PROJECT SUMMARY

Overview:

Online educational systems, and the large-scale data streams that they generate, have the potential to transform education as well as our scientific understanding of learning. Computer Science Education (CSE) researchers are increasingly making use of large collections of data generated by the click streams coming from eTextbooks, interactive programming environments, and other smart content. However, CSE research faces barriers that slow progress: 1) Collection of computer science learning process and outcome data generated by one system is not compatible with that from other systems. 2) Computer science problem solving and learning (e.g., open-ended coding solutions to complex problems) is quite different from the type of data (e.g., discrete answers to questions or verbal responses) that current educational data mining focuses on.

We propose efforts to build community and capacity among CSE researchers, data scientists, and learning scientists toward reducing these barriers and facilitating the full potential of data-intensive research on learning and improving computer science education. We bring together CSE tool build communities with learning science and technology researchers toward a software infrastructure that supports scaled and sustainable data-intensive research in CSE that contributes to basic science of human learning of complex problem solving. We propose a set of community-building and infrastructure capacity-building support whose ultimate goal is to develop and disseminate infrastructure that facilitates three aspects of CSE research: (1) development and broader re-use of innovative learning content that is instrumented for rich data collection, (2) formats and tools for analysis of learner data, and (3) best practices to make large collections of learner data and associated analytics available to researchers in CSE, data science, or learning science. To achieve these goals, we propose to engage a large community of researchers to define, develop, and use critical elements of this infrastructure toward address specific data-intensive research questions. We will host workshops, meetings, and online forums leveraging existing communities and building new capacities toward significant research outcomes and lasting infrastructure support.

Intellectual Merit:

Our project is the first attempt to design an infrastructure that can support various kinds of research in CSE domain as a one-stop-shop, and is the first to focus on full-cycle educational research infrastructure in any domain. If successful, CSE tool developers and educators will be become more productive at creating and integrating advanced technologies and novel analytics. Learning researchers will have better tools for analyzing the huge amounts of learner data that modern digital education software produces. Data scientists will have rich new datasets in which to explore new machine learning and statistical techniques.

Collectively, these efforts can reduce barriers to educational innovation and support scientific discoveries about the nature of complex learning and how best to enhance it. We will support scientific investigations through community meetings and mini-grants to others addressing questions such as: What is the optimal ratio of solution examples and problem-solving practice? How do computational thinking skills emerge? In what quanta are programming skills acquired? Can automated tutoring of programming be effective at scale in enhancing student learning?).

Broader Impacts:

This proposal represents the first step toward building a community of practice that will broadly impact both computer science education and research on it. We aspire to have direct impact on enhancing scientific productivity of at least 20 computer science education researchers and at least 10 learning science researchers even at this early stage of the proposed work on the research infrastructure. Their discoveries and technological innovations will in turn help tens of thousands of students in the strategically important field of computer science. Many of the innovations proposed can directly impact learning in any discipline. Educational software will more quickly be developed in the future, that more easily generates meaningful learner data, which in turn can be more easily analyzed.

1 Motivation and goals

Computer Science was an early adopter of computing technology to support and advance the educational process.[1] For many years, Computer Science Education (CSE) researchers lead the application of computers in education by pioneering new approaches and tools such as the use of hypertext, eTextbooks, animation, and automatic assessment of student answers to problems. But progress in this field faces friction that must be overcome for each new advance. To further understanding of the pedagogical effectiveness of new tools, Computer Science researchers and educators need to collect and analyze larger volumes of educational data on a finer grain level. To make the best use of these tools requires both their interoperability and the integration of their data streams in ways that currently do not occur.

On one hand, state-of-the art learning content (that we refer to as *smart content*) is interactive. This opens an opportunity to collect larger volume of fine-grained educational data. On the other hand, there is no widely used infrastructure to support sufficient interoperability of smart content or the collection of student data. Without proper infrastructure that supports content and data interoperability, each CSE research team has to encapsulate its novel smart content in a self-contained system. Each group pays the overhead of supplying core required subsystems of identifying users, storing scoring and log data, and exposing this information to instructors. In essence, each such tool is building a mini-LMS. We can cite dozens of modern CSE systems and environments, each one usually offering one kind of content that supports one aspect of CSE: a collection of animated algorithms, an application for automatic assessment of programming assignments, a practice system offering one kind of programming problems, to name a few. While each tool might look like a success, the current situation creates obstacles for the progress of research on both human learning and educational technology.

Most important in the context of this proposal, it is hard for learning scientists and computer science education researchers to collect and analyze student learning data needed to advance their knowledge on how students learn and how learning is affected by different learning tools and pedagogical approaches. For each aspect of the educational process we can name smart content solutions that support this aspect. But encapsulating this content in isolated systems prevents both the collection of data and a sufficient level of dissemination and impact that is critical for getting a sufficient volume of data. It is impratical for instructors to install, explain, and use several different systems in any one course. But using one tool that can log student behavior only for that tool leaves the rest of the student learning process invisible to researchers. When more than one tool is used, the collected data are hard to integrate. It is also hard for CSE researchers to gain access to meaningful experimental data on novel educational tools beyond their own classes. So they need to achieve considerable scale and course coverage to gain measurable results. Small tools, even highly innovative, have fewer opportunities to be broadly used and evaluated. With the lack of collected data, it is hard to understand the process of learning, learn from experience, and introduce more efficient tools and technologies based on this understanding. Since educational data are important for advancing research in several areas from learning technology to educational data mining to learning sciences, the lack of data hinders research progress in these fields.

The ACM ITiCSE working group on the use of smart content in CSE [4] considered this situation

[1] ACM Special Interest Group on Computer Science Education is one of the early SIGs. The SIGCSE Bulletin was established in 1968. The annual SIGCSE conference started in 1970, publishing numerous papers on technical innovations for CSE.

and argued that the problems could be resolved by introducing a software infrastructure that can support scaled and sustainable data-intensive research in CSE. The working group stressed required aspects such as re-usability of smart content and fine-grained centralized collection of student data. An infrastructure for data collection and data-intensive research in CSE is important, however, well beyond the community of Computer Science educators. Learning science researchers need these data to gain research insights into the needs of CS learners, discover learning science principles relevant to improving CSE, develop advanced technologies that address student needs and leverage learning science principles, and test those advanced technologies in real classes and make valid inferences about what works.

Our project attempts to realized and extend the vision of the Working Group by considering the needs and prospects of both the CSE and Learning Science communities. Engaging researchers from both fields, we are designing an infrastructure that can support both traditional technology-focused CSE research as well as human-focused learning science research in the area of CSE. We envision a scientific infrastructure for CSE research that facilitates a wide variety of researchers and instructors in exploring research questions of scientific and practical interest. The following is a sampling of such questions that our vision could help researchers address.

1. **How do we broaden participation?** Sub-questions include:
   (a) What causes lack of participation among underrepresented populations in CS?
   (b) How do factors like experience, motivation, and pedagogy affect participation?
2. **How do we automatically assess coding problems to provide more feedback?**
3. **How does automated feedback affect the learning process?**
4. **What are the quanta of programming and other computer science knowledge acquisition and knowledge transfer (often referred to as concepts, skills, or knowledge components)?**
5. **How do students learn fundamental computational thinking concepts like state, algorithmic process, representation, and abstraction?**
6. **How do students develop the management skills needed to successfully complete medium and large programming assignments?**

2 Towards an Infrastructure for Data-Intensive Research in CSE

For the past decade, course materials have increasingly moved online, often organized within learning management systems (LMS). Course pedagogy has been augmented by various interactive services, such as discussion forums, chats, and wikis, which might also be integrated into the LMS, to provide communication and collaboration among learners, and between learners and teachers. A more powerful breed of online resources, that we refer to as Smart Learning Content (SLC), has emerged during the past 10+ years. SLC goes well past prior approaches, to provide a higher level of interactivity and engagement. It includes feedback adapted to the learner, such as learner-controlled animation, dynamic visualization, and learner-led simulation. The interactivity of SLC is important from the perspectives of both CSE and Learning Science communities. Studies in several domains demonstrate that interactive support for active student learning results in significant improvements in student learning [32, 33, 38, 44]. Studies of SLC in CSE also confirm high educational effectiveness for these novel learning technologies. As a bonus, the richer interaction log data produced by learners while interacting with SLC enables learning scientists to understand student learning and cognition on a much finer-grain level than it was possible with traditional learning content.

Examples of SLC within the CSE domain include program visualization and simulation tools [52], algorithm animations [50], programming problems [5], automatic assessment services for programming exercises [12], intelligent tutoring systems [49], and various forms of personalization and navigation support tools [20]. To support data collection, personalization, and sophisticated forms of feedback, SLC is most often offered through small content-focused systems that authenticate the user, models the learner, aggregates data, reports to the instructor, and supports some form of learning analytics. As mentioned above, the encapsulation of SLC within small-scale systems requires both a lot of duplication of effort and reduced ability to share the data collected.

There have been past attempts to address problem related to reuse of SLC, and problems with collecting large volumes of educational data produced by SLC. In particular, research-oriented infrastructures such as KnowledgeTree [3], MEDEA [54], and APeLS [7] were proposed for integrating multiple kinds of SLC into LMS-like systems while supporting data collection, learner modeling, and personalization. The LTI standard from IMS [9] has emerged as a mechanism to support SLC integration into LMS. While this standard does not resolve the critical problem of representations for learner data, it does provide a communication channel that can be used to develop more advanced infrastructures for smart content integration within CSE. An example of a system that takes advantage of this interoperability is OpenDSA—a framework for developing electronic textbooks that can include SLC [14].

The problem of centralized learner data collection was originally addressed in the field of user modeling and intelligent tutoring systems by introducing user modeling servers such as Personis [28] and CUMULATE![58]. These servers offered a standard protocol that can be used by any learning tool to report rich data about its interaction with each student to the centralized store. Since the key goal of the student modeling servers was to maintain an up-to-date runtime model of every user for the purpose of personalization, this work paid specific attention to deriving student knowledge from data and offering efficient access to student models. Later, the emergence of learning analytics and educational data mining emphasized the importance of centralized data collections well beyond its original use in personalized learning. It led to several new proposals for data collection standards, such as MOOCdb [55] or DataShop server [29]. Two important emerging standards are the Experience API [35] developed by ADL, and Caliper [8] from IMS.

Archival data collection solves only a part of the needs. It is also important to provide support to the research community in the form of affordances to analyze the data. A notable project for collecting learner data and building analytics around it is Carnegie Mellons LearnSphere project, which is currently support by an NSF Cyberinfrastructure grant **(CISE-ACI-1443068, 2015-2020, $5M)**. LearnSphere is creating data infrastructure building blocks to integrate the sharing and use of educational data and learning analytic methods. It has facilitated discoveries indicating a six times bigger relationship to learning outcomes from online active doing with feedback than from reading online text or watching online videos [32, 33]. LearnSphere integrates across a number of existing data repositories, including educational technology clickstream data (in CMU's DataShop), massively open online course data (in MIT's MOOCdb and Stanford's DataStage), and student writing and discourse data (in CMU's DiscourseDB). This integration is fueling a wide variety of research on learning science and technology [49, 2, 25, 31, 37, 57]. LearnSphere's DataShop [29] stands as the world's largest open repository for educational technology data, with over 1300 educational technology datasets supporting over 125 data mining or secondary data analysis studies. This project will leverage the integrated analytic infrastructure that LearnSphere provides toward extending it for analytics specialized for computer science education. We have begun to make progress in this direction in that LearnSphere's DataShop currently contains seven computer science education datasets including about 3.2 million data points contributed by 10,700 students. While this data can be downloaded, there are only a few available analytic methods that are directly relevant and the ones that are (e.g., learning curve analytics) require much special purpose processing, for example, to transform program solution submissions into incremental graded solution steps (cf. [48]). We propose to engage the Computer Science Education community in developing new analytics that are specific to questions of their interest.

Every relevant piece of work we have discussed above when taken individually serves its purpose well. However, despite the recognized need, there is no software infrastructure that brings individual pieces and parts of the process together providing a "one-stop shop" to radically increase opportunities for open experimentation and data analytics for CSE researchers and learning scientists. This proposal suggests a set of community-building, design, and research steps that are necessary to enable future development and broader use of a *infrastructure for data-intensive research in CSE*. To achieve this goal, we propose to bring together leaning scientists, learning technologists, and CSE practitioners in defining, designing, and demonstrating proof-of-concept for critical elements of this infrastructure. We believe that the results of our work can significantly advance research progress in the fields of CSE and Learning Science.

3 Research Questions in CS Education and Learning

We now present a brief analysis of the research community that we want to support, and then discuss a range of research questions that can be facilitated by the proposed infrastructure. As mentioned previously, the problems related to teaching and learning CS topics are the focus of both the community of CSE researchers and the community of Learning Science researchers (see Figure 1). The first community focuses on developing and evaluating efficient pedagogical approaches and learning tools for CSE. The interests of this community are represented by ACM SIGCSE with its conferences (SIGCSE, ITiCSE, ICER), providing space to share and discuss research results. The Learning Science community has chiefly focused on studying and modeling human learning and developing and evaluating advanced techniques and technologies to enhance learning. This group includes a wide range of researchers from psychologists and cognitive scientists to computer scientists and other technologists. Their interests are represented by several societies and corresponding conferences such as Cognitive Science, International Conference of the Learning Sciences, Artificial Intelligence in Education, Educational Data Mining, Learning Analytics and Knowledge. Within this large community, our project targets researchers who focus on Computer Science as a learning domain. Many Learning Science researchers are also interested to apply the knowledge of human learning to developing better technology and pedagogy, thus exploring topics that are of interest to both the CSE and LS communities. Following the community interests reviewed above, research questions in CSE and learning could be classified into questions focused on developing efficient pedagogical approaches and tools, questions focused on understanding how humans learn computer science topics, and questions that integrate the understanding of human learning with improving the efficiency of education.

A typical research question from the CSE group is whether a specific technology-based pedagogical approach or a novel tool positively impacts one of the target educational parameters such as

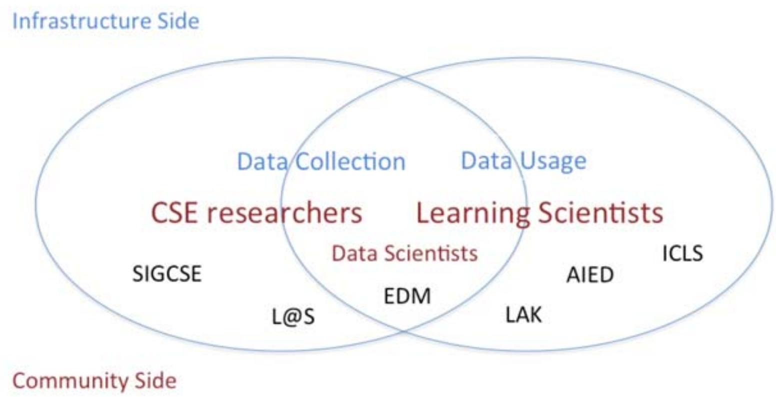

Figure 1: Research Communities engaged in collection and analysis of educational data in Computer Science Education. We will build capacity in overlapping communities of researchers (educators, data scientists, learning scientists) by tapping into and linking across the conference communities (SIGCSE, L@S, EDM, etc.) in which they participate.

improving learning gains, increasing student engagement and retention, or broadening participation. For example, Naps et al. [42] investigated whether adding prediction steps to algorithm visualization increases student engagement and understanding. Hsiao et al. [21] examined how a social comparison interface affects student problem solving performance and motivation. OpenDSA [13, 14] studied the impact of visualization and interactive algorithms simulation exercises on improving learning. While the majority of this research is focused on a single novel educational tool in isolation, some studies supported by early infrastructures were able to examine how students learn while working with multiple tools. For example, Hosseini et al. [18] evaluated the comparative impact of two kinds of interactive program examples used in parallel in a Python programming course.

Research questions in the Learning Sciences community focus on deeper issues related to knowledge, learning, and cognition. In this research, CSE is frequently considered as an interesting and challenging context to answer research questions that could enhance our understanding of human learning well beyond Computer Science. For example, Koedinger and colleagues [34, 30] have worked on data-driven improvement for conceptualization of a knowledge quanta model used in a mastery learning environment such as an intelligent tutor. A semi-automated search algorithm has been shown to significantly improve the accuracy of modeling student knowledge acquisition across several domains, including mathematics, statistics, languages, but not computer science. As another example of a Learning Science question that pertains to the use of computer-assisted educational technology at scale, Ritter and colleagues [47] show that middle-school and high-school teachers higher rate of skipping ahead in a Carnegie Learnings Cognitive Tutor has a tangible negative effect on student performance later in the school year.

Research questions at the crossroads of Learning Sciences and CSE attempt to bridge the gap between the two fields. Frequently these questions focus on assessing how different learning tools based on current understanding of human learning affect student learning process and its outcomes. Answering these research questions can lead to both creating better learning tools and processes and improvement of our understanding of human learning. For example, Trafton and Reiser [53] attempted to find which combination of problems and worked-out examples is most efficient for learning programming. They designed several combinations of these, driven by two competing theories on how students learn from examples, and compared them in a classroom study. The results of the study not only helped to find the most efficient approach to use examples in programming classes, but also provided supporting evidence for the competing theories.

While the sample research questions examined above are considerably different, in all cases the ability to answer them requires the ability to organize classroom or lab studies that engage students in working with different educational tools, to collect learning data, and to extensively analyze these data. Our goal is to radically improve all components of this process. This will make it easier to answer existing research questions while also opening the way to a new generation of research questions that require a richer variety of learning content, extensive fine-grained data collection, and more advanced data analysis. By lowering the threshold to conducting state-of-the-art research on CSE and providing a broader access to learning data, it will engage a much wider research community and speed up progress in this area of research.

4 The Target Software Infrastructure and its Components

The ultimate target of our work is to engage a broad research community in discussing, designing, and prototyping an infrastructure that facilitates a broad range of research on Computer Science Education and human learning. We seek to support all aspects of the research process—from planning and organizing research studies with a rich variety of educational tools to data collection and analysis. As a starting point for our infrastructure work we will use the proposal presented by an ITiCSE Working Group that examined the problems of adoption of SLC [4]. That working group included 13 CSE researchers (including two project PIs and several external collaborators) pursuing a broad variety of research approaches in the field. Through a community-based process we will refine the proposed infrastructure based on the analysis of stakeholders' needs, existing solutions and best practices, while also reaching agreement on several data representation and archiving standards that were not discussed by at ITiCSE.

The infrastructure is based on three best practices in the field: interoperability of SLC, data standards, and broader access to research data. The Working Group proposed to support these functionalities by organizing the infrastructure into three layers: the delivery layer, the application layer, and the data layer (Figure 2). The delivery layer is formed by various student-facing SLC clients. The application layer is composed of multiple SLC servers that deliver various kinds of SLC, and receive back performance and click stream data. The data layer hosts all kinds of Learning Record Storage systems that collect and store information about students' work.

Interoperability of SLC is supported by the separation of delivery platforms and SLC servers, which breaks the currently predominant encapsulation of SLC in stand-alone systems. By delivery platforms, we mean learning management systems (LMS), MOOC platforms, or other course delivery mechanisms. To increase the reusability of SLC, it should not be encapsulated in a single stand-alone system, but being hosted on independent SLC servers and repositories. The separation of SLC servers from delivery platforms should be based on a standard communication interface.

Data standards facilitate the collection of data about learners' interactions with SLC. Smart content typically engages users in interactive work and can produce a rich interaction trace. This trace is critical to collect for both research and practical needs, but LMS have do not store it. At the moment, these rich traces are typically either not collected or are encapsulated in standalone formats and analysis tools. To address data collection problems, the infrastructure introduces

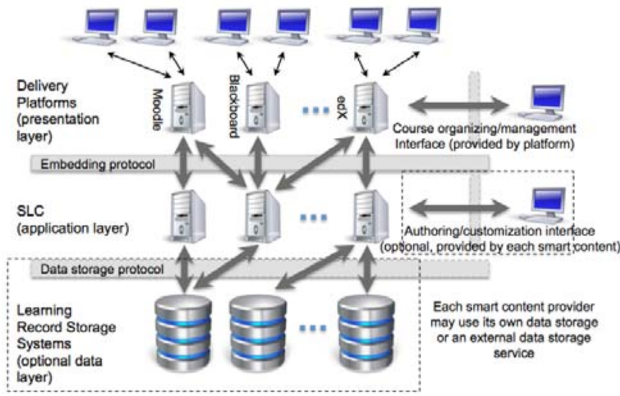

Figure 2: A three-tier infrastructure for flexible reuse of SLCs and data collection.

independent learning record store (LRS) systems. The idea of an independent component for collecting user data has been already explored in the field of adaptive educational systems (where it is known as student model or user model servers) [28, 58]. It has also been advocated by DoD Advanced Distributed Learning lab (ADL) as a part of their new Total Learning Architecture [36]. LRS systems support a standard data reporting protocol to collect data (i.e., xAPI) as well as standard data access protocol. In our model, the SLC servers that communicate directly with the client will then be responsible for delivering the learning records in the agreed-upon formats.

Broader access to research data ensures the ability of the broader research community to access and analyze the data. This increases the value of the collected data by supporting research data analysis and educational data-mining well beyond what the original tool developers had envisioned or could achieve on their own. The data access and analysis component of the target infrastructure will integrate and expand LearnSphere. LearnSphere has a basic learning data analytics and data visualization toolset that has been actively used by researchers in the past (e.g., [1, 56, 2, 25, 31, 37, 48, 49, 57]) and will be generalized or expanded to address the needs of CSE researchers. For example, LearnSphere provides customizable analytical methods for predicting student performance and learning outcomes from behaviors in online activities within a course [33]. These methods could be adapted to address questions of broadening participation and what student and instructional factors yield greater course completion and better acquisition and retention of computer science knowledge and thinking skills. An important contribution of LearnSphere is the ability to examine learning on the level of elementary knowledge components (KC). While there has been progress in applying existing KC-level learning analytics to computer science data [22, 48] there are many open challenges and opportunities in applying advanced learning analytics to computer science. LearnSphere provides a set of empirical methods for evaluating these knowledge component hypotheses.

The CMU team is actively building an analytic extension to LearnSphere, called Workflows. Workflows offer a collection of data import, processing, and data mining components. We will build LearnSphere Workflow components specifically focusing on analyzing CSE learning data. These components could include automated code parsers for multiple languages to extract programming constructs, fitting predictive models of student learning for varying granularity of knowledge quanta vocabularies, etc. We will also use the LearnSphere analytic workflow authoring tool to build out methods that can provide automated support for grading and tagging programming solutions and other open-ended solutions (e.g., problem or proof solutions in a machine learning class).

LearnSphere has already been used for storing learning data coming from CS courses, and for supporting scientific investigations of CS learning ([48]). Building upon the experience of the LearnSphere team, we plan to use its capabilities for long-term storage and management of the data and analytic methods this project generates.

5 The Community of Practice

Building a community of practice centered on the target research infrastructure is the principle goals of our proposal. The community of practice is vital for the adequate design of the infrastructure as well as for its broader support and sustainability beyond the design and development stages.

We plan to start building a community of practice by engaging collaborators into the design, prototyping, and piloting the components of the infrastructure. We have identified the first cohort of collaborators among those who already have solid experience in topics that are critical for designing the infrastructure. Many of these prospective collaborators were contacted on the stage of proposal preparation and expressed their interest to collaborate in attached letters. We expect that more collaborators will join the effort over the course of the project.

We will use several approaches for community engagement: working groups focused on specific issues (such as data collection and representation for a specific category of smart CSE content), focused working meetings, extended project meetings, open workshops, and direct support through min-grants. We expect that much planning and design work will be done by working groups, with proof of concept provided by tool builders supported through the mini-grants, and then the design-prototype cycle reviewed and evaluated during a "reflection" phase involving both focused and extended meetings. As progress is achieved on specific aspects of the infrastructure, we will focus on publicizing it through open workshops, standing conferences in the CSE and LS communities, and other dissemination channels. To support both sides of community building, we request funding for mini-grants and workshops, as described below. The end result will be an active Community of Practice centered around various aspects of the infrastructure such as data collections, reusable smart content, and a data analysis framework.

5.1 Mini-grants for engaging partners and collaborators

A primary component of our community building effort will be proactive outreach to various members of the CSE and LS communities who have a track record of developing SLC tools, data collection, and data analysis of CSE data. We already have letters of support from a number of such potential collaborators. Our budget proposal includes a pool of money that will be redistributed in small amounts to such collaborators, that we term mini-grants. The requirement is that they use some component of the infrastructure in some way, with our active support, and give us feedback on the experience. Examples include modifications to their tools to interoperate with existing tools that support the infrastructure, changes to their interaction data formats to conform to existing data analysis toolkits within the infrastructure, use and testing of such data analysis tools, or use of databases from third parties—as provided in the infrastructure—for CSE research. These mini-grant recipients will then play key roles in the working groups described in the next section.

Our budget allows us to provide consulting money to such collaborators, typically at the rate of $2000 per semester. The OpenDSA project has successfully used this model to improve the level of evaluation and feedback provided by key adopters. While the amount of money received by any such collaborator is small compared to standard research grants, for many faculty, especially at smaller colleges, just that little incentive is enough to overcome the hurdle of getting started on such a collaboration.

As we generate broader collections of student analytics data, the possibilities of discovery research while mining the data improves. As an example of the type of work that large databases make possible, consider the work of Shaffer and Edwards. For many years, they had been collecting data that proved a correlation between starting early on larger programming projects and high scores. Edwards was finally able to demonstrate [11] a causal relationship through a retrospective analysis of Web-CAT submission data. Retrospective in that the data had not been collected as part of a deliberate effort to answer this question, but rather that the question was answered by examining an existing data set collected as a natural part of using some CSE tool. In particular, it was determined that students who sometimes perform well and sometimes do not tended to work early on those projects where they performed well, and did not tend to work early on those where they did not perform well. This is just one illustration of the type of mining for relationships that could be possible if we can collect and make available to researchers the wealth of data coming from SLC components.

One special partner for our effort will be Ensemble, the NSDL portal for Computer Science Education [51]. Two of us (Brusilovsky and Shaffer) were part of the original Ensemble team or provide significant contributions to site. We will work with Boots Cassell, PI for Ensemble. Various collaborations include using Ensemble for community communications; providing information through Ensemble about various SLC tools, tutorials, and infrastructure that others can use; and generally using Ensemble as the distribution portal for information about the infrastructure effort.

5.2 Working Group and Workshops

Working groups and workshops will serve as broader community engagement mechanisms on different stages of the project. The working groups will have a task to design and possibly prototype corresponding components of the infrastructure. On the earlier stage of the project, we convene several working groups to discuss interoperability details, data collection formats, data analysis approaches. For example, a number of systems exist that auto-grade small programming exercises. In general, these systems work by providing a problem statement, starting code for the student, a collection of test cases, a model answer (that provides the solutions to the test cases), and “wrapper” code that embeds the student’s answer to make an executable program. It would be useful to the community to have a standard representation for such programming exercises, so that they can be shared in various ways. Our collaborator support fund would be used to identify volunteers to lead working group, perhaps in conjunction with a national conference such as SIGCSE, to make progress on defining standards that the community can use. Individual working groups will communicate through online or in-person focused meetings. We also expect that all working groups will meet together annually at the joint project meeting to discuss the progress, coordinate work, and plan next stages of design. Funding to support working group members to attend the annual meeting is requested.

Once the project reaches its early dissemination stage, we will focus on workshops and similar dissemination forms. We will organize annual open workshops in conjunction with major CSE and learning analytics conferences like ACM SIGCSE and Educational Data Mining. We will leverage these to engage at least 25 new CSE researchers and at least 10 new learning science and data analytic researchers each year with the project. We will actively advertise the work done on this project among the members of all engaged communities. All PIs have extensive experience in organizing workshops at key conferences. In addition, the LearnSphere team at Carnegie Mellon has more than a decade of experience developing a community of practice focused on a component key to this proposal learning analytics tools for educational researchers. The team organized an annual LearnLab Summer School. One of the tracks of the school is specifically dedicated to investigations of learning data Educational Data Mining. As part of that track, school participants are trained to use LearnSpheres predictive learning modeling capabilities and data visualizations. Educational Data Mining community feedback to LearnSphere Workflows was recently solicited at a dedicated workshop titled "Educational Data Analysis Using LearnSphere" in conjunction with the 9th International Conference on Educational Data Mining in 2016. A bridge between this project to the Learning Sciences researchers will serve as one of the pillars of building an overall CSE community of practice and would attract more learning scientists to look at the student data coming out of CS courses be it as a result of this project or otherwise.

6 Project Outcomes and Evaluation

The overarching goal for this project is to advance the progress of CSE research through building the community of CSE tool builders and data analysts. As such, metrics to evaluate the success of our project are focused on what changes we can make in the broader CSE research community to bring them on board. We seek objective measures of community involvement related to each of our primary foci: interoperability of SLC and LMS such that data can be collected, formats for data in key sub-communities, support for tools and methods of analysis of student analytics data, and use of analytics databases by researchers other than the ones who created them originally.

At one level, the success of our community engagement efforts can be measured by some simple metrics, such as the number of SLC tools that can provide data in the proper formats, number of data sets produced, number of cases where researchers use data from other groups for their analysis, number of participants in our focused working groups, and number of participants in our larger meetings. More qualitatively but of deeper importance, we can try to count the following occurances.

- Number of SLC tools from outside collaborators that become inter-operable.
- Number of collected and contributed datasets.
- Number of users and contributors to the data analysis tools.
- Number of research studies and results enabled by our support for interoperability and data analysis tools.

The ultimate evidence of impact to be produced by the target infrastructure will be the number and quality of "success cases" that can be achieved during the grant period and also in the sustainable character of the community such that many more such success cases will follow. For us, a success case has the following features.

1. A research question is pursued (involving collaborators not in the research groups of the Co-PIs) that is distinctly facilitated by our work.
2. Prior to our intervention, there is one or more barriers to addressing this question because of lack of infrastructure and the necessary standards.
3. Efforts of this project demonstrably removes that barrier.

4. The collaborating researcher successfully addresses the research question using our infrastructure or analysis tools, and produces novel scientific results.

Following are specific examples of how our infrastructure will facilitate innovative data instensive research and scientific discovery by CSE and Learning Science researchers. The examples given involve researchers who have supplied letters of collaboration.

Success case example from Norman Bier.

1. *Research question.* To what extent does instruction in Computational Thinking result in learning of broad computational thinking concepts, or does such thinking primarily emerge from the practice of skills like programming and algorithm development?
2. *Barriers.* a) It is difficult to share high quality online assessments, assignments, and activities among CS educators. b) Most such online activities do not generate log data with sufficient semantic tags that can be used to identify underlying concept or skill demands and whether or not student performance indicates they are meeting those demands. c) We lack of access to data on follow-up courses.
3. *Barriers removed.* Greater community support, interoperability, and sharing of data generation and analysis methods will yield sharing of high-quality, online activities that are instrumented for quality data collection. LearnSphere will facilitate sharing of analytics that integrate across different kinds of learning process data as well as different kinds of shorterm and longterm outcome data.
4. *Research outcome.* Data intensive research will produce scientific discoveries on the nature and transferability of computational thinking skills and insights for improving CS Education.

Success case example from Lauri Malmi.

1. *Research question.* What is the impact of different kind of worked-out programming examples (such as annotated examples and animated examples) on student ability to solve various programming problems.
2. *Barriers.* No existing systems provide several kinds of programming examples and problems for CSE. As a result, little futher progress on this topic has been made since the classic work of Trafton and Reiser [53].
3. *Barriers removed.* Community support and better system and data interoperability will aid the integration of smart content from activity servers providing code examples (such as worked out code examples via NavEx [6]) with one serving programming problems (such as programming problems supported by the system CloudCoder developed by our collaborators Spacco and Hovermeyer [45]). Both will be instrumented for data collection, including facilitating controlled experimentation of different ratios and orderings of examples and problems.
4. *Research outcome.* Data intensive research will help produce scientific output (including publications) that reveals which kind of examples provide stronger impact on student ability to sevre programming probles. Potentially, the impact of examples could depend on the user current level of knowledge (that could be more reliably estimated with a more complete trace of student activities), with novice learners having larger need to work with examples in order to succeed in problem solving. Also, the impact of work with examples on problem solving could depend of specific example-problem pairs with specific type of examples providing best support for specific kind of problems.

7 Work Organization and Project Schedule

The project work will be organized around two sides of the target infrastructure, data collection and data usage. The data collection components should support the ability to broadly integrate various types of SLC while enabling it to generate detailed educational data in standard formats that are archived for later use. The data usage components should support storage, archiving, and various types of analysis of the collected data by different research communities.

To advance the work on designing data collection and representation components of the infrastructure, we will organize several working groups focused on different types of educational data collected by SLC tools and on research questions that can be addressed with analysis of this data. The groups will include representatives from CS Education, Data Science, and Learning Sciences and will represent the interests of both data generators and data analyzers. Examples of such working groups include (1) *program snapshots* collected by different tools that support students working on programming problems, either a larger scale such as Web-CAT [12], or at smaller scale such as CloudCoder [45], (2) *example exploration traces* collected by SLC tools that support student work with algorithm or program visualizations, and (3) *problem-solving traces* such as those collected by tools for Parson code-construction problems [17]. These types of data have emerged as an important source of data-driven research in both CSE and Learning Science communities. For example, program snapshots have been used for examining individual differences between students [19] and for building intelligent support for problem solving [43, 48, 49]. Each working group meeting will include representatives from tool developers and CSE researchers who have past experience with this type of data as well as data analysts and learning scientists interested in this type of data. Taken together, our research team and our external collaborators who already provided support letters include several key players in the area of program snapshots (Edwards, Spacco, Petersen, Howemeyer, Hellas, Rivers, Yudelson), examples (Malmi, Hosseini, Shaffer, Yudelson, Brusilovsky, Naps), and problems (Ihantola, Barnes, Hsiao, Brusilovsky, Kumar).

The working groups will focus on discussing, designing and providing proof-of-concept for communications, data collection, and data as well as research questions and analytic goals for the specified types of content. The designs and prototypes will be piloted by developers of existing SLC tools; they will produce samples of educational data for further consideration. The results of the pilots will be discussed at the joint working meetings that will bring together researchers from the CSE and Learning Science communities, and builders of SLC tools. This meeting will run annually in early Spring in association with the ACM SIGCSE conference. Pilot data and group discussions will serve as an input for the next design cycle. Over the course of the project we expect to run three iterations of the design-pilot cycle, with Spring focused on design, Summer focused on demonstrating proof-of-concept, and Fall focused on reflecting on lessons learned from the previous stages. The final design proposals will be prepared and distributed at the end of the last design round in Spring and Summer 2020, with a final summary meeting to examine lessons learned during the entire project in Summer 2020.

The CMU group will support researchers and educators in the working groups to formulate research questions to pursue through data analytics and especially to design and develop analytic strategies to address such questions toward novel scientific discovery. We will leverage the many resources available through LearnSphere and especially support the community in using and sharing analytic components through the web-based workflow authoring tool that LearnSphere provides. A key to the vision is community standards for data formats of CS learning activity and outcome data will be emergent not only from collaboration among working group members, but especially

	2018 Spring	2018 Summer	2018 Fall	2019 Spring	2019 Summer	2019 Fall	2020 Spring	2020 Summer
Tool Working Groups	Design: Snapshots; Traces	Proof of Concept: Snapshots; Traces	Reflection: Snapshots; Traces	Design2: Snapshots; Traces; Parsons; TBD	Proof of Concept2: Snapshots; Traces; Parsons; TBD	Reflection2: Snapshots; Traces; Parsons; TBD	Community expansion	Dissemination: lessons learned
Data Generation Milestones	Generation design	Generation proof of concept	Generation reflection	New generation design	New generation proof of concept	New generation reflection	Generation tool sharing & reuse	Document & disseminate
Data Analysis Milestones	Analysis design	Analysis proof of concept	Analytics reflection	New analytics design	New analytics proof of concept	New analytics reflection	Analytics sharing & reuse	Document & disseminate
Meetings	Kick-off: SIGCSE 2018	Data Focus: EDM 2018 or AIEd	Reflect Meet: Pitt/CMU or VT	CSE Meeting: SIGCSE 2019	Data Focus: EDM 2019 or AIEd	Reflect Meet: Pitt/CMU or VT	Dissem.Meet: LAK or L@S	Results: Pitt/CMU or VT

Figure 3: Our timeline targets data generation and data analysis milestones to be achieved by the community during and between working group meetings three times per year.

from the pull to reuse powerful existing or emergent analytic strategies and methods.

The project is a collaboration between three core research teams and multiple external partners who bring together critical expertise that is necessary to succeed in this challenging endeavor. Each team will primary responsibilities with most tasks supported by more than one team. On the discussion, design, and proof-of-concept side, the Pitt and Virginia Tech teams will focus on smart content interoperability and data collection. This continues their existing collaboration from the ITiCSE working group on Smart Content where Brusilovsky and Edwards served as co-organizers. The CMU team will focus on data processing, archiving, and analysis aspects, leveraging its extensive experienced maintaining DataShop and LearnSphere systems. The CMU and Pitt teams will collaborate on designing data collection and storage formats. The Virginia Tech team will focus on community building among the tool builder within the CSE community, while Pitt and CMU will work with the data analytics and Learning Science communities.

While any given working group is likely to have five to ten participating members, a key part of advancing the work is the mini-grants described in Section 5. Within each working group, one tool (developed by someone **not** a part of the CMU, Pitt, or VT teams) will be selected to deploy proof-of-concept for communications and dataset generation. The mini-grants will serve as seed money to encourage active participation by the selected collaborator. The Virginia Tech team will take the lead in working directly with the collaborating tool builders to make the necessary modifications to the associated SLC tool to deliver data in a suitable format for use by the analysis tools. The CMU team will take the lead in working with the collaborating data intensive researchers (some of which overlap with CSE tool builders and some that do not) to support them in developing or extending analytics and turning insights form analytics into both research products and educational innovations.

A schedule of major research activities is shown in Figure 3. We plan for four general themes and associated working groups organized around sub-communities interested in generation and analysis of data associated with different kinds of instructional tools. In the first year, we will start two such working groups. Since there already exists the beginnings of a community of practice organized around program snapshots [46], this will be the first target for a working group and one or more mini-grants for data generation and/or analysis. The working group and associated mini-grants will enhance the generation of program snapshot data and will improve upon existing analytic progress already achieved for program snapshots [19, 43, 48, 49]. The second working group will target the exploration traces of various program examples as generated by use of algorithm or program visualizations and interactive worked-out example exploration systems. These first two groups would have two full annual cycles of design-prototype-reflect. The third and fourth groups will start no later than the second year. A candidate for the third group is small programming problems such as Parsons program construction problems [17] and parameterized semantics problems [20]. The fourth group is deliberately left to be decided later as we get feedback from the community. One possibility is to focus on genearation and analysis of learning outcome data both in shorter time scales, like course unit quizzes and course final exams, and in longer time scales, like subsequent CS course performance or even job performance. We will hold meetings three times a year that will bring together interdisciplinary researchers from Computer Science Education, Data Sciences, and Learning Sciences with interests in either or both generating and/or analyzing computer science data. As shown in Figure 3, some of these meetings will be associated with relevant conferences so as to lower barriers to participation from relevant communities (e.g., SIGCSE attracts computer science educators, EDM and LAK attract data scientists interested in learning data, AIEd and L@S attract learning science and technology researchers).

All Co-PIs will participate in each major activity according to the general split of responsibilities described above. The overall project coordination will be provided by Brusilovsky. See Management and Coordination Plan for more details.

8 Intellectual Merit

Our project is the first attempt to provide a range of CSE research support functionalities as a one-stop-shop, and is the first to focus on full-cycle educational research infrastructure in any domain. If successful, CSE tool developers will be become more productive at creating and integrating advanced technologies and novel analytics. Learning researchers will have better tools for analyzing the huge amounts of learner analytics that modern digital education software produces. Researchers without direct access to a pool of students will be able to explore data sets of learner data from a broad range of institutions. Our success will also pave the way (and provide components) to creating similar infrastructures in other educational domains. Collectively, these efforts can reduce barriers to educational innovation and support scientific discoveries about the nature of complex learning and how best to enhance it, and support scientific investigations not only by our team, but especially by others (e.g., What is the optimal ratio of solution examples and problem-solving practice? How do computational thinking skills emerge? In what quanta are programming skills acquired? Can automated tutoring of programming be effective at scale in enhancing student learning?).

9 Broader Impacts

This proposal represents the first step toward building a community of practice that will broadly impact both research and education. We aspire to have direct impact on enhancing scientific productivity of at least 25 computer science education researchers whose modified tools will be used by over 100 instructors during the three year period of the grant, and many more in the years that follow. Their discoveries and technological innovations will in turn help tens of thousands of students in the strategically important field of computer science. Many of the innovations proposed can directly impact learning in any discipline. Educational software will more quickly be developed in the future, that more easily generates meaningful learner data, which in turn can be more easily analyzed.

10 Results from Prior NSF Support

Peter Brusilovsky is the PI for a 3-year project **Open Corpus Personalized Learning (IIS-1525186, 2015-2018, $499,758)**. The preliminary results are presented in [24, 23, 39]. **Intellectual Merit** This project is the first attempt to translate efficient closed-corpus adaptive hypermedia technologies to the open-corpus context. **Broader Impacts** Over the course of the project we will impact several hundred students in several courses by providing more advanced learning support with personalized guidance. At the moment of writing, six classes with an average of 50 students each were supported by the prototype system.

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 IUSE (DUE-1432008)** *Collaborative Research: Assessing and Expanding the Impact of OpenDSA, an Open Source, Interactive eTextbook for Data Structures and Algorithms.* PIs: C.A. Shaffer, J.V. Ernst, T.L. Naps (U Wisconsin-Oshkosh), S.H. Rodger (Duke U), $998,402, 01/01/201512/31/2017. **Intellectual Merit** These awards support the OpenDSA project, and active collaborations involving Virginia Tech, and Aalto University (Helsinki), Duke University, and U Wisconsin at both Madison and Oskhosh, among others. Related publications include [10, 15, 14, 13, 16, 26, 27]. **Broader Impacts** include dissemination of Algorithm Visualizations, interactive problems, and eTextbooks to thousands of CS students.

NSF TUES Type II Award (DUE-1245589), *CodePractice: Developing Coding Skills Using Social and Adaptive Drill-and-Practice Exercises*, $321,090 (07/01/13-06/30/17), PIs: Stephen Edwards and Manuel Prez-Quiones. **Intellectual merit** This project is developing and evaluating a new drill-and-practice system for code writing questions that allows students and instructors to write their own questions, includes rich data analysis of the performance of both questions and students, that provides social features for students to get help when they are stuck, and that provides adaptive suggestions for what to practice next. **Broader impacts** The system is currently in the evaluation phase, and is seeing use by over 1,000 students in multiple courses. The project has developed a community of 18 universities eager to serve as external adopters, and developed a collection of over one thousand code writing and multiple choice questions for use by students during practice and by instructors for homework assignments. The project has produced one Ph.D graduate and two MS theses [40, 41], and is expected to produce an additional thesis and multiple conference papers as the project evaluation completes.

Koedingers prior NSF support as PI includes **LearnSphere (CISE-ACI-1443068, 2015-2020, $5M)**. **Intellectual Merit** LearnSphere is creating data infrastructure building blocks to integrate the sharing and use of educational data and learning analytic methods. It has facilitated discoveries indicating six times bigger relationship to learning outcomes from online active doing with feedback than from reading online text or watching online videos [32, 33]. Other related publications include [2, 25, 31, 37, 57, 48, 49]. **Broader Impact** LearnSphere's DataShop stands as the world's largest open repository for educational technology data. DataShop contains nearly 1300 educational technology datasets and has supported over 125 data mining or secondary data analysis studies. Directly relevant to this project, LearnSphere's DataShop already contains 7 computer science education datasets including about 3.2 million data points contributed by 10,700 students.

References

[1] Norman Bier, Sean Lip, Ross Strader, Candace Thille, and Dawn Zimmaro. An approach to knowledge component/skill modeling in online courses, 2014.

[2] Sebastien Boyer, Ben U Gelman, Benjamin Schreck, and Kalyan Veeramachaneni. Data science foundry for moocs. In *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*, pages 1–10. IEEE, 2015.

[3] Peter Brusilovsky. Knowledgetree: A distributed architecture for adaptive e-learning. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 104–113. ACM, 2004.

[4] Peter Brusilovsky, Stephen H. Edwards, Amruth N. Kumar, Lauri Malmi, Luciana Benotti, Duane Buck, Petri Ihantola, Rikki Prince, Teemu Sirkiä, Sergey A. Sosnovsky, Jaime Urquiza-Fuentes, Arto Vihavainen, and Michael Wollowski. Increasing adoption of smart learning content for computer science education. In *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference, ITiCSE-WGR 2014, Uppsala, Sweden, June 23-25, 2014*, pages 31–57, 2014.

[5] Peter Brusilovsky and Sergey A. Sosnovsky. Individualized exercises for self-assessment of programming knowledge: An evaluation of quizpack. *ACM Journal of Educational Resources in Computing*, 5(3):6, 2005.

[6] Peter Brusilovsky and Michael V Yudelson. From webex to navex: Interactive access to annotated program examples. *Proceedings of the IEEE*, 96(6):990–999, 2008.

[7] Owen Conlan and Vincent P Wade. Evaluation of apels–an adaptive elearning service based on the multi-model, metadata-driven approach. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 291–295. Springer, 2004.

[8] IMS Global Learning Consortium. Caliper analytics background. `http://www.imsglobal.org/activity/caliperram`, 2017.

[9] IMS Global Learning Consortium. Learning tools interoperability. `https://www.imsglobal.org/activity/learning-tools-interoperability`, 2017.

[10] Matthew L Cooper, Clifford A Shaffer, Stephen H Edwards, and Sean P Ponce. Open source software and the algorithm visualization community. *Science of Computer Programming*, 88:82–91, 2014.

[11] S.H. Edwards, J. Snyder, M.A. Prez-Quioes, A. Allevato, D. Kim, and B. Tretola. Comparing effective and ineffective behaviors of student programmers. In *Proceedings of the fifth international workshop on Computing education research workshop (ICER'2009)*, pages 3–14, 2009.

[12] Stephen H Edwards and Manuel A Perez-Quinones. Web-cat: automatically grading programming assignments. *ACM SIGCSE Bulletin*, 40(3):328–328, 2008.

[13] E. Fouh, D.A. Breakiron, S. Hamouda, M.F. Farghally, and C.A. Shaffer. Exploring students learning behavior with an interactive etextbook in computer science courses. *Computers in Human Behavior*, pages 478–485, December 2014.

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

[15] Eric Fouh, Monika Akbar, and Clifford A Shaffer. The role of visualization in computer science education. *Computers in the Schools*, 29(1-2):95–117, 2012.

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

[17] Juha Helminen, Petri Ihantola, Ville Karavirta, and Satu Alaoutinen. How do students solve parsons programming problems? execution-based vs. line-based feedback. In *LATICE '13: Proceedings of the 2013 Learning and Teaching in Computing and Engineering*, page 5561, 2013.

[18] Roya Hosseini, Teemu Sirkiä, Julio Guerra, Peter Brusilovsky, and Lauri Malmi. Animated examples as practice content in a java programming course. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pages 540–545. ACM, 2016.

[19] Roya Hosseini, Arto Vihavainen, and Peter Brusilovsky. Exploring problem solving paths in a java programming course. In *Psychology of Programming Interest Group Annual Conference, PPIG 2014*, pages 65–76, 2014.

[20] I-H Hsiao, Sergey Sosnovsky, and Peter Brusilovsky. Guiding students to the right questions: adaptive navigation support in an e-learning system for java programming. *Journal of Computer Assisted Learning*, 26(4):270–283, 2010.

[21] I-Han Hsiao, Fedor Bakalov, Peter Brusilovsky, and Birgitta König-Ries. Progressor: social navigation support through open social student modeling. *New Review of Hypermedia and Multimedia*, 19(2):112–131, 2013.

[22] Yun Huang, Jose Gonzalez-Brenes, and Peter Brusilovsky. The fast toolkit for unsupervised learning of hmms with features. In *The Machine Learning Open Source Software workshop at the 32nd International Conference on Machine Learning*. University of Pittsburgh, 2015.

[23] Yun Huang, J Guerra, and Peter Brusilovsky. A data-driven framework of modeling skill combinations for deeper knowledge tracing. In *Proceedings of the 9th Intl. Conf. on Educational Data Mining (Accepted)*, 2016.

[24] Yun Huang, Michael Yudelson, Shuguang Han, Daqing He, and Peter Brusilovsky. A framework for dynamic knowledge modeling in textbook-based learning. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, pages 141–150. ACM, 2016.

[25] James Max Kanter and Kalyan Veeramachaneni. Deep feature synthesis: Towards automating data science endeavors. In *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*, pages 1–10. IEEE, 2015.

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

[27] Ville Karavirta and Clifford A Shaffer. Creating engaging online learning material with the jsav javascript algorithm visualization library. *IEEE Transactions on Learning Technologies*, 9(2):171–183, 2016.

[28] Judy Kay, Bob Kummerfeld, and Piers Lauder. Personis: a server for user models. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 203–212. Springer, 2002.

[29] Kenneth R Koedinger, Ryan SJd Baker, Kyle Cunningham, Alida Skogsholm, Brett Leber, and John Stamper. A data repository for the edm community: The pslc datashop. *Handbook of educational data mining*, 43, 2010.

[30] Kenneth R Koedinger, Emma Brunskill, Ryan SJd Baker, Elizabeth A McLaughlin, and John Stamper. New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine*, 34(3):27–41, 2013.

[31] Kenneth R Koedinger, Sidney D'Mello, Elizabeth A McLaughlin, Zachary A Pardos, and Carolyn P Rosé. Data mining and education. *Wiley Interdisciplinary Reviews: Cognitive Science*, 6(4):333–353, 2015.

[32] Kenneth R. Koedinger, Jihee Kim, Julianna Zhuxin Jia, Elizabeth A. McLaughlin, and Norman L. Bier. Learning is not a spectator sport: Doing is better than watching for learning from a MOOC. In *Proceedings of the Second ACM Conference on Learning @ Scale, L@S 2015, Vancouver, BC, Canada, March 14 - 18, 2015*, pages 111–120, 2015.

[33] Kenneth R Koedinger, Elizabeth A McLaughlin, Julianna Zhuxin Jia, and Norman L Bier. Is the doer effect a causal relationship?: how can we tell and why it's important. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 388–397. ACM, 2016.

[34] Kenneth R Koedinger, Elizabeth A McLaughlin, and John C Stamper. Automated student model improvement. *International Educational Data Mining Society*, 2012.

[35] Advanced Distributed Learning Lab. Lab experience api (xapi). `http://tincanapi.com/overview/`.

[36] Advanced Distributed Learning Lab. Total learning architecture. `https://www.adlnet.gov/tla/`.

[37] Nan Li, Noboru Matsuda, William W Cohen, and Kenneth R Koedinger. Integrating representation learning and skill learning in a human-like intelligent agent. *Artificial Intelligence*, 219:67–91, 2015.

[38] M Lovett, O Meyer, and C Thille. Open learning initiative: Testing the accelerated learning hypothesis in statistics. *Journal of Interactive Media in Education*, 2008.

[39] Rui Meng, Shuguang Han, Yun Huang, Daqing He, and Peter Brusilovsky. Knowledge-based content linking for online textbooks. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 18–25, 2016.

[40] P. Mohan. Student perceptions of various hint features while solving coding exercises. Master's thesis, Virginia Tech, 2015.

[41] K.P. Murali. Codeworkout: Design and implementstion of an online drill-and-practice system for introductory programming. Master's thesis, Virginia Tech, 2016.

[42] Thomas L Naps, James R Eagan, and Laura L Norton. Jhavéan environment to actively engage students in web-based algorithm visualizations. In *ACM SIGCSE Bulletin*, volume 32, pages 109–113. ACM, 2000.

[43] Andy Nguyen, Chris Piech, Jon Huang, and Leo Guibas. Codewebs: scalable homework search for massive open online programming courses. In *the 23rd International World Wide Web Conference (WWW 2014)*, pages 491–502, 2014.

[44] John F Pane, Beth Ann Griffin, Daniel F McCaffrey, and Rita Karam. Effectiveness of cognitive tutor algebra i at scale. *Educational Evaluation and Policy Analysis*, 36(2):127–144, 2014.

[45] Andrei Papancea, Jaime Spacco, and David Hovemeyer. An open platform for managing short programming exercises. In *Proceedings of ICER 2013 (International Computing Education Research)*, August 2013.

[46] Thomas W. Price, Neil C.C. Brown, Chris Piech, and Kelly Rivers. Sharing and using programming log data. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, page 729, March 2017.

[47] Steve Ritter, Michael Yudelson, Stephen E Fancsali, and Susan R Berman. How mastery learning works at scale. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pages 71–79. ACM, 2016.

[48] Kelly Rivers, Erik Harpstead, and Ken Koedinger. Learning curve analysis for programming: Which concepts do students struggle with? In *Proceedings of the 2016 ACM Conference on International Computing Education Research*, pages 143–151. ACM, 2016.

[49] Kelly Rivers and Kenneth R Koedinger. Data-driven hint generation in vast solution spaces: a self-improving python programming tutor. *International Journal of Artificial Intelligence in Education*, pages 1–28, 2015.

[50] C.A. Shaffer, M.L. Cooper, A.J.D. Alon, M. Akbar, M. Stewart, S. Ponce, and S.H. Edwards. Algorithm visualization: The state of the field. *ACM Transactions on Computing Education*, 10:1–22, August 2010.

[51] Frank M Shipman, Lillian Cassel, Edward Fox, Richard Furuta, Lois Delcambre, Peter Brusilovsky, B Stephen Carpenter II, Gregory Hislop, Stephen Edwards, and Daniel D Garcia. Ensemble: a distributed portal for the distributed community of computing education. In *International Conference on Theory and Practice of Digital Libraries*, pages 506–509. Springer, 2010.

[52] Juha Sorva, Ville Karavirta, and Lauri Malmi. A review of generic program visualization systems for introductory programming education. *ACM Transactions on Computing Education*, 13(4):15:1–15:64, 2013.

[53] J.G. Trafton and B.J. Reiser. The contributions of studying examples and solving problems to skill acquisition. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, pages 1017–1022, 1993.

[54] Mónica Trella, Cristina Carmona, and Ricardo Conejo. Medea: an open service-based learning platform for developing intelligent educational systems for the web. In *Workshop on Adaptive Systems for Web-Based Education: tools and reusability (AIED'05)*, pages 27–34, 2005.

[55] Kalyan Veeramachaneni, Zachary Pardos, O Una-May, Franck Dernoncourt, Colin Taylor, et al. Moocdb: Developing data standards and systems for mooc data science. 2013.

[56] Maaike Waalkens, Vincent Aleven, and Niels Taatgen. Does supporting multiple student strategies lead to greater learning and motivation? investigating a source of complexity in the architecture of intelligent tutoring systems. *Computers & Education*, 60(1):159–171, 2013.

[57] Xu Wang, Diyi Yang, Miaomiao Wen, Kenneth Koedinger, and Carolyn P Rosé. Investigating how student's cognitive behavior in mooc discussion forums affect learning gains. *International Educational Data Mining Society*, 2015.

[58] Michael Yudelson, Peter Brusilovsky, and Vladimir Zadorozhny. A user modeling server for contemporary adaptive hypermedia: An evaluation of the push approach to evidence propagation. In *International Conference on User Modeling*, pages 27–36. Springer, 2007.