**Engaged Student Learning (Design and Development I)**

1. Introduction

We propose to improve the experience of learning about computing by diverse students in diverse STEM majors through data-centric resources (Figure 1) that stimulate and sustain engagement and are delivered through a robust web-based learning platform. First, *pedagogically enriched data* is created that adds scaffolding and context to realistic data. This work builds on our instructional framework for big data and real-time data. Second, a *model of social impacts* is developed that structures one way by which the realistic data is enriched. We have had positive experience with a prototype of this model. Third, the *immediacy and interactivity of feedback* will be improved: static code analysis of targeted student code improves immediacy and interactive visualizations specifically designed for pedagogically enriched data improves interactivity. This later work builds on our expertise in interactive visualizations for traditional data structures and algorithms [1]. Fourth, two *scaffolded environments* using the pedagogically enhanced data will be built, an extended visual programming environment and a newly created block-based programming environment. This work builds on our experience to extend a block-based environment. Extensive *multi-methods assessment* will be conducted to determine the effects of these resources on motivation and self-regulation in three different computing courses. We will deliver our new resources integrated with other course materials using the *Open edX* online learning platform [2].

|  |
| --- |
| **Figure 1: Data-Centric Resources** |
|  |

By pedagogically enriched data we mean data that is intrinsically realistic and is equipped with the scaffolding to make it highly usable by learners and instructors. Data is intrinsically realistic if it concerns real-world events (e.g., scientific, economic or social), is from authoritative sources (e.g., government agencies, social media), and is of genuine scale and complexity (not a “toy” version). We use data that is big, real time, and/or geo-located. Intrinsically realistic data raises the level of student motivation by engaging students in authentic experiences [3-5]. We are, of course, among several who use realistic data to increase student motivation [3, 6]. However, our vision goes far beyond the "raw" data itself to include four aspects of scaffolding. First, the data should be related to important social impacts and issues of ethical professional behavior. This aspect of the scaffolding is important because it grounds the data in concerns that both engage students’ interest and help form their sense of professional identity. Second, the data should be layered. Our own experience has shown that "big data", while highly motivational, is too complex for early assignment where the student’s skill set is still limited. Layering allows the student to work with data at varying levels of complexity as their knowledge and skill increases. Third, questions should be provided that can be answered by using the data, thus engendering a sense of data-driven inquiry illustrating the utility of programming for solving real world problems. Fourth, the data should be classified in a taxonomy of data characteristics that assist instructors in deciding whether the data is a candidate for an assignment that they have in mind. We propose to create this taxonomy.

We have anecdotal evidence from our first offering of a Computational Thinking class that inclusion of 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. 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 associated with big data (e.g., privacy). Social impacts is included in the CS Principles [7]. We address the challenge of providing a model of social impacts and ethical behavior that is accessible to beginning students.

Immediate 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 immediate feedback to small programing exercises can be provided. We will develop a mechanism for analyzing targeted code that is developed by students as an answer to posed questions (e.g., "Write an algorithm that..."). An important challenge in this work is to develop an authoring tool 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 within the Open edX framework.

Interactive feedback through visualizations allows a student to gain insight into the dynamics of algorithms and the manipulations of data structures. We have developed a significant collection of such visualizations in our OpenDSA framework [61]. We propose to develop corresponding visualizations appropriate for big data (our primary form of realistic 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. We also address the challenge of incorporating visualization capabilities into the block-based language (Blockly) used in one of our classes. With the exception of the stand-alone visual environment these interactive visualizations will be integrated into the Open edX framework.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 1: High Level Plan** | | | | |
| **Course** | **Pedagogically Enriched**  **Data** | **Social Impacts** | **Automatic**  **Feedback, Visualization** | **Environments** |
| **CT** | **x** | **x** | **x** | **block-based** |
| **CS1** | **x** | **x** |  | **visual** |
| **CS3** | **x** |  | **x** |  |

The pedagogically enriched data 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 [8]. 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 [9]. 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 our Computational Thinking course. This scaffolded environment allows 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).

The assessment of the impact of these resources will involve three distinct groups as summarized in Table 1. First are students in a newly created Introduction to Computational Thinking (CT) class that is open to all majors, including majors in all STEM fields. Second are computer science and other STEM majors in their first programming class (CS1). 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. Third are more advanced computer science majors confronting the conceptual and practical intricacies of algorithms and data structures (CS3). 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. Each course is offered each semester with enrollments ranging from 30 to 80 students. PI Kafura will teach the CT course each semester during the project. Co-PIs Shaffer and Tilevich will teach CS3 and CS1, respectively, at least one per year during the project.

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-regulation as well as on course-specific cognitive gains. A particular assessment goal is to determine the relative importance of motivation and self-regulation in introductory versus more advanced courses.

We make our resources available in the Open edX learning platform for several reasons. It is a stable open platform with a large user community and significant developer support. It has a more (in comparison with Runestone that we have been using) friendly tool for content developers (Studio). It has a number of features (e.g., roles and problem directives) that are critical to our use. It also has built-in analytics that can help with fine-grain assessment.

**Broader Impacts**

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 future domain specialists. The early introduction of social impacts addresses another issue 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 [10-13], 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.” [14]. Our dissemination will include training of high school teachers though a High School 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 Open edX community. First, the integration into Open edX 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.

1. Background – Related and Preliminary Work

2.1 Socio-Constructivism and Motivation Theories

Our work is grounded in well-researched educational theories of cognitive and motivational concerns. We leverage 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 needed metacognitive tools. Therefore, we use the MUSIC Model of Academic Motivation [15] and Self-Regulation theory [16] as the underpinning for our work.

Socio-Constructivism is an evolution of Constructivist learning theory that emphasizes the role of context in learning. Constructivism, which has already seen some application within Computer Science Education [17], posits that knowledge is actively and recursively constructed from prior knowledge rather than being passively absorbed through direct instruction and textbook readings. 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 way that this culture is made concrete within the learning environment is Anchored Instruction, an approach where a problem is embedded within a frame story (the anchor). 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.

The MUSIC Model of Academic Motivation is 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 meant for both design and evaluation and has been extensively validated, making it a reliable device [18]. The MUSIC model identifies five key constructs [15]:

* Empowerment: The amount of control that a student feels that they have over their learning.
* Usefulness: The expectation of the student that the material they are learning will be valuable to their short (tactical) and long term (strategic) goals.
* Success: The student's belief in their own ability to complete 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.
* Caring: The student’s perception of other stakeholders' (e.g. instructor, classmates) attitudes toward them.

Students are motivated when one or more of these constructs is sufficiently activated. Students' subjective perception of these constructs is more important than objective reality. The MUSIC Model of Academic Motivation Inventory (MMAMI), a well-validated instrument, is used to measure engagement through these five aspects.

In the Self-Regulated Learning (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. In an algorithms and data structures (CS3) course at Virginia Tech we have observed that students often have difficulty with the course programming projects. These projects involve concepts such as dynamic memory allocation, recursion, file processing, differing interpretations of the bytes that represent data, and a collection of interacting classes. The designs are more complex than students are used to, as the projects stress interacting classes requiring appropriate separation of concerns. Too large a fraction of these students are unable to properly manage their time. Research on procrastination [19, 20] indicates that motivation and self-efficacy for time management are key determinants for avoiding procrastination. In this project, we will provide big data as an available source of "interesting" projects. We will study the extent to which such projects become intrinsically motivating. We will focus on self-efficacy related to project management. To assess students' self-regulation, we rely on the well-validated Motivated Strategies for Learning Questionnaire (MSLQ) [21].

2.2 Issues in social impacts/ethics

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 [22] 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” [23]. To educate computing professionals, the new CS Principles Course [7] 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 has 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 [24] 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 [25] notes: "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 are often missing early in the curriculum and/or foster a "culture of disengagement" [26] 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 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." [22] Similarly, in 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." [22] 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 big data provides an important motivating context for the exploration of social impacts.

2.3. Automatic Feedback and Interactive Visualization

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 computer science.

For this project, we will build several content modules related to ethics and big data using OpenDSA technology [61]. 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). 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 algorithm visualization research community [27-29].

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 [30, 31]. We do make use of simple multiple choice and give-a-number style questions. But we also include many interactive exercises. We make extensive use of “algorithm simulation” or “proficiency” exercises, as pioneered by the TRAKLA2 project [32]. (The TRAKLA2 developers from Aalto University in Helsinki are active participants in OpenDSA, having developed the JSAV graphics library [33, 34] and several OpenDSA exercises.) 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. We have experience with small-scale programming exercises that 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.

2.4. Preliminary Results from CT class

In consultation with Dr. Yanna 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 [22] and also in value-sensitive design [35] from the human-computer interaction community. The model (1) connects computational knowledge and skills with the power to affect society by creating new capabilities in and models of the world; (2) identifies stakeholders, the different perspectives of the stakeholders as well as divergences or conflicts between them, and how the stakeholders' values and well-being are impacted; and (3) posits pressures affecting each stakeholder that shapes the stakeholder’s behavior in socially desirable or undesirable ways.

|  |
| --- |
| **Figure 2: Preliminary Assessment Data** |
|  |

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. From students' interactions with PI Kafura it was clear that they 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.

Preliminary data from the first offering of the Computational Thinking class provides some positive indication of the benefits of using real-world big data as well as some negative indication regarding the use of NetLogo [36] as a tool of initial exploration of computation (see Figure 2)

In Fall 2014 NetLogo was used because: (1) it allows students to perform significant computational activities by manipulating user interface controls; (2) it has a rich library whose component models appeal to a wide variety of disciplines, appealing to each student’s interest and allowing a student to interact with students in other majors using other models, and (3) it is scaffolded so that the underlying code can be exposed, but only as needed to deepen the exploration of computing.

These properties of NetLogo stimulate student's engagement and concretize an interdisciplinary perspective of computational principles. However, preliminary data from the CT class shows that students found NetLogo to be less interesting, less helpful, and less useful in comparison to the real-world data (see Figure 1).

We believe that the somewhat negative reaction to NetLogo is due to 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 are implicitly defined making it harder to see how the model works. The textual nature of the language forces students to confront issues of syntax that would be better to defer.
* No social aspects are defined for the NetLogo models. Thus, instructors must separately devise materials that are relevant to each of the NetLogo models in use.

2.5 Related work

We share with the Media Computation approach [37] the idea of providing a unifying, open-ended resource (images and sound are used in Media Computation vs. big data in our course). However, we believe that big data is seen by students as “useful”---which is more engaging---while media computation is seen as only “interesting” [38]. We share a common goal with courses that use real-world data for motivational purposes. Examples include using on-line data [39] and assignments that produce useful tools [40, 41]. We also share a concern for using resources that relate social impacts. Among this work is [14] that uses a values-oriented approach to exploring the social implications in various computational modeling assignment and [42] 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 [3] that also uses big data in such areas as life sciences, political science, and social media. Like us, they have also explored allowing students to choose the data for a major project. In two of our courses we also have a shared view with courses that used block-based programming (Snap!, Scratch, or App Inventor).

We propose to extend prior work in a number of ways. Our proposed work integrates and extends these concerns for engagement, realism, social impact, and big data. The integration is achieved by extending the "raw" big 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, our assessment data will add to the body of knowledge on impacts and limits of the big data approach. We add to a block-based programming approach 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).

1. Proposed Work

3.1. Pedagogically-enhanced data

We propose to build on our previous work of creating and using big data. To support the newly developed Computational Thinking class a gallery of approximately 25 big data streams were developed in the CORGIS framework [43] . We propose to enrich these "raw" big data streams to create resources with increased pedagogical potential. Each "pedagogically rich" big data stream will include the following elements.

*Layered complexity*. In the Computational Thinking class we saw that having “slices” of the data available in simple to more complete forms would allow the students to engage with the data early (using simple forms) and progress to more complete and complicated forms of the data as their knowledge and skills developed. This structuring allows student to more directly connect their learning at each step with the data of their choice.

*Social impacts*. The material included here is related to the model of social impacts that we will also develop (see Section 3.2). 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.

*Research questions*. This set of questions, answerable using the data, contributes 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.

*Classification*. To improve the accessibility by an instructor a taxonomy for classifying the data will be developed. This taxonomy addresses similar issues as the CSG-Ed Rubric developed by the ACM ITiCSE Working Groups [14]. Of particular concern is the programming and data concepts that are a prerequisite for using the data and what additional programming and data concepts the data can support.

3.2 Model of Social Impacts

We will elaborate the prototype social impacts model described in Section 2.4 and integrate this with the collection of big data streams. 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 other parts of our work. We will also integrate similar resources into the big data streams we are developing. We will draw on numerous bodies of work including popular readings ( [44], [45] ), reports and writings related to big data ([46], [47]), and on-line resources for ethics ( [48], [49], [50]).

3.3 Immediate and Interactive Feedback

We discussed above the value of automatically assessed practice exercises. A specific type of exercise that we focus on will be small programming exercises. CS3 students in particular need more practice with programming exercises than is possible with manual grading. For example, the types of recursion (typically on trees and other recursive structures) is more sophisticated than these students have encountered in the past

There is considerable work on automated assessment of programming exercises via testing [51-53]. 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 generating 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. Our OpenDSA system already has limited support for static analysis heuristics of this nature.

In the lower-level Computational Thinking course, similar feedback acts as a guide for students to learn the basics of programming. When tasked with writing a function to average a list of numbers through iteration, students should know to use certain constructs – an external variable, a looping statement, and mathematical division. For such a problem, static analysis heuristics could be given that specify those constructs must be present – if the student tries to compile their program without it, they are directed to “just-in-time” instructional materials to help guide them. The helpfulness of that assistance can be calibrated to the student’s individual success level and sense of frustration within the course (measured through automatic metrics such as number of attempts and supplemented with data from the instructor).

In our piloted prototype systems, the static analysis is limited to regular expressions that are checked against the student’s code, e.g., requiring iteration by checking that the word “for” is present in the text of the program, at the start of a line, and with whitespace immediately afterwards (signaling its use as a programming identifier). This kind of analysis is limited – even complicated regular expressions are unable to account for constraints such as “initialize a variable outside of a ‘for’ loop, and then modify that variable inside the loop’s body”. Worse, it is delicate and unsustainable, since every exercise requires unique analysis that is difficult to modify.

We propose a flexible system of reusable heuristics that can help instructors define and then enforce constraints on students’ code by indicating specific errors and giving targeted feedback individualized for the student. These heuristics will operate over parsed abstract syntax trees, so that the program can be intelligently analyzed. Prioritization of the suggested heuristics, matched with targeted support, can provide a guiding experience for students. For example, the instructor might specify the following constraints and priorities in a problem requiring the student to write an average function:

1. High priority: Use a ‘for’ loop in their code. Link to section in textbook on iteration – unless they’ve already read that section recently, in which case directly state that they haven’t used a for loop.
2. Medium priority: Define a variable X outside a ‘for’ loop, and then modify X inside the ‘for’ loop’s body. Link to section in textbook on using variables for state – unless they have been struggling a lot with this problem, in which case suggest they ask an instructor.
3. Low priority: Ensure that the variable X is used after the ‘for’ loop in a mathematical division operation. Link to Wikipedia article on the formula for finding an average – unless they have already been given that hint, in which case explicitly suggest that they need to divide still.

In general, static code analysis struggles with the open-ended nature of programming. However, in an educational setting, the students’ code should always resemble some specific form, which is critically useful information to guide the analyzer. Our collection of reusable, flexible heuristics will empower instructors to rapidly define the form of the function, guide students to success, and ensure that they have used the proper coding techniques.

To improve the interactivity of feedback, the OpenDSA project has developed a sophisticated support system for developing rich interactive visualizations in HTML5 [33]. We have already developed algorithm visualizations and exercises for much of undergraduate data structures and algorithms 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.

3.5 Scaffolded environments

**3.5.1 Block-based environment**

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. 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 both a block-based langue and, when selected by the student, a text-based language.
* includes presentation of the research questions and social impacts information associated with the pedagogically enriched data.

Figure 3 shows a partial design of the interface of the envisioned environment. The left part of the figure shows the “Interface” tab containing controls that set parameters and display a visualization of crime data from the FBI (one of our current data sources). The “Algorithms” tab shown on the right displays the block-based computation that produces the visualization. The other tabs would display other parts of the pedagogically enhanced data – the social impacts, research questions, and the abstraction of the data. Not shown is a tab on the Algorithms page that would show the Python for the block-based code.

|  |
| --- |
| **Figure 3: Mockup of the Block-based Environment** |
|  |

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.

**3.5.2 Visual environment**

At Virginia Tech, the introductory programming class 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.

A disconcerting outcome is that 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.

|  |
| --- |
| **Figure 4: Example of Greenfoot Errands Planner Exercise** |
|  |

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. (See Figure 4.)
* “Weather Reporting Lab”-- for a map of the US, show the current weather as it updates

As these examples demonstrate, 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.

1. Assessment

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. Specific research questions and metrics for the implementation and impact study are discussed in Table 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 2: 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 [54] 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 [55-58]. 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**: Student achievement item-level outcomes from the 2014-2015 academic year will be examined in the CT, CS 1, and CS 3 course offerings. This measure is to determine intact degree of validity concerning measurement and reliability of result consistency for currently deployed student content evaluation. Also, this measure will assist in the formation of a non-treatment achievement baseline enabling further characterization of study group achievement progressions. Once item-level analyses have been conducted, a CT, CS1, and CS3 panel team (comprised of instructors of these designated courses) will be convened for the purposes of reviewing their specified content assessments.  After panel team content examination, each team will then build parallel items that will serve as project pre-assessments. The cognitive content achievement pre-measures will be administered to the study group prior to the onset of the intervention-based instruction and activities and then the post-measure 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 measures will be conducted to determine significant changes within-subject comparison and non-treatment baseline 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 [59]. 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 [21, 60]. 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.

1. Project Organization

The proposed work is organized as shown in the following table that identifies the major goals to be accomplished in each year of the project.

|  |  |
| --- | --- |
| **Year** | **Goals** |
| 1 | Develop additional big data resources  Develop (Fall) and trial run (Spring) of scaffolded environment for CT class  Refine Social Impacts model (Spring)  Develop automated feedback support and authoring environment  Port existing components to Open edX (Fall) and integrate (Spring) automated feedback support  Develop (Fall) and trial run (Spring) of visual environment (Greenfoot) integration  Dissemination activities |
| 2 | Assessment of motivation and self-regulation in CT class  Assessment of motivation and self-regulation in CS3 class  Develop enriched data streams by adding social impacts  Develop taxonomy of data streams  Develop interactive visualizations  Improvement of visual environment; assessment of motivation and self-regulation in CS1 class  Dissemination activities |
| 3 | Assessment in CT, CS2 and CS3  Analysis of data from multi-methods studies  Dissemination activities |

The project team is well qualified to carry out the proposed work. The team has a track record of collaborative work. Team members Shaffer and Ernst have collaborated on a previous educational project. Team members Kafura, Shaffer and Tilevich are currently collaborating on the use of big data in various courses. PI Kafura and Dr. Yanna Lambinidou, a consultant on this project (see Budget and Budget Justification) have collaborated on the definition of the preliminary social impacts model and its use in the Computational Thinking course. The team has extensive expertise with algorithm and data visualization (Shaffer), frameworks for big data and real time data access (Tilevich), computational thinking (Kafura), and web-based frameworks for course materials (Shaffer, Kafura).

1. Dissemination

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 Open edX promotes adoption by creating visibility in the growing community of educators using Open edX 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, allowing adopters wide latitude in choosing which and how much of the developed resources to adopt. For example, an adopter may:

* use only one (or a small number) of the raw big data streams for use in an existing assignment
* use the entire library of raw big data streams to give students enhanced self-direction
* use the entire library of pedagogically rich data streams so that the social impacts could be included
* 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
* use all of the artifacts in together with the learning resources in the Open 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, Open edX).

1. 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. **Intellectual Merit:** This award supported the design, construction of technology support for, and assessment of a course in Computational Thinking. The course was offered in Fall, 2014. An 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. **Broader Impact:** The project is a model of a university-level course that engages students from a wide variety of disciplines in computational thinking. It also provides additional evidence and available resources for using a "big data" approach to instruction in computer science courses. This heightened engagement and real-world connections help to attract and retain students from under-represented populations.

**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 [61-63] and five conference papers [63-67] 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 [1, 27, 33, 68, 69]. **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 [9] and SPLASH-E 2013 and 2014 [8, 43]. **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 [70, 71]. 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 [72-74]. **Broader impact**: The focus on using an object-oriented system design enables the cyber-infrastructure to be reusable, adaptable, and scalable.

**REFERENCES**

[1] E. Fouh*, et al.*, "Design and architecture of an interactive eTextbook – The OpenDSA system," *Science of Computer Programming,* vol. 88, pp. 22-40, 2014.

[2] Open edX. (2015. Available: <http://code.edx.org/>

[3] R. E. Anderson*, et al.*, "Introductory programming meets the real world: using real problems and data in CS1," presented at the Proceedings of the 45th ACM technical symposium on Computer science education, Atlanta, Georgia, USA, 2014.

[4] A. E. Egger, "Engaging student in earthquakes via real-time data and decisions.," *Science,* vol. 336(6089), pp. 1654-1655, 2012.

[5] M. Waldman, "Keeping it real: utilizing NYC open data in an introduction to database systems course," *J. Comput. Sci. Coll.,* vol. 28, pp. 156-161, 2013.

[6] T. T. Yuen and K. A. Robbins, "A Qualitative Study of Students' Computational Thinking Skills in a Data-Driven Computing Class," *Trans. Comput. Educ.,* vol. 14, pp. 1-19, 2014.

[7] College Board. (2014, December 2014). Computer Science Principles Performance Assessment. Available: <http://media.collegeboard.com/digitalServices/pdf/ap/ap-computer-science-principles-performance-assessment.pdf>

[8] A. C. Bart*, et al.*, "Using real-time web data to enrich computer science projects," in *Splash-E*, Indianapolis, Indiana, USA, 2013.

[9] A. C. Bart*, et al.*, "Transforming introductory computer science projects via real-time web data," presented at the Proceedings of the 45th ACM technical symposium on Computer science education, Atlanta, Georgia, USA, 2014.

[10] L. J. Barker*, et al.*, "Exploring factors that influence computer science introductory course students to persist in the major," presented at the Proceedings of the 40th ACM technical symposium on Computer science education, Chattanooga, TN, USA, 2009.

[11] J. Carter*, et al.*, "Motivating all our students?," presented at the Proceedings of the 16th annual conference reports on Innovation and technology in computer science education - working group reports, Darmstadt, Germany, 2011.

[12] B. DiSalvo and A. Bruckman, "From interests to values," *Commun. ACM,* vol. 54, pp. 27-29, 2011.

[13] A. Fisher and J. Margolis, "Unlocking the clubhouse: the Carnegie Mellon experience," *SIGCSE Bull.,* vol. 34, pp. 79-83, 2002.

[14] M. Goldweber*, et al.*, "A framework for enhancing the social good in computing education: a values approach.," *ACM Inroads,* vol. 4, pp. 58-79, 2013.

[15] B. D. Jones, "Motivating Students to Engage in Learning: The MUSIC Model of Academic Motivation," *International Journal of Teaching and Learning in Higher Education,* vol. 21, pp. 272-285, 2009.

[16] A. Bandura, "Social cognitive theory of self-regulation," *Organizational behavior and human decision processes,* vol. 50, pp. 248-287, 1991.

[17] M. Ben-Ari, "Constructivism in computer science education," *Journal of Computers in Mathematics and Science Teaching,* vol. 20, pp. 45-73, 2001.

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

[19] P. Steel, "The nature of procrastination: A meta-analytic and theoretical review of quintessential self-regulatory failure.," *Psychological Bulletin,* vol. 133(1), pp. 65-94, 2007.

[20] R. M. Klassen*, et al.*, "Academic procrastination of undergraduates: Low self-efficacy to self-regulate predicts higher levels of procrastination. ," *Contemporary Educational Psychology,* vol. 33, pp. 915-931, 2008.

[21] P. R. Pintrich*, et al.*, "Reliability and predictive validity of the Motivated Strategies for Learning Questionnaire (MSLQ). ," *Educational and Psychological Measurement,* vol. 53, pp. 801-813, 1993.

[22] Y. Lambrinidou*, et al.*, "Ethnography in Engineering Ethics Education: A Pedagogy for Transformational Listening," presented at the 121st American Society for Engineerng Education Annual Conference & Exposition, Indianapolis, IN, 2014.

[23] ABET Engineering Accreditation Commission. (2014, December 2014). Criteria for Accrediting Engineering Programs. Available: <http://www.abet.org/uploadedFiles/Accreditation/Accreditation_Step_by_Step/Accreditation_Documents/Current/2015-2016/E001%2015-16%20EAC%20Criteria%2011-7-14.pdf>

[24] Association for Computing Machinery. (2014, *Code of Ethics*. Available: <http://www.acm.org/about/code-of-ethics>

[25] Association for Computing Machinery. (2014, December 2014). *Software Engineering Code of Ethics and Professional Practice*. Available: <http://www.acm.org/about/se-code>

[26] E. A. Cech, "Culture of Disengagement in Engineering Education?," *Science, Technology, and Human Values,* vol. 39 (1), pp. 42-47, 2014.

[27] A. Korhonen*, et al.*, "Requirements and design strategies for open source interactive computer science eBooks," presented at the Proceedings of the ITiCSE working group reports conference on Innovation and technology in computer science education-working group reports, Canterbury, England, United Kingdom, 2013.

[28] T. Naps*, et al.*, "Evaluating the educational impact of visualization," presented at the Working group reports from ITiCSE on Innovation and technology in computer science education, Thessaloniki, Greece, 2003.

[29] T. L. Naps*, et al.*, "Exploring the role of visualization and engagement in computer science education," presented at the Working group reports from ITiCSE on Innovation and technology in computer science education, Aarhus, Denmark, 2002.

[30] J. H. McMillan and J. Hearn, "Student self-assessment: The key to stronger student motivation and higher achievement," *Educational Horizons,* vol. 87 (1), pp. 40-49, 2008.

[31] H. Andrade and A. Valtcheva, "Promoting learning and achievement through self-assessment," *Theory Into Practice,* vol. 48(1), pp. 12-19, 2009.

[32] L. Malmi*, et al.*, "Visual algorithm simulation exercise system with automatic assessment: TRAKLA2," *Informatics in Education,* vol. 3(2), pp. 267-288, Septmenber 2004 2004.

[33] V. Karavirta and C. A. Shaffer, "JSAV: the JavaScript algorithm visualization library," presented at the Proceedings of the 18th ACM conference on Innovation and technology in computer science education, Canterbury, England, UK, 2013.

[34] V. Karavirta. (2013, *JSAV github repository*. Available: github.com/vkaravir/JSAV

[35] B. Friedman*, et al.*, "Value Sensitive Design and information systems," in *Human-computer interaction in management information systems: Foundations*, P. Zhang and D. Galletta, Eds., ed Armonk, New York; London, England: M.E. Sharpe, 2006, pp. 348-372.

[36] U. Wilensky. January 2015). *NetLogo*. Available: https://ccl.northwestern.edu/netlogo/

[37] M. Guzdial, "A media computation course for non-majors," *SIGCSE Bull.,* vol. 35, pp. 104-108, 2003.

[38] M. Guzdial and A. E. Tew, "Imagineering inauthentic legitimate peripheral participation: an instructional design approach for motivating computing education," presented at the Proceedings of the second international workshop on Computing education research, Canterbury, United Kingdom, 2006.

[39] P. DePasquale, "Exploiting on-line data sources as the basis of programming projects," presented at the Proceedings of the 37th SIGCSE technical symposium on Computer science education, Houston, Texas, USA, 2006.

[40] D. Stevenson, E.*, et al.*, "Steganography and cartography: interesting assignments that reinforce machine representation, bit manipulation, and discrete structures concepts," presented at the Proceedings of the 36th SIGCSE technical symposium on Computer science education, St. Louis, Missouri, USA, 2005.

[41] D. E. Stevenson and P. J. Wagner, "Developing real-world programming assignments for CS1," presented at the Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education, Bologna, Italy, 2006.

[42] A. Erkan*, et al.*, "Sustainability themed problem solving in data structures and algorithms," presented at the 43rd ACM technical symposium on Computer Science Education, Raleigh, North Carolina, USA, 2012.

[43] A. C. Bart*, et al.*, "Motivating Students with Big Data: CORGIS and MUSIC," in *Splash-E*, Portland, Oregon, USA, 2014.

[44] H. Abelson*, et al.*, *Blown to Bits*. Upper Saddle River, NJ: Addison-Wesley, 2008.

[45] D. Rushkoff, *Program or Be Programmed*. New York, NY: OR Books, 2010.

[46] Executive Office of the President. (2014, Decmber 2014). Big Data: Seizing Opportunities, Preserving Values. Available: <http://www.whitehouse.gov/sites/default/files/docs/big_data_privacy_report_5.1.14_final_print.pdf>

[47] K. Davis, *Ethics of Big Data*: O'Reilly Media, 2012.

[48] ACM Committee on Computers and Public Policy and Peter G. Neumann (Moderator). Forum on Risks to the Public in Computers and Related Systems [Online]. Available: <http://catless.ncl.ac.uk/Risks/>

[49] K. Bowyer. December 2014). *Ethics and Computing*. Available: <http://www3.nd.edu/~kwb/nsf-ufe/>

[50] Center for Engineering Ethics and Society. (2007, January 2015). *Online Ethics Center for Engineering and Science*. Available: <http://www.onlineethics.org/>

[51] C. Douce*, et al.*, "Automatic test-based assessment of programming: A review," *J. Educ. Resour. Comput.,* vol. 5, p. 4, 2005.

[52] S. H. Edwards, "Using software testing to move students from trial-and-error to reflection-in-action," presented at the Proceedings of the 35th SIGCSE technical symposium on Computer science education, Norfolk, Virginia, USA, 2004.

[53] S. H. Edwards*, et al.*, "Experiences using test-driven development with an automated grader," *J. Comput. Sci. Coll.,* vol. 22, pp. 44-50, 2007.

[54] B. D. Jones, "The MUSIC model of academic motivation: User guide to assessing the MUSIC model components," ed. Blacksburg, VA., 2014.

[55] B. D. Jones, "Strategies to implement a motivation model and increase student engagement," presented at the annual meeting of the International Society for Exploring Teaching and Learning, Nashville, TN, 2010.

[56] B. D. Jones*, et al.*, "The effects of a collaborative problem-based learning experience on students’ motivation in engineering capstone courses," *Interdisciplinary Journal of Problem-based Learning,* vol. 7(2), 2013.

[57] B. D. Jones and G. E. Skaggs, "Measuring students’ motivation: Development of and validity evidence for the MUSIC Model of Academic Motivation Inventory," *The International Journal of Educational and Psychological Assessment,* In press.

[58] B. D. Jones and J. L. Wilkins., "Validity evidence for the use of a motivation inventory with middle school students," presented at the annual meeting of the Society for the Study of Motivation (Poster), Washington, D.C., 2013.

[59] P. R. Pintrich, "A conceptual framework for assessing motivation and self-regulated learning in college students," *Educational Psychology Review* vol. 16, 385–407 2004.

[60] A. R. Artino. (2005, September 28, 2012). Review of the motivated strategies for learning questionnaire. Available: <http://search.proquest.com/docview/62004605?accountid=14826>

[61] M. L. Cooper*, et al.*, "Open source software and the algorithm visualization community," *Science of Computer Programming,* vol. 88, pp. 82-91, 2014.

[62] E. Fouh*, et al.*, "The role of visualization in computer science education," *Computers in the Schools,* vol. 29, pp. 95-117, 2012.

[63] C. A. Shaffer*, et al.*, "Building an online educational community for algorithm visualization," presented at the Proceedings of the 41st ACM technical symposium on Computer science education, Milwaukee, Wisconsin, USA, 2010.

[64] C. A. Shaffer*, et al.*, "Algorithm visualization: a report on the state of the field"," presented at the SIGCSE '07: Proceedings of the 38th SIGCSE technical symposium on Computer science education,, New York, NY, USA, 2007.

[65] C. A. Shaffer*, et al.*, "Getting algorithm visualizations into the classroom," presented at the Proceedings of the 42nd ACM Technical Symposium on Computer Science Education - SIGCSE '11, Dallas, TX, USA, 2011.

[66] C. A. Shaffer*, et al.*, " Truly Interactive Textbooks for Computer Science Education," presented at the Proceedings of the Sixth Program Visualization Workshop, Darmstadt, Germany, 2011.

[67] C. A. Shaffer*, et al.*, "OpenDSA," presented at the Proceedings of the 11th Koli Calling International Conference on Computing Education Research - Koli Calling '11, Koli, Finland, 2011.

[68] E. Fouh*, et al.*, "Exploring students learning behavior with an interactive etextbook in computer science courses," *Computers in Human Behavior,* vol. 41, pp. 478-485, 2014.

[69] S. Hall*, et al.*, "Education Innovation for Data Structures and Algorithms Courses," presented at the Proceedings of ASEE Annual Conference, Atlanta, GA, 2013.

[70] E. Tilevich*, et al.*, "Creating stimulating, relevant, and manageable introductory computer science projects that utilize real-time web-based data (abstract only)," presented at the Proceedings of the 45th ACM technical symposium on Computer science education, Atlanta, Georgia, USA, 2014.

[71] E. Tilevich*, et al.*, "Creating stimulating, relevant, and manageable introductory computer science projects that utilize real-time large, web-based datasets (abstract only)," presented at the Proceedings of the 46th ACM technical symposium on Computer science education, Kansas City, Missouri, USA, 2015.

[72] J. V. Ernst*, et al.*, "Professional development system design for grades 6-12 technology, engineering, and design educators," presented at the American Society for Engineering Education Annual Conference and Exposition, 2013.

[73] J. V. Ernst, "Authentic assessment in performance-based stem education activities," presented at the Scaling STEM: Transforming Education Matters Annual Conference, 2012.

[74] L. Segedin*, et al.*, "Transforming teaching through implementing inquiry: A national board-aligned professional development system," presented at the Association for Career and Technical Education Research, 2013.