to self-gauge how well they have understood the concepts presented. Self-assessment can increase learner's motivation, promote students' ability to guide their own learning and help them internalize factors used when judging performance [50, 3]. We do make use of simple multiple choice and give-a-number style questions, for which we use the Khan Academy Exercise Infrastructure [38] to generate individual problem instances. We also include many interactive exercises. We make extensive use of "algorithm simulation" or "proficiency" exercises, as pioneered by the TRAKLA2 project [47]. (Note that the TRAKLA2 developers from Aalto University in Helsinki are active participants in OpenDSA, having developed the JSAV graphics library [37, 35] and several OpenDSA exercises. One could view OpenDSA as "TRAKLA3".)

In algorithm proficiency exercises, students are shown a data structure in a graphical interface, and must manipulate it to demonstrate knowledge of an algorithmic process. For example, they might show the swap operations that a given sorting algorithm uses. Or they might show the changes that take place when a new element is inserted into a tree structure. Other OpenDSA exercises make use of small simulations for algorithms or mathematical equations to let students see the effects that result from changing the input parameters. Small-scale programming exercises are automatically assessed for correctness. These problems are similar to small homework problems traditionally given in such a course, but which have been hard to grade.

d. Preliminary Results from CT class (Dennis and Cory)

Social Impacts.

In consultation with Dr. Yanna Lambrinidou [Lambrinidou] we have developed a three-part prototype model for engaging students with the topic of social impacts and responsible professional behavior. This model builds on work in ethics education [Lambrinidou] and also in value-sensitive design [VALUE] from the human-computer interaction community. The first part of the model connects computational knowledge and skills to the power to affect society. Computation enables the creation of new capabilities (social media, dating sites, computational science, etc.) and creates models or analysis of the real world. The new capabilities can change the way society is structured or operates in both positive (social activists' use of social media to organize in the "Arab spring") and negative (use of social media for cyberbullying) ways. Model or analysis based on computation allow citizens, scientists, and polity makers to understand the world in new ways. Concretely, it has an impact on products, services, and government policies. The role of analysis and models based on big data can be clearly illustrated in this model (as can other forms of computational modelling). The second part of the model asks students consider for a given system or model who are stakeholders, their interests, and the way sin which the stakeholders are affected. This exploration seeks to identify the different perspectives of the stakeholders as well as divergences or conflicts between them. For each stakeholder the student seeks to understand how the stakeholders' values and well-being are impacted. The third part of the model focuses the attention on what pressures might be affecting the behavior of each stakeholder.

The prototype model was used in the Computational Thinking class designed by PI Kafura and offered for the first time in Fall, 2014. In one assignment students were asked to apply the model in two different cases: (1) a case where injury and death was caused by a medical device (an X-ray machine) that was badly programmed and (2) a case where the system performed exactly as intended but raised the potential for harm (a story in the L.A. Times of a mother unable to get her daughter to an emergency room because the mother's car was disabled remotely by the company holding the lien on the car). From students' personal interaction with PI Kafura it was clear that students generally found the social impacts material and assignments interesting and engaging. Further study and more formal assessment of the motivational aspects of the social impacts material is included as part of the proposed work.

Big Data and Tools

e. Other courses using big data (Dennis and Cory)

We share with the media computation approach [Guzdial] the idea of providing a unifying, open-ended resource (images and sound in media computation vs. big data data in our course). However, we believe that big data is seen by students as “useful” which is more engaging than media computation which is seen as “interesting” [Guzdial&Tew].

We share a common goal with courses that use real-world data for motivational purposes. Examples include using on-line data [DePasquale], and assignment that produce useful tools [Stevenson]. While sharing a common goal our approach uses big data.

We also share a concern for using resources that relate social impacts. Among this work is [Goldweber] that uses a values-oriented approach to exploring the social implications in various computational modelling assignment and [Erkan] that uses sustainability issues to frame problems used in a data structures and algorithms class. We differ from these approach in the resource (big data in our case) to which the social concern is connected.

We share the most with the work of [Andersen] that also uses big data in such areas as life sciences, political science, social media, and text sources. Like us, they have also explored allowing students to choose the data for a major project. We propose to extend this work in a number of ways

Our proposed work integrates and extends these concerns for engagement, realism, social good, and big data. The integration is achieved by extending the "raw" bid data streams with elements connected to a model of social impacts. The extension involves the development of interactive visualizations, static analysis for immediacy of feedback, the access to big data through two different block-based programming languages, and other supporting technology. We also extend the application of this approach outside of mainstream computer science education to the general university student population via a computational thinking course. Finally, we add additional assessment data to add to the body of knowledge on the impacts and limits of the big data approach.

In two of our courses we also have a shared view with courses that used block-based programming (Snap!, Scratch, or App Inventor). What we add to a block-based programming approach is the connection to realistic big data sources and the ability to embed the programming in a “book” form to better integrate learning materials (see Section 4).

3. Proposed Work

a. Pedagogically-rich data streams (Dennis and Cory)

We propose to build on our previous work of creating and using big data streams. To support the newly developed Computational Thinking class a gallery of approximately 25 big data streams were developed. The data represented a wide range of subject areas including cancer, drug usage, education, automobile fuel statistics, world economic data, and exoplanet data. These data streams are represented in JSON and accessible through a simple Python interface. Many of the big data streams have a multi-layer structure of Python lists and dictionaries.

We propose to enrich these "raw" big data streams to create resources with increased pedagogical potential. Each "pedagogically rich" big data stream will include:

* a structured set streams of varying complexity. In using the data streams in the Computational Thinking class we saw that it would be useful to have simpler versions of the data stream available for use by students earlier in the class. This allows the student to develop a familiarity with the data farther in advance of its use in the project and better support their learning of the data structure concepts. For

example, a multi-layer data stream of crime reports across the United States has various categories and sub-categories of crimes organized by state and by year. This data is complex enough that students must have a good understanding of lists and dictionaries used together in nested forms. Thus, the student cannot begin to work with the data until those skills have been acquired using other data that is potentially less interesting to the particular student. However, a simplified version of the crime data might be a single list of the total number of crimes in a given state across the years. The student could work with this data while learning about lists. A simple dictionary form would organize data for a single year as a key-value pair for each state with the state name as the key and the total crimes as the value. Successively more complex intermediate forms could also be provided. This structuring allows student to more directly connect their learning with the big data stream of their choice.

* material related to social impacts. The material included here is related to the model of social impacts that we will also developed as described in the next section (see XXX). This model will be applied to each of the data streams. For example, the crime data stream described above would identify stakeholders (e.g., law enforcement, legislators, homeowners, insurance companies), their interests (crime prevention, resource allocation, relocation decisions, policy rates), and the pressures which might affect their behavior (demonstrate success, re-election arguments, family safety, profitability). The inclusion of this material enhances the motivational power of the big data approach. As noted earlier, there is evidence to believe that social factors are effective for retaining under-represented groups.

* possible research questions. For each data stream there would be a list of questions that the student should be able to answer using the data. These questions contribute to the student's understanding of data-driven research and evidence-based decision making. The questions also support the real-world character and social impacts of the work by posing issues (in the form of a question) that is of evident interest to one or more of the stakeholders. For example, questions for the crime data stream could include:

What is the overall crime rate in a given year?
What is the trend in property crimes over time?
How does the rate of violent crimes in state X compare to that in state Y?
Which state has the lowest overall crime rate in a given period?

To improve the accessibility by an instructor of the collection of data streams a taxonomy for classifying the data streams will be developed. This taxonomy addresses the same issues as the CSG-Ed Rubric developed by the ACM ITiCSE Working Groups [Goldweber]. Of particular concern to us is the programming and data concepts that are a prerequisite for using the data stream and what additional programming and data concepts the data stream can support. This taxonomy allows instructors to focus on particular data streams. The taxonmy also provides us a well to help design the structured set of streams of varying complexity.

b. Social Impacts. (Dennis)

We will elaborate the prototype model described above (see section X.X) and integrate this model with the collection of big data streams also being developed in this work. The elaboration defines a check-list of characteristics that can be used in exploring the impact on a stakeholder. A tentative list of characteristics is:

- *privacy*: especially for big data what is the impact on the stakeholder's sense of self and ability to control disclosure of personal information (for example, consider social media and cyberbullying).
- *pervasiveness*: especially in the Internet of things what is the impact on the stakeholder of the pervasiveness and invisibility of computing (for example, the ability to invisibly and remotely disable a vehicle does not announce the vehicle's availability for an emergency situation as would having the vehicle towed away).
- *power*: especially with availability of big data what is the impact on the stakeholder's ability to advance their interests or mission (for example, how does the availability of highway traffic data enable a Department of Transportation manage and plan for roadways versus enabling stalking of an individual's movements).
- *privilege*: especially with the differences between groups in society and between societies what is the impact on a stakeholder due to the differential access to data or computing (for example, a stakeholder without access to the same data as other stakeholders is likely to be disadvantaged in policy debates where that data is used by one side).

For the elaborated model we will develop resources that can be used by ourselves and others to present the model in traditional or flipped classroom settings, materials for exercises, and rubrics for evaluation. These "stand-alone" resources can be adopted by others without also using any other part of our work. We will also, of course, integrate similar resources into the big data streams we are developing. We will draw on numerous bodies of work including popular readings ([Abelson], Rushkoff]), reports and writings related to big data ([WhiteHouse], [Kord]), and on-line resources for ethics ([Risks], [NSFW] ,[OEC]).

The assessment of these materials will be done as part of the assessment of motivation in the Computational Thinking class (PI Kafura) and the Introduction to Computer Science class (co-PI Tilevich).

c. automatic feedback via static analysis of algorithms in (Cliff and Cory)

We discussed above the value of automatically assessed practice exercises. A specific type of exercise that we focus on will be small programming exercises. There are a number of existing systems that support this [REFS]. Most take the student's solution to a "sandbox" where the solution is compiled and executed. A standard approach is to use something equivalent to unit tests to make sure that the student's solution has the correct behavior. We have found that it is also necessary to make sure that students develop a solution that is "done the right way" as well as generates the correct output. For example, recursive tree functions should limit their concerns to the current node as much as possible, while we find that students often want to (unnecessarily) check the values of child nodes as well. Thus, we build in static analysis heuristics to better constrain and assess the solutions that the student provides.

- **block-based/Python environment (Blockly)**
- **Java environment/Greenfoot**

d. algorithm and data visualizations for big data (Cliff)

The OpenDSA project has developed a sophisticated support system for developing rich interactive visualizations in HTML5. We have already developed algorithm visualizations and exercises for much of undergraduate DSA content, and will be expanding this content to more advanced topics in programming languages, finite automata, and complexity theory in the coming year under other NSF supported projects. This rich body of materials and expertise in their creation will allow us to create a collection of materials appropriate to the non-major students in the CT course. Many of the fundamental data structures and themes relevant to big data are already covered within the existing collection of materials. We will be able to tailor those materials and create new materials more appropriate for this constituency. The OpenDSA infrastructure allows us to create small-scale simulation environments and interactive exercises that will let students "play with" big data streams and principles.

e. Scaffolded environment for CT class (Dennis)

A heavily scaffolded environment for use in the Computational Thinking class will be developed and assessed. This environment will replace the current use of NetLogo. In the current design of the class NetLogo is used for three reasons:

- it allows students to perform significant computational activities by manipulating user interface controls. Thus, the students can have a computational experience from the very beginning of the course.
- it has a rich library whose component models appeal to a wide variety of disciplines. Thus, students can see relevant computational applications in areas of interest to them as well as interact with students in students in other majors using other models.
- it is scaffolded so that the underlying code can be exposed, but only as needed to deepen the exploration of computing. Thus, students can gradually expand their understanding of how the model they have selected is constructed.

These properties of NetLogo stimulate student's engagement and concretize an interdisciplinary perspective of computational principles.

However, NetLogo has a number of significant drawbacks with respect to the computational thinking course. These are:

- the computational models of NetLogo are not related to the use of big data in the remainder of the course. Thus, there is a degree of discontinuity between major parts of the course
- the programming language of NetLogo is idiosyncratic and textual. For example, the basic agent is a "turtle", iteration is concealed as "ask"ing an implicit group of agents, and basic properties agents (i.e., location, color) are implicitly defined making it harder for students to see how the model works. The textual nature of the language means that students have to confront issues of syntax that would be better to defer.
- there are no social aspects defined for the NetLogo models. Thus, instructors must separately devise materials that are relevant to each of the NetLogo models in use.

On balance, these properties of NetLogo motivate the design, development, and assessment of an environment that:

- offers an early-to-use interface for manipulating big data streams
- has a library of big-data streams relevant to multiple disciplines
- is scaffolded for incremental and progressive exploration of the underlying computation that is expressed in a non-textual language
- includes material on social impacts.

We propose to build such an environment.

In constructing this environment we will:

- use standard user interface Python libraries to create an environment with simple UI controls and rich visualizations,
- use of Blockly as the first implementation language that is presented when students "look under the hood". Using Blockly creates a seamless transition to the next part of the course which focuses on algorithms in Blockly.
- use a method to import a data stream that is compatible with the pedagogically-rich data streams developed in other parts of the research.

Using Blockly creates a seamless transition to the next part the course which focuses on algorithms in Blockly. Providing compatibility with the pedagogically-rich data streams insures that a variety of data streams will be available and that they will include material on social impacts.

- **technology support**
    - **integrating into OpenEDX (Cliff)**

[DENNIS -- I DON'T KNOW YET IF THIS MAKES SENSE. BUT WE SHOULD KNOW MORE IN A FEW DAYS.]

- **algorithm and data visualization capabilities from OpenDSA**
- **automated feedback**
- **block-based programming access to big data**

e. Use

- **application/use**

Course	**Big Data**	**Social Impacts**	**OpenEDX (feedback, viz)**	**Scaffolded Env.**
CT	**x**	**x**	**x**	**x**
CS1	**x**	**x**		
CS3	**x**		**x**	

- **use cases**
    - **CT (Dennis)**
    - **CS1 (Eli)**

At Virginia Tech, CS 1114 is an introductory programming class for majors, with a small of percentage of non-majors exploring their interest in computing. To emphasize the object-oriented domination of the current programming landscape, the course follows the objects-first teaching methodology using the Greenfoot integrated development environment (IDE). Greenfoot is a domain-specific visual language for creating 2D games and simulations. Greenfoot enables the user to create a hierarchy of game Actors with a click of a mouse, while only the program logic is implemented by writing Java code. Greenfoot instills introductory learners with the importance of design-first software development principle, while a Virginia Tech extension to the IDE also conveys the primacy of test-driven development practices. Unfortunately, the programming assignments currently given with this IDE focus exclusively on entertainment. That is, from an educational perspective, one could characterize these assignments as only arousing situational interest. Hence, these assignments fail to engage and motivate those students for whom a sense of authenticity and usefulness becomes a primary means of engaging with a discipline.

The disconcerting reputation of CS-1114 is a "weed-out course" because a large percentage of students registered for this course decide not just to withdraw, but to not pursue computer science as their major. Our hypothesis is that *students taking this course fall into two major categories: the ones who find computer science naturally appealing and engaging, and those who could be convinced to pursue computing as their major, if shown that the discipline provides a powerful tool for solving real-world problems*. The first category of students is served well by the current project offerings of the class, whereas the second category requires novel educational interventions. It is the latter category of students which will primarily benefit from carrying out the proposed project.

We posit that big data enhancements have great potential benefit not only for students in this course, but for all students who use similar visual environments to smooth their acculturation into their professional sphere. Specifically, we propose to create and evaluate Greenfoot-specific bindings that would make it possible to enhance the existing project offerings of this introductory IDE with current and future CORGIS big data libraries. There is also a unique opportunity to leverage Greenfoot as a powerful data visualization tool.

Example projects that can be created using the proposed technology include:

- "Tweet Analysis" -- given a bunch of hashtags, retrieve the top tweets for it, and make a word cloud to graphically visualize frequency of words and their synonyms.
- “Errands Planner”--given a campus map, a list of class meetings/locations, and a list of day errands, produce an animated itinerary for doing the errands around a student’s classes for a given day.
- “Weather Reporting Lab”-- for a map of the US, show the current weather as it updates

To concretely demonstrate the proposed deliverable of this part of the project, consider Figure X. This figure shows how a typical Greenfoot 2D World can be enhanced with a CORGIS big data library.

FIGURE goes here

As this figure clearly demonstrates, big data can effectively increase the real-world relevance of programming assignments in an introductory course. The associated social impacts can motivate a range of problems and meaningfully contextualize them. The proposed evaluation of this deliverable will assess how strongly our hypothesis is supported by experiences in the classroom.

Evaluation notes:

Motiations: MMAMI for quantative, MUSIC for qualitative. Possibly some self-efficacy data too.

Cognitive: final exam, final project

- **CS3 (Cliff)**

At Virginia Tech, CS3114 Data Structures and Algorithm Analysis is a key course for all CS majors and minors. Students often have difficulty with the programming projects in this course, since they tend to require complex concepts such as dynamic memory allocation, recursion, file processing, differing interpretations of the bytes that represent data, and a collection if interacting classes. The designs are more complex than they are used to, as they stress interacting classes requiring appropriate separation of concerns. Too large a fraction of these students fail the course due to inability to properly manage their time. Research on procrastination [REF] indicates that motivation and self-efficacy for time management are key determinants for avoiding procrastination. In this project, we will be providing big data streams as an available source of "interesting" projects. We will be able to study the extent to which such projects become intrinsically motivating. We can also study issues of self-efficacy related to project management.

Another important issue for CS3 is practice with programming exercises. For example, the types of recursion (typically on trees and other recursive structures) is more sophisticated than these students have encountered in the past. And often, their prior experience with recursion does not include enough practice. So we will take advantage of the OpenDSA system and the improved support for programming exercises that we are developing to provide more practice on a number of topics. These include recursion on trees, basic pointer manipulation, and hands-on use of various data structures in small practice functions.

f. Assessment (from Jeremy Ernst)

Progressions of learner motivation and engagement, outcome proficiency, and self-regulated learning behaviors, in the context of data rich course experiences, will be assessed in Virginia Tech's CT, CS1, and CS3 courses. Within these course offerings, student sections will be categorized as control and experimental groups where control groups receive lecture-based instruction while the experimental groups experience a data rich scaffolded environment. The robust CT and CS enrollment and course offering frequency provides an opportunity to simultaneously compare learning approaches while minimizing confounding factors. Specific research questions and metrics for the implementation and impact study are discussed in Table XX.

Table XX: Research questions and data sources

Data Source	Pre/Post MUSIC	Pre/Post CT/CS	Pre/Post MSLQ	Semester Interviews
RQ1 Do big data applications support increased classroom motivation and engagement?	X			
RQ2 Do data rich course experiences, supported with scaffolded technologies, promote student outcome proficiencies?		X		

RQ3 Do data rich student group outcome proficiencies exceed that of student groups not exposed to data rich experiences?		X		
RQ4 Do big data applications impact student self-regulated learning behaviors?			X	
RQ5 Does course success vary across student populations (CS major and non CS major), student demographical categories and social dimensions?		X		X
RQ6 Does classroom motivation and engagement vary across student populations (CS major and non CS major), student demographical categories and social dimensions?	X			X
RQ7 Do self-regulated learning behaviors vary across student populations (CS major and non CS major), student demographical categories and social dimensions?			X	X

RQ1: The MUSIC Inventory (Jones, 2014) will be administered in a pre-measure/post-measure format to gauge study group student increases in motivation and engagement. The specific constructs measured through the instrument are: 1) Empowerment, 2) Usefulness, 3) Success, 4) Interest (situational), and 5) Caring. The Inventory consists of 26 prompts where students select from response options on a 6-point Likert Scale. The response options range from 1 – Strongly Disagree to 6 – Strongly Agree. The Virginia Tech-based MUSIC Inventory researchers have conducted studies to provide rigorous validation evidence for the metric (Jones & Skaggs; 2014; Jones and Wilkins; 2013, Jones, 2010; Jones, Epler, Mokri, Bryant, and Paretti, 2013). Study group students will be administered the MUSIC Inventory within the first full week of class and then again immediately after the intervention has been completed. Analyses of the repeated MUSIC Inventory measures will be conducted to determine significant changes within-subject comparison.

RQ2 and **RQ3**: A parallel item cognitive pre-assessment and post-assessment will be administered to the student study group and to a control group to measure degrees of student cognitive content achievement concerning block-based programming, Python, and Java. Study team content representatives will devise these aligned cognitive content measures where a subject matter evaluator panels will then be formed. There will be a CT team, a CS1 team, and a CS3 team comprised of instructors of these designated courses for the purposes of reviewing their specified content assessments. After content revisions are made, the evaluator panel will then divide into an instructor pre-assessment groups and instructor post-assessment groups. The pre-assessment group will administer the pre-assessment to their students and the post-assessment group will administer the post-assessment to their students. This process is in efforts to establish content and concurrent validity as well as test-retest reliability and internal consistency reliability. Once refined, the pretests will be administered to the study group prior to the onset of the intervention-based instruction and activities and the posttest administered at the completion of the planned intervention-based instruction and activities for the CT group, the CS1 group, and the CS3 group. Analyses of the repeated measures will be conducted to determine significant changes within-subject comparison and between group comparison.

RQ4: The Motivated Strategies for Learning Questionnaire (MSLQ) will be administered in a pre-measure/post-measure format to gauge changes in student self-regulated learning behaviors. The MSLQ was developed based on a social-cognitive view of motivation and self-regulated learning (SRL) in the college classroom (Pintrich, 2004). The conceptual framework for SRL in the college classroom contains two dimensions, composed of four areas for regulation. Phase 1 involves goal setting, planning and activation of perceptions, and knowledge. Phase 2 concerns monitoring processes. Phase 3 refers to control or regulation. Phase 4 represents reaction and reflections on the self and the task or context. The reliability and validity of the MSLQ were tested through statistical methods over several waves of data collection. Two confirmatory factor analyses were used to determine the utility of the theoretical model and the operationalization for the scale (Pintrich et al., 1993; Artino, 2005). Student participants will also be administered the MSLQ within the first full week of class and then again immediately after the intervention has been completed. Analyses of the repeated measures will be conducted to determine changes within-subject comparison.

RQ5: CT, CS1, and CS3 data sources from RQ2, paired with student demographics will be used to determine if course experiences have differential impacts on student academic outcomes among CS non-majors, CS majors at the introductory program level, and CS majors entering the secondary level of the program. In addition to major and program level, gender, ethnicity, and first generation college status will be factored in analyzing success across student demographic and social dimensions.

RQ6: Data from RQ1, paired with student major, level, demographics, and social dimensions will be used to determine if course experiences have variability in impact on student motivation and engagement.

RQ7: Data from RQ4, paired with student major, level, demographics, and social dimensions will be used to determine if course experiences have variability in self-regulated learning behaviors. Stratified end of semester follow-up interview protocols will be employed in efforts to explore specific elements of course experiences, instructional materials, organizational structure, classroom environment, and their influences on course successes, motivation/engagement, and self-regulated learning behaviors.

g. Dissemination (Dennis)

Our dissemination plan is organized around community-building activities supported by technology practices that facilitate adoption.

Community-building Activities

We will create a wiki-based web site as a central point for the distribution of all resources developed in the project. The wiki-based nature will also encourage contributions from the community of adopters. Natural ways to contribute are through the addition of new data streams, new projects, and new social impacts activities.

We will promote awareness of the developed resources through BOF sessions and workshops at primary computer science education conferences. Publication of the results demonstrating the benefits of the resources will be done in venues with wide visibility in the computer science education community.

The use of OpenEDX promotes adoption by creating visibility in the growing community of educators using OpenEDX for course delivery.

We will bring awareness of our work to high school teachers through an annual high school teachers workshop sponsored by our department. PI Kafura has twice participated in this workshop. We will also participate in an annual Girls in Computing Day sponsored by our department for junior-high girls. By engaging young girls with realistic and socially-meaningful computing experiences we hope to positively affect their subsequent study and career choices toward computing and STEM fields.

Technology Practices
A key element of our dissemination activity is the use of a "componentized" approach to the design and development of the resources created during the project. This means that adopters have wide latitude in choosing which and how much of the developed resources will be adopted. For example, a adopter could choose any of the following levels of use:
1. use only one (or a small number) of the raw big data streams for use in an existing assignment to improve the student engagement with that assignment
2. use the entire library of raw big data streams to give students enhanced self-direction
3. use the entire library of pedagogically rich data streams so that the social impacts dimension could be included in the course experience

4. use the entire library of pedagogically rich data streams and the scaffolded execution environment so that a progressive exploration of algorithmic manipulation of the data streams can be done in a supportive environment
5. use all of the artifacts in item 4 together with the learning resources in the EDX framework. Alternatively, an adopter may use only the social impacts module in an existing course to enrich the discussion of professional ethics, or use only the scaffolded execution environment with their own data streams. Other combinations are also possible.

Adoption is also facilitated by the use of open-source standards and widely used tools (e.g., Sculpt, OpenEDX).

5. Timeline

6. Results from Prior NSF Support

Award (TUES-1444094): PI Dennis Kafura, 2014-15; Co-PIs Cliff Shaffer and Eli Tilevich. *Scaffolding Big Data for Authentic Learning of Computing*; $97,658. **Intellectural Merit:** This award supported the design, construction of technology support for, and assessment of a course in Computational Thinking. The course was offered for the first time in Fall, 2014 using project-based exploration of complex phenomena by algorithmically manipulating large-scale and/or real-time data streams from real-world sources. A complete electronic book was created (see think.cs.vt.edu/book) that included immediate feedback questions, features for group collaboration, and interactive use of the block-based programming language (Blockly). The initial results show that the students were highly motivated and engaged by the course design and the use of real-world data. For example, students reported high average scores in all five areas of the MUSIC model, with no strong standard deviation. The results indicate that students “Agreed” in the belief that they were empowered, able to succeed, cared for, and that the course was interesting and useful. **Broader Impact:** The project provides a model for how university-level courses can be designed to engage students from a wide variety of disciplines in a computational thinking experience. It also provides additional evidence and available resources for using a "big data" approach to instruction in computer science courses. This form of heightened engagement and real-world connections help to attract and retain students from populations under-represented in the computing field.

Award (CNS-1132227): D. Tatar, PI; C. Corallo, co-PI, D. Kafura, co-PI; M. Perez-Quinones, co-PI; S. Harrison, co-PI), *Planning Grant: Integrating Computational Thinking Into Middle School Curriculum***.** 10/1/11-1/31/15. $199,998 + REU Supplement $24,000. **Intellectual Merit:** This planning grant took an integrated approach to developing computational thinking in middle school curriculum. Core curricular teachers identified key instructional problems for which computational thinking or proto-computational thinking solution were devised by students trained in in iterative, participatory ideation, design and development. Applicable findings from the CE21 praoject include heuristics about scope of complexity and depth of subject matter that can be developed in single semester projects and the limits of concept transfer. **Broader Impact:** The work in the planning grant involved 6 Henrico County K-12 teachers and about 270 students. Two of the schools involved were failing schools, and our work included a 6th grade class in a failing school that had not passed their 5th grade high stakes mathematics tests. We continue to work on ideas that were developed in this context.

Awards (CNS-0540509 and CNS-0810850): PI Dennis Kafura, 2006-2008; *Collaborative Research: Alliance between Historically Black Universities and Research Universities for Collaborative Education and Research in Computing Disciplines***;** $139,295; **Intellectural Merit:** This grant facilitated the participation of Virginia Tech in an alliance of three R-1 universities and five HBCU institutions to strengthen undergraduate computing programs, create and maintain research experiences for undergraduates, and support ongoing research and teaching partnerships among faculty members. The alliance also developed a "pod" model for research collaboration involving faculty and students from an R-1 partner and an HBCU partner with the goal of increasing the level of graduate degrees pursued by HBCU students. **Broader Impacts:** During the period of the award a research-oriented course was offered in a short format for undergrads--with weekly meetings over a 4-6 week period--and in a graduate course format, for grad students planning on pursuing Ph.D. degrees. Students were encouraged to generate research products, culminating in a paper worthy of publication in a peer-reviewed conference. Many students who took part in this seminar course participated in summer research experiences--and ultimately toward graduate study. These activities sought to increase African-Americans' entry into computing research careers, support new faculty in maximizing their career potential, and produce a steady progression of role models for undergraduate students, indirectly increasing the participation of African-Americans in computing professions more generally.

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. **NSF SAVI/EAGER Award (IIS-1258571)** *Dynamic Digital Text: An Innovation in STEM Education*, PIs: S. Puntambekar (UW-Madison), N. Narayanan (Auburn), and C.A. Shaffer (2013). $247,933, 01/2013 -- 12/2014. **NSF CCLI Phase 1 Award (DUE-0836940)** *Building a Community and Establishing Best Practices in Algorithm Visualization through the AlgoViz Wiki.* PIs: C.A. Shaffer, S.H. Edwards. $149,206, 01/2009-12/2010. **NSF NSDL Small Projetc (DUE-0937863)** *The AlgoViz Portal: Lowering Barriers for Entry into an Online Educational Community.* PIs: C.A. Shaffer, S.H. Edwards, $149,999, 01/2010-12/2011. **Intellectual Merit** The first two projects provided online infrastructure (the AlgoViz Portal (http://algoviz.org) and related community development efforts to promote use of AV in computer science courses. This work was an important precursor to OpenDSA, as it allowed us to interact with many CS instructors and AV developers, leading us to an understanding of the fundamental missing parts in existing DSA instruction. They also initiated many of the international collaborations that lead to OpenDSA. Three journal papers [Shaffer10, Fouh:AV11, Cooper14] and five conference papers [ShafferSIGCSE07, ShafferSIGCSE10, ShafferSIGCSE11, ShafferPVW11, ShafferKoli11] have been produced relating to this work.The second pair of (ongoing) awards support the initial phases of OpenDSA, and an active collaboration involving Virginia Tech and Aalto University (Helsinki), among others. Publications related to this work so far include [KorhonenWG13, Karavirta:ITiCSE13, Hall13, ITiCSEWG13, Fouh14CHP, Fouh14SCP]. **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 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 2014 [BartSIGCSE14] and SPLASH-E 2013 & 2014 [BartSPLASHE13,BartSPLASHE14]. 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 workshops offered at SIGCSE 2014 and SIGCSE 2015 to introduce the developed technology to our peers in other institutions [WorkshopSIGCSE14,WorkshopSIGCSE15]. 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.

NSF EHR Project (DRL-1156629) *Transforming Teaching through Implementing Inquiry (T2I2)* project. PI: J. Ernst, Co-PIs: L. Bottomley, A. Clark, V.W. DeLuca, S. Ferguson. $1,997,532 for 09/2011-8/2015. Intellectual Merit: This full research and development project explores the use of cyber-infrastructure tools to significantly enhance the delivery and quality of professional development for grades 8-12 engineering, technology, and design educators. The goal is to study whether the use of highly interactive cyber-infrastructure tools increases the educators': 1) understanding of how to address student learning needs 2) ability to manage, monitor, and adjust the learning environment 3) use of self assessment to enhance teaching ability and 4) engagement in a community of practice. Results to date have shown that sixteen teachers from five states (teaching grades 6-12) have attained satisfactory competency on the learning objects [Ernst13; Ernst12; Segedin13] Broader impact: The focus on using an object-oriented system design enables the cyber-infrastructure to be reusable, adaptable, and scalable.

======= Parts of the original outline that have been filled in but put here for reference===

a. Goal

- **improve the learning experience for diverse students in diverse majors through motivating contexts supported by scaffolded technology**

b. Approach

- **enhance level of student engagement**
  - **use pedagogically rich big data streams**
  - **incorporate awareness of social impacts and ethical behavior through a model accessible for students**
- **increase immediacy and interactivity and integration**
  - **techniques for automated feedback on algorithm exercises/problems**
  - **interactive algorithm and data visualizations for big data scenarios**
- **technology support**
  - **scaffolded environment for access to big data streams**
  - **integration of automated feedback and interactive elements with instructional materials forming a seamless environment that avoids the distraction of switching between tools/contexts.**

c. Assessment in contexts that differ in

- **student populations (major vs. non-majors)**
- **level (CS0 vs. CS1 vs CS3)**
- **environment and language (block-based programming and Python, visual environment and Java, Java)**

======= Material below is from Cory earlier that he will incorporate into above outline ======

Motivation

Engaging CT, CS students

Leveraging more than Interest

-> Usefulness: Social Impacts, the Tools that they're learning

-> Empowerment through personalization

-> Caring through Cohorts, Pairs, Groups

Skills ~ Knowledge

Existing data from this semester goes here?

Background

Educational Theories

Situated Learning Theory

Situated Learning Theory, originally proposed by Lave and Wenger, argues that learning normally occurs as a function of the activity, context, and culture in which it is situated\cite{lave-situated}. Therefore, tasks in the learning environment should parallel real-world tasks, in order to maximize the \textit{authenticity}.
Contextualization is key in these settings, as opposed to decontextualized (or ``inert'') settings.
The key difference is that learning is driven by the problem being solved, rather than the tools available – therefore, the problem being solved should lead directly to the tool being taught.

A critical element of these situated environments is the need for social interaction and collaboration, as learners become involved in and acculturated by ``Communities of Practice'' \cite{brown1989situated}. Members of a CoP share purposes, tools, processes, and a general direction -- they should have a commonly recognized domain.
This interaction occurs not only between individuals and the experts of the community (commonly represented by teachers) in an apprenticeship model, but also occurs between different learners as they adapt at uneven paces.
This communication between peers leads to growth among both individuals, especially when access to authentic experts is limited.
As the learner develops into an expert, they shift from the outside of the community's circle to the center, becoming more and more engaged -- this is the process of ``Legitimate Peripheral Participation''.

Authenticity is another crucial, recurring theme within Situated Learning Theory.
All instruction and assessment must be aligned with reality such that success in the former leads to success in the latter.
However, there is a subtle nuance here -- authenticity is a perceived trait, not an objective one. Students derive value from their learning only if they \textit{perceive} authenticity, regardless of whether the instructor has successfully authenticated the experience.

The original work in Situated Learning Theory was categorically not about pedagogy or instructional design- it described how people learn and the importance of context and collaboration, but it did not recommend a particular style of classroom.
However, subsequent research by Brown \cite{brown1989situated} and others expanded the theory so that it could be applied to the design of learning experiences. These expansions often naturally dictate the use of active learning techniques, demphasizing the role of lecture in favor of collaborative, problem-based learning activities.

Choi & Hannafin \cite{situated-cognition} describe a particularly useful, concrete framework for designing situated learning environments and experiences. This framework has four key principles:

Context: ``*... The problem's physical and conceptual structure as well as the purpose of activity and the social milieu in which it is embedded*"\cite{rogoff1984everyday}, context is driven not just by the atmosphere of the problem at hand, but also by the background and culture surrounding the problem.

A good context enables a student to find recognizable elements and build on prior understanding, eventually being able to freely transfer their learning to new contexts.

Content: The information intending to be conveyed to the students.

If context is the backdrop to the learning, then content might be seen as the plot.

Naturally, context and content are deeply intertwined with each other, and its difficult to talk about one without referencing the other; in fact, content is an abstract entity that needs to be made concrete through contextualization when it is delivered to the learner.

If the information is too abstract, than it will never connect with the learner and will not be transferable to new domain.

However, if it is too grounded in a domain, then it will not be clear how it can be re-applied elsewhere.

Ultimately, content must be given in a variety of forms to maximize transfer.

Two useful methods for building content are anchored instruction (exploring scenarios, or anchors, in the context based on the content) and cognitive apprenticeship (mediating knowledge from an expert to the novice learner in a mentoring relationship).

Facilitations: The modifications to the learning experience that support and accelerate learning.

Facilitations provide opportunities for students to internalize what they are learning by lowering the barriers that can surround situated experiences, possibly at the cost of some amount of authenticity.

These modifications might be technological in nature, but they can also be pedagogical.

Although there are many different forms that Facilitations can take, Scaffolding is one of the most common.

Scaffolding is a form of support that is intended to extend what a learner can accomplish on their own.

This support is required at the onset of the learning process, but is unnecessary once a sufficient threshold has been passed; during this transition, the amount of scaffolding can be tuned to the learners understanding.

In Computer Science, for instance, students often take advantage of software libraries and frameworks to create sophisticated graphical programs that would be beyond daunting if implemented from scratch.

Assessment: The methods used to assess the learning experience and the progress of the student.

Choi & Hannafin gives special attention to the “teach to the test” problem, and how assessment needs to change to measure students ability to solve authentic problem (as opposed to their ability to solve the test’s specific problem), and to be able to transfer their understanding when solving different but related problems.

It is important that assessment is measured against the individualized goals and progress of a learner, requiring that any standards used be fluid and adaptable to different learners personal situations.

Of course, assessment should be an on-going part of the learning process, providing feedback and diagnostics.

Ultimately, the learner should join in the process of assessment as they transition to an expert – being able to meta-cognitively self-evaluate the effectiveness of ones methods and communicate results to others are key abilities of experts.

Situated Learning Theory has seen limited application in Computer Science Education Research. For instance, Guzdial and Tew \cite{guzdial2006imagineering} used the theory to innovatively explore and deal with the problem of inauthenticity within their Media Computation project. An earlier paper by Ben-Ari \cite{ben2004situated} explores its application and limitations. This paper is somewhat hasty in its application of SL Theory by taking a macro-level view -- they narrowly look to Open-Source and Industry Software Development communities as the only potential CoPs and interpret SL Theory as strictly requiring constant legitimacy, largely ignoring the possibility for gradual development of authenticity within individual courses and modules throughout a curriculum.

The MUSIC Model of Academic Motivation

Situated Learning is a theory of learning, but is not a comprehensive motivational framework -- it describes how people learn, but it is limited in explaining why people commit to learning.

Instead, we use the MUSIC Model of Academic Motivation as a lens to explore why people choose to participate and excel in Computer Science.

The MUSIC Model is a holistic model specifically designed to explain engagement in education, setting it apart from more domain-unspecific motivational frameworks.

Derived from a meta-analysis of these other theories, the model is a tool meant for both design and evaluation that has been extensively validated and utilized in other educational domains, making it a reliable device [TODO: Jones validity paper citation]

The MUSIC model identifies five key constructs in motivating students [TODO: Jones description paper citation]:

Empowerment: The amount of control that a student feels that they have over their learning -- e.g., course assignments, lecture topics, etc..

Usefulness: The expectation of the student that the material they are learning will be valuable to their short and long term goals. There is no clear delineation of the time-scale for these goals, but there is nonetheless a distinction between strategic skills that students need to be successful in careers and personal interests and the tactical skills they need to complete present-day tasks.

Success: The student's belief in their own ability to complete assignments, projects, and other elements of a course with the investment of a reasonable, fulfilling amount of work.

Interest: The student's perception of how the assignment appeals to situational or long-term interests. The former covers the aspects of a course related to attention, while the latter covers topics related to the fully-identified areas of focuses of the student.

Caring: The students perception of other stakeholders' attitudes toward them. These stakeholders primarily include their instructor and classmates, but also can be extended to consider other members of their learning experience (e.g., administration, external experts, etc.).

Students are motivated when one or more of these constructs is sufficiently activated.
They are not all required to achieve maximal levels, and in fact that is not always desired -- it is possible, for instance, for a student to feel too empowered, and become overwhelmed by possibilities.
Much like in Situated Learning Theory, students' subjective perception of these constructs is a defining requirement and is more important than objective reality.

The MUSIC model is often used as an organizational framework and an evaluative tool.
As the former, it is a list of factors to consider when building modules, assignments, and content of a course.
At all times, instructors can consider whether they are leveraging at least one construct to motivate their students.
As the latter, it offers both a quantified instrument (MMAMI) and a structure to anchor a qualitative investigation on.

Big Data

Big data has been loosely described as quantities of information that cannot be handled with traditional methods \cite{manyika2011big}.
But "traditional methods" is a vague phrase that has different meanings to different learners. To a Humanities major in their first CS-0 course, the traditional method to sum a list is to use Excel. In this scenario, "big data" means anything that won't comfortably fit into Excel's working memory.
However, to a third-year Computer Science major, the traditional method would be to write an iterative or recursive sequential loop; being given big data forces them to explore parallel models of execution.
Clearly, "bigness" is a function of the learner's experience, but that is still not a solid definition.

A more precise definition is the "3V Model" \cite{douglas2012importance}, which posits that there are three dimensions that distinguish big data from ordinary, run-of-the-mill data:

Volume: The total quantity of the information, usually measured in bytes or number of records. However, this also extends laterally: the number of fields in the structure of the data also impacts the complexity and size. The threshold at which data becomes big is a function of the hardware and software being used -- for instance, embedded systems may consider gigabyte-sized files to be big, while modern servers might not struggle until the petabyte level.

Velocity: The rate at which new information is added to the system. High velocity big data implies a distributed architecture, since new data must be arriving from somewhere. The dynamicity of data can vary widely across these architectures, with data updating every year, every day, or even multiple times a second.

Variety: The format or formats of the data. Ideally, data are always distributed in a way that is readily accessible -- for instance, simple text-based formats such as CSV and JSON are widely supported, relatively lightweight, and human-readable. More sophisticated data formats for image and audio are also typically well-supported, although still more complicated. However, projects using specialized, compressed binary formats or, more dangerously, multiple formats (e.g., image archives organized with XML files), are more complex.

There are many challenges inherent to using Big, real-world data.

Intentionally secured data,
Unintentionally obfuscated data,
Non-uniform topologies
Distribution of Data
Storage
Stability of Access
Efficiency

Computational Thinking (CT = Abstraction + Algorithms)
Prior work in Computational thinking - how are we different?
*CS Principles
*Media Computation

Digital Textbook

Ethics
Collaboration
Block-based programming
Introductory Programming

Proposal
Ethical Framework
CORGIS - Social Impacts, Questions, Data Structures, Pedoagogical Power
Block-based programming
Data Analysis Environment
Auto/Semi-auto assessment
Cohorts?
Interactive Book content
Community/extension workshops

Evaluation
MMAMI

Qualitative MUSIC: Surveys, Focus groups
Assessment of knowledge

References

J. Ernst, A.C. Clark, V.W. DeLuca, and L. Bottomley. Professional development system design for grades 6-12 technology, engineering, and design educators. In Proceedings of the American Society for Engineering Education Annual Conference and Exposition, June 2013.

J.V. Ernst. Authentic assessment in performance-based stem education activities. In Proceedings of the Scaling STEM: Transforming Education Matters Annual Conference, April 2012.

L. Segedin, J.V Ernst, and A.C. Clark. Transforming teaching through implementing inquiry: A national board-aligned professional development system. In Proceedings of the Association for Career and Technical Education Research, 2013.