Collaborative Research: An Infrastructure for Sustainable Innovation and Research in Computer Science Education

CISE Community Research Infrastructure (CCRI)
Due January 11, 2022

Innovative technology to support highly interactive smart learning content combined with recent advances in data-driven learning science is rapidly changing Computing Education Research (CER). The widespread adoption of new educational technology such as learning management systems (LMS) and interactive content are generating large volumes of learning data. New tools for analyzing big data leveraged by AI (e.g., deep learning for assessment) in turn improve both content and pedagogy, thus setting up a virtuous cycle fueling learning discoveries and leveraging innovation in AI: Online technologies → big data analysis → better online technologies.

To accelerate this promise, we propose a dedicated socio-technical research infrastructure for Computing Education Research built on our highly successful SPLICE project, called SPLICE-Portal. SPLICE-Portal is needed because too many computing education innovations are not evaluated or scaled across populations, and because sharing of innovative pedagogical interventions, data, analytic techniques, and resulting discoveries is too limited. SPLICE-Portal provides the potential to collect and share learning data at scale from many institutions on a collection of innovative interventions and analyzed by many researchers using many approaches. Following our past successes in socio-technical infrastructure creation and use, SPLICE continues to be as much about researcher community building as it is about the technical infrastructure. SPLICE-Portal will expand the number of Computing Education Researchers while facilitating novel research through sharing and use of research-based best practices, of innovative learning technologies, of high volume/quality data, of advanced AI and statistical methods optimized for CE goals, and of rigorous evaluation methods to demonstrate large, lasting, and replicable impacts on student achievement.

Prior NSF-funded work enabled us to form an active community of tool developers and data analysts. We now must broaden active participation to include educators as well. This will set up another virtuous cycle tying the research community (who are developing innovative pedagogical systems) to instructors who are using the online systems to generate the massive data required by the research community to drive the next cycle of innovation.

Keywords: Computing Education; Data Mining; Machine Learning; Interoperability

Intellectual Merit: SPLICE-Portal will facilitate scientific advances in Computing Education Research including: 1) understanding and breaking barriers to scale in instructor adoption of applications of research-based best practices and innovative learning technologies, 2) using high volume/quality learner data to unlock mysteries of human learning in the complex context of computing education and to produce innovations that optimize student learning effectiveness, efficiency, and engagement, and 3) developing new algorithms that produce better learning analytics or better automated learning support. These scientific advances cross disciplinary boundaries, come from technical and social scientists, and will be disseminated through the SPLICE-Portal infrastructure and community as well as usual publication.

Broader Impact: SPLICE-Portal will have direct and immediate impact on hundreds of researchers during the proposal phase and beyond. It will reduce barriers to educational innovation and support scientific discoveries in the many scientific communities that contribute to computing education research but where many researchers and educators currently isolated into separate silos. The discoveries and innovations enabled by SPLICE-Portal will in turn help tens of thousands of students in the strategically important field of computer science.

1 Motivation and Goals

Computing Education Research (CER) research is active and rapidly expanding. The ACM Special Interest Group on CS Education (SIGCSE) is one of the largest ACM SIGs. CER is also attracting the interest of CISE researches from a number of other domains such as AI, Machine Learning, and Data Science. This interest is stimulated by the increasing availability of large volumes of learning data collected in traditional and online learning contexts. We now have new ways to analyze such data including modern AI and machine learning techniques, and through the lens of new pedagogical theories. At the same time, researchers are developing new approaches to pedagogy using highly interactive "smart" learning content. This includes things like online programming exercises, proficiency exercises, and AI-driven intelligent tutors. This combination of new data, smart content, and better pedagogical techniques that could leverage these data promises to radically improve computing education. However, this promise is hobbled by the lack of adequate computing education research infrastructure. Without this infrastructure, researchers are currently limited to smaller-scale research, typically exploring one novel tool at a single institution or exploring a single dataset with their own analysis tools. Even the largest projects today explore at best a single tool or pedagogy across institutions, or work with multiple datasets produced in a single institution. While the community as a whole has a wealth of pedagogical interventions, datasets, and data analysis approaches, the lack of infrastructure makes it unreasonably difficult to collect and reuse this community wisdom and build upon the work of others.

The goal of this proposal is to develop SPLICE-Portal– an Infrastructure for Sustainable Innovation and Research in Computing Education to enhance and scale CER by leveraging the power of data-driven AI and ML. To do so, we need to overcome 3 challenges: data (there is not enough quantity and quality of data to develop, test and benchmark data-driven methods), analytics (developing and sharing data mining and AI methods for CER is highly siloed and disconnected) and evaluation (while "smart" learning tools are being developed by many researchers in the community, they are not easily deployed and replicated across to other institutions). To address these challenges, we will leverage our connections to the CER community to integrate existing projects into a large collection of resources including datasets, analytical approaches, and reusable smart learning content. We will build tools and user services that enable the community to reuse these resources and contribute to the collection.

The proposal leverages the results of a NSF-supported multi-year community building and planning project "Community-Building and Infrastructure Design for Data-Intensive Research in Computer Science Education". Through a series of 15 workshops we presented and discussed the ideas of a CER Infrastructure to over 1000 researchers and formed a diverse community of researchers interested in Standards, Protocols, and Learning Infrastructure for Computing Education (SPLICE, cssplice.org). We further engaged this community through working groups, collaborative projects, summer school courses, and online forums in discussing, conceptualizing, and prototyping various components of the proposed infrastructure.

SPLICE refined our vision of the infrastructure, helped us to understand the needs and priorities of key research communities involved in CER, and identified efficient community engagement approaches. We propose to address these needs and more effectively engage these communities. We will go the next step by effectively reaching computing educators to use the course material, which will in turn allow collection of massive amounts of data for use by the research community. In this way, we reach critical mass on the virtuous cycle: Online technologies $\rightarrow$ big data analysis $\rightarrow$ better online technologies.

We will build a research infrastructure for CER that facilitates a wide variety of researchers in exploring research questions of scientific and practical interest. The following is a sampling of such questions that our vision will help researchers address.

1. **How do we broaden participation of CS and Learning Science Researchers in CS Education Research?** Sub-questions include:
   (a) What are current barriers to participation, especially among underrepresented populations? How do factors like experience, motivation, and pedagogy affect participation?
   (b) How can community building and infrastructure tools and services reduce those barriers and increase participation?
2. **Can we better apply techniques for automatic assessment and feedback to more cognitively demanding, open-ended CS tasks, like programming and proof writing?**
3. **How does automated feedback affect the learning process? When do novel techniques for selected-response activities, such as mixed-up code problems and learner sourced selections, produce more effective or efficient student learning?**
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)? How are they best ordered and presented?**
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?**

Community engagement and outreach efforts are a main priority of our work, as we grow the infrastructure to include a broader range of researchers and integrate instructors into the process, as they are critical to data generation. To enhance the SPLICE community of practice built around our planning and design project and to better engage researchers from the fields of AI, Machine Learning, and data mining, we are joining forces with another multi-year community building effort – a sequence of workshops on Educational Data Mining in Computer Science Education (CSEDM) organized at the international conferences on Educational Data Mining and Learning Analytics and Knowledge. Continuing our outreach and community building efforts through a set of workshops, working groups, small grants, summer schools, data challenges, and other activities, we expect to increase our reach, engaging researchers from different communities as pilot users and co-developers of the proposed infrastructure. We allocate special efforts to engage researchers from smaller colleges and HBCU currently underrepresented in CER and form a broader and more inclusive community of practice. Community engagement and outreach will be not just the product of our project, but a core metric for developing and evaluating the infrastructure.

2 Targeted Research Communities

SPLICE-Portal will support two broad research communities we refer as Computing Education Research and Advanced Learning Science and Technology.

The Computing Education Research (CER) community focuses on discoveries about learning within computing disciplines that produce demonstrably effective and efficient pedagogical approaches and learning tools (e.g., using Peer Instruction halves failure rates [67]). The interests of this community are represented by ACM SIGCSE with its conferences (SIGCSE, ITiCSE, ICER). Through tutorials and workshops at these conferences, SPLICE will benefit veteran researchers in this community but also work to convert educational practitioners into doing computing education research. We hypothesize that more would CS educators would do research in conjunction with their courses if it were easier and SPLICE will make it so. **The Advanced Learning Science and Technology (ALST) research community** pursues discoveries that yield novel techniques and technologies that enhance learning. They might leverage AI, data, and educational theory to automatically support students (e.g., intelligent tutoring systems [88]) and further our understanding of learning processes (e.g., how student self-explanation enhances skill acquisition and conceptual development [96, 97]). This group includes computer scientists, engineers, cognitive scientists, psychologists, educators and social scientists. Their interests are represented by societies and corresponding conferences such as Artificial Intelligence in Education (AIED), Learning at Scale (LS), Educational Data Mining (EDM) and Learning Analytics and Knowledge (LAK). SPLICE outreach to these conferences will help convert more ALST researchers into focusing on Computing as a learning domain. SPLICE will engage new participants in both communities and empower all to answer more ambitious computing research questions.

A typical computing education research study involves evaluating the impacts on students of a novel educational technology (e.g., enhanced compiler error messages [6]) or of a novel pedagogical intervention (e.g., Pair Programming [34]). Many such interventions have demonstrating efficacy in enhancing student learning or engagement, including Parson's problems [99, 23], worked examples [38], automated programming hints [60, 76], feedback [30, 59], and metacognitive support [68]. However, this research is limited by two key challenges. First, interventions developed and evaluated by one researcher rarely make it into other classrooms, especially across institution – computing education research interventions have 2.38% replication rate [35]. This limits not only the generalizability and reliability of the research, but also its impact on learners. This is particularly problematic because CS learners vary widely (e.g., K-12 to university students, end users, and informal learners), and what works in one context may not work in another. Without data from larger, more diverse populations, it is also difficult to study interventions' impact on underrepresented students (e.g., women, racial minorities, students with disabilities), which is a core focus of CER[7]. A second challenge is that much computing education research is done by instructors, many of whom are new to education research, and may lack experience designing educational studies, deploying tools, collecting rich data from their classes, and analyzing it. Despite the fact that these tasks share many commonalities across studies, there are few tools to support new researchers in conducting experiments, leading many to instead write "experience reports," focusing on more anecdotal data. Both of these challenges can be addressed by a well-designed infrastructure that enables reuse of innovative smart learning content (SLC) across classrooms, supports the use of SLC collections for classroom studies, and facilitates data collection and evaluation.

Typical studies in the ALST research community use data to answer research questions, such as how to develop algorithms that automatically generate help for students (e.g., data-driven programming hints and feedback [72, 76]), how to model and predict student outcomes (e.g., predicting student performance on future problems [64]), and how to further learning theories using analytics (e.g., discovering student misconceptions by clustering and visualizing their code [83]). A core challenge for this research community is finding high-quality, well-labeled, and sufficiently large datasets with which to investigate research questions. For example, modern deep learning approaches such as Code2Vec can automatically label programs with high accuracy, but are typically trained on datasets with 2+ million programs [3] and fare worse on small, educational datasets [83] with less than 1000 students. Additionally, there are few benchmarks or agreed-upon standards for common modeling tasks in the CER domain (e.g., knowledge tracing), making it difficult to establish or improve upon the state of the art. It can be especially difficult for newcomers, such as computing education researchers interested in crossing over into ALST research, to find the right tools and datasets to get started. Analyzing program code to develop smart learning content often requires domain-specific tools (e.g., abstract syntax tree and compiler error analysis), but existing tools for sharing such analytics workflows in the ALST community (e.g., LearnSphere.org's web-based workflow authoring tool) lack support for these CS-specific tasks. These challenges can be addressed by better infrastructure for collecting, aggregating, and analyzing high-quality (i.e., labeled, annotated, documented) CS data and wide re-use of analytics approaches.

A final challenge is collaboration and networking *across* the two communities. Many computing education researchers already use AI and machine learning to develop learning tools and predictive models (e.g., [28, 63]), but may lack the deep expertise of the ALST community on applying such techniques to education. Similarly, many ALST researchers study computing education (e.g., [61, 51]), but may lack the pedagogical expertise and disciplinary theory of computing education researchers. Our prior work has begun to bridge this gap by holding 14 workshops at the intersection of CER and ALST, including SPLICE and CSEDM workshops. This nascent community now needs research infrastructure to help connect CER and ALST researchers to share and benefit from each other's expertise and data.

3 Contributing Research and Practice

For the past decade course materials have increasingly moved online, often organized within LMS augmented by interactive services such as discussion forums, chats, and wikis. This online revolution has been further accelerated by the COVID-19 pandemic. At the same time, increased online interactions have produced new data and driven new opportunities for the social sciences, including the learning sciences. The convergence of online learning opportunities and learning science advances has produced a revolution in research-based, technologically-advanced online resources. We refer to these as Smart Learning Content (SLC). SLC is not simply text and lectures online, but interactive learning experiences that engage students in challenging tasks. Some SLC applies AI technologies to offer intelligent analysis of student problem solutions, provide feedback adapted to the learner's solutions, and select new tasks adapted to the learner's needs and capabilities. Examples of SLC include program visualization and simulation tools [85], algorithm animations [81], code analysis problems [12], Parson's problems [23], automatic assessment services for programming exercises [22], intelligent tutoring systems [79], and interactive worked examples [38]. Studies repeatedly show that use of SLC results in significant improvements in student learning [49, 50, 56, 65, 38]. Beyond the intrinsic benefits of active learning, SLC collects rich learner interaction log data. This data enables researchers to create scientific models of learning with much finer-grain precision than is possible with traditional learning content [46].

Our team's past efforts have supported the computing education research community to construct online courses by linking together multiple SLC components. To make such linking more routine, we encouraged the adoption of the Learning Tools Interoperability (LTI) standard [17]. One example was a collaboration with partners at Aalto University, Finland to augment an open source Acos SLC server [84] with LTI connectivity. To further promote the use of LTI-enabled SLC and solicit community feedback we created a live catalog of SLC for computing education [37].

This progress is facilitating online course advancement, but more is needed to leverage data-driven research opportunities. Current systems typically do a poor job of standardized reporting of student data to facilitate communication across SLC components, to support data sharing and interpretation, and to facilitate sharing of analytic methods matched to those data standards.

To address these challenges, SPLICE will build user community consensus and implement tools that generate and analyze standardized data. We leverage emerging standards for learning interactions including Experience API [52] developed by ADL and Caliper [16] developed by IMS. User community support is needed as these standards have not achieved community acceptance and more tool development is needed to not only enhance these standards, but to provide real value in their use. Such enhancements are critical to enriching a science of learning relevant to computing education. Caliper, for example, does not yet support the collection of complex data generated by SLC for computing, such as snapshots of student programs submitted to assessment. SPLICE will drive such standards enhancements by creating easy-to-use web-based analytic workflows and by supporting researcher to use to make discoveries from complex learner interaction data.

Our team is well positioned to provide user support and tools to drive a new generation of computing education research. We build on past efforts at creating infrastructure support for learning data and analytics. Co-PI Koedinger led CMU's LearnSphere project (supported by NSF CISE-ACI) to provide analytic workflows that integrate across learning data silos. LearnSphere integrates across different data types and existing repositories. These include CMU's DataShop for intelligent tutoring clickstream data [46], MIT's MOOCdb [89] and Stanford's DataStage both for massively open online course data, and CMU's DiscourseDB for student writing and discourse data. To support easier development and reuse of data analytic methods, LearnSphere pioneered a web-based workflow authoring tool. In it analytic routines encapsulate specific data processing approaches as workflow components that include data import, processing, and data mining. LearnSphere supports users in re-configuring its continually growing set of components in data-flow sequences that make complex data import, transformation, analysis and report easy to replicate, reapply and adapt. LearnSphere has facilitated discoveries such as multiple replications of the so-called "doer effect": Across multiple online course settings, student learning outcomes are typically about six times more highly associated with how much they do in the course (e.g., working with interactive SLC) than with how much online reading or lecture video watching they do [49, 50]. More broadly, this integration is fueling a wide variety of research on learning science and technology [79, 9, 43, 48, 53, 95].

These efforts are a strong basis to build from, however, a computing education research community is not sufficiently served by these efforts. We need a social-technical infrastructure that provides existing and new computer education researchers with a one-stop shop where they get both support (the social part) and research tools and services (the technical part) for achieving open experimentation and data analytics in computing education. We suggest a set of development, research, and community-building steps that are necessary to implement our vision of SPLICE *Standards, Protocols, and Learning Infrastructure for Computing Education.* Our work can significantly advance research progress in the fields of CER as well as related fields of Machine Learning, Datamining, and Learning Science.

4 Infrastructure Description

The target SPLICE infrastructure will support all aspects of the CER cycle: developing a rich variety of novel educational tools and environments, planning and organizing research studies with these tools, data collection, data analysis, and data-driven experiments. Through discussions, experiments, and pilot efforts performed in collaboration with many partners, SPLICE developed a consistent vision of this infrastructure based on the analysis of stakeholders' needs, existing solutions, and best practices. We assembled a diverse community of active collaborators and prospective users to work with us in implementing and piloting the components of the infrastructure, and we will incorporate their feedback throughout development. A representative set of letters from this community is attached to this proposal. We will extend this community through a series of engagement and outreach efforts. Thus the outcome of our project will be both the target infrastructure and a diverse community of users who have already started to work with it.

4.1 Fundamental infrastructure The target infrastructure is centered around two types of resources that were found most critical for the success of CER during our design stage: SLC and related services, and datasets. SLC enables a wide range of user studies and classroom experiments that are necessary to implement and evaluate new pedagogies and technologies in courses. Datasets are critical to analyse the results of these experiments, complement them with data-driven studies, and create new knowledge about learning. The infrastructure therefore has two main hubs — the content hub and the data hub. Each supports a structured collection of resources of its target type and a set of documents, tools, and services that enables users of these resources for their CER needs. Both types of resources are user-contributable, reusable, and extendable. Dedicated tools and services to support these functionalities on several levels of resource aggregation. The hubs are directly connected: all types of resources and services in the SLC hub produce a flow of learner data when interacting with learners. These data are assembled in the learning record stores and passed to the data hub for archiving, analysis, and reuse. In turn, data-driven SLC artifacts can make use of data in the data hub to adapt learning content to users (e.g., by using trained student models [98]). Access to both hubs is provided through central host several services that support the infrastructure as a whole with access to information repositories (documentation, case studies, best practices) and social services (finding like-minded users, recommending collaborators).

4.2 Tools, resources, and data sets **The content hub** is centered around a repository of reusable SLC items and services. The re-usability of these components is supported by their adherence to LTI and Caliper standards ensuring the connection of SLC to learning management systems and allows centralized collection of learner data in learning record stores (LRS). To adapt Caliper for the specifics of computing education data, we will develop Caliper *profiles* for key types of computing SLC. While we build on existing standards, our new content hub infrastructure is needed to overcome two fundamental challenges with hosting SLC for CER.

First, while considerable research on building learning content repositories has been done in the past [74], SLC is different from traditional learning content: an SLC item is essentially an interactive service that communicates with the learner working with a learning activity (a problem, a worked example, an animation, etc), collects data, and provides feedback [11]. To support it, each SLC item has to be hosted on a Web server that is typically separate from the Learning Management System (LMS). To explore the feasibility of building an SLC repository during project planning stage, we collaborated with SPLICE community members to develop a live catalog of computing SLC presented through Canvas [37]. Our experience is that with adequate support documentation, most tool developers can easily convert their stand-alone systems into providers of standard-compliant SLC [84, 57, 36]. Workshop discussions indicated that less experienced SLC developers will need more assistance. To support these prospective contributors, we will develop additional tools: a model open-source SLC server built on the basis of the ACOS server [84] and authoring tools to contribute most popular types of content (programming problems and examples) to be hosted on several "standard" SLC servers maintained by our project members [21, 39].

Second, as discovered in our prior work, is the challenge of adding SLC to a LMS such as Canvas. While LTI is designed to integrate a single external item (e.g., a programming exercise) into a an LMS provider, many SLC consist of *collections* of items (e.g., 50 problems), making integration difficult. To support easy re-use of SLC, we will also develop model *content integrators* built on our existing open source tools OpenDSA [26] and MasteryGrids [55]. A content integrator is an intermediary between the LMS and SLC, supported by LTI and Caliper standards. It provides access to a structured set of SLC items that could be connected to the target delivery platform as a single service, organized for example as an eTextbook. While SLC could be re-used without integrators, item by item, these tools extend SLC reusability from the level of items to the level of collections, such as a collection or items pre-designed by an experienced researcher or educator to support a specific course or a specific textbook. Integrators can also provide additional functionalities not supported by a given LMS, such as progress dashboards. New content servers and tools can be contributed by the community, extending the functionality of the infrastructure. Contributing standard-compliant SLC and tools will be supported by a range of documentation from specifications to tutorials and services provided by project staff.

The data hub will be centered around a repository of reusable learning datasets hosted on LearnSphere's DataShop [46], currently the world's largest open repository for educational technology data, with over 1300 educational technology datasets. The work with LearnSphere will allow us to leverage its Workflow mechanism, which enables the users to reuse not just datasets, but also data analytics. Workflows offer an extendable collection of data import, processing, and data mining components, which could be assembled into structured pipelines. Just like content integrators, workflows increase the level of reusability from items to item sets.

The reusability of datasets and workflows is supported by data representation standards. Since DataShop has not been developed to support CER data, we will augment it with a set of standards to represent typical CER data. This work began under a SPLICE working group that developed and integrated with DataShop the ProgSnap2 [71] standard for representing program snapshot data. Over ten datasets in this format were collected, with about one million code snapshots, and they were used to enable cross-institution CER research on code predictors of student success [71], as well as two CER analytics competitions [70, 69]. Similarly, existing analytic methods (e.g., learning curve analytics) require much special purpose processing to be used with CER data, for example, to transform program solution submissions into incremental graded solution steps (cf. [77]). We will augment LearnSphere with components than will support CER-specific analytics approaches and will work with CER-specific data standards.

4.3 User services To facilitate users and contributors, we will build a set of services around the two hubs of resources and tools. During the first year of the project, we will develop several search services to help infrastructure users locate relevant SLC items, datasets, and analytic components as well as contributed item aggregates such as structured SLC collections and complete workflow. Search services will support a variety of metadata for filtering results such as domain (i.e., Python programming), a topic within the domain, type of data produced by content or processed by an analytic component, etc. We will provide services that support contributors adding new SLC items, datasets, and methods. Another set of services will support users in building and contributing aggregates of individual items — i.e., designing a collection of SLC items to support learning a specific topic or whole course or assembling a complete data workflow from analysis components. In the next two years we will develop more advanced services to better address the needs of our diverse community. It will include item recommendation services that will proactively suggest SLC, datasets, and aggregates matching to users personal interests and needs. Collaborator recommendation services will suggest users with similar or complementary interests. We will also extend contribution services with support services for automated extraction of metadata, for example processing SLC code to extract *knowledge components* and processing text fragments to extract keyphrases. To implement these advanced services we will leverage infrastructures for item and content recommendations developed by PI Brusilovsky in previous NSF-supported projects [14, 86, 73, 15]. The extendable nature of the infrastructure will also support complex support services as well as more advanced SLC types, tools, and analytics approaches extending the value of the infrastructure could be added on the next stage of development with the support of the extension funding.

4.4 Community Engagement Our prior work with SPLICE has engaged a broad segment of the CER community in discussing, designing, and prototyping SLC. We have broadly convinced them of the importance of using appropriate integration protocols for smart content such as the LTI protocol. We have had success promoting specialized data standards such as PEML and ProgSnap2 [71] through a working group model. Our community of tool builders is already strong, and the tools they have implemented reach many thousands of students each semester.

Our weaker links to date have been in agreeing on standards for data collection and analysis. On the data collection side, this was due to the slow progress on the Caliper standard, which is somewhat out of the control of the CER community. We believe that this standard has now reached a stage that can benefit from our focused efforts on data channels between smart content components. This will leave propagating analysis tools and public data sets within the CER community as our biggest concern going forward. These mechanisms exist for the very small CER community, where people are slowly buying in to these efforts. However, the bigger issue is getting broad uptake among the front-line CS educators at all levels. This was not a major focus of SPLICE, but must be a metric of success for SPLICE-Portal since their students' use of these systems powers the virtuous cycle: online technologies → big data analysis → better online technologies.

We use several approaches for outreach and community engagement. "First contact" for reaching target users will be through dedicated annual workshops organized at conferences that bring together researches in our two key contributing communities: CER and ALST researchers. We engage the CER community by hosting workshops at the annual ACM SIGCSE meeting. The SIGCSE annual conference historically has brought together the Computing Education *researcher* community with a broad cross-section of the CS *instructor* community. Thus, this venue allows us to reach what is admittedly an unusually focused and energized subset of the instructors who are interested in CS education practice and research. To engage the ALST research community, we will continue our series of CSEDM workshops, led by PI Price and colleagues. These workshops (five and counting) rotate among popular conferences for the community (EDM, AIED, LAK, LS;

see Section 2). Additionally, we will expand the CSEDM Data Challenge [70, 69] to an annual event. The Data Challenge brings together ALST researchers to compete in a competition to solve a pressing problem in the field (e.g., predicting student dropout), using *multiple* public CS datasets (in future hosted on SPLICE-Portal in a common format). Competitors will submit their solution as a reusable analytics component (also hosted on SPLICE-Portal via Tigris), which can then be used by other researchers or instructors (e.g., as an early-warning dashboard).

The next level of engagement and support is through websites and other informational repositories. These are geared toward supplying immediate support to the instructor who has developed a casual interest in adopting tools, and needs to be helped over the hump of actually getting started. Based on our SPLICE work, we have some experience with designing such support sites (cssplice.org, which was geared toward providing support for the CER community). Many of the co-PIs on this proposal also have significant experience with supporting a user community for our own tools like OpenDSA, Code Workout, Mastery Grids, and DataShop.

We will continue our support for working groups for specific topics, such as the successful ProgSnap and PEML working groups. But we will broaden the definition of these to go beyond our traditional technical groups organized around a given standard. We will set up working groups focused on support for the instructor community that wants to adopt our infrastructure. Such working groups can focus explicitly on outreach and support for this community.

To engage most experienced and authoritative members of our target communities, we will also create a formal Advisory Board. Our original SPLICE project had many unofficial advisors coming out of our many workshops, and as leaders of the various working groups that SPLICE relied on for creating standards like PEML and ProgSnap. As we move to the phase of broader community engagement, including significant instructor engagement, we will identify 6-10 candidates for an Advisory Board from outside the PI institutions. This will include senior researchers with experience in tool building, CER experiment design, data analysis, and machine learning, and learning sciences. We will also include instructors who are active users of the tools. Besides online meetings as appropriate, the Advisory Board is expected and compensated (see Budget) to meet at the project meetings in conjunction with the planned workshops.

4.5 Community Outreach Co-PI Barnes serves as Community Outreach Director to ensure that a broad and diverse community of users is engaged in ongoing outreach. Planned outreach will expand the number and diversity of people attending each annual (1) SPLICE workshop, (2) SIGCSE & Tapia workshops, and (3) LearnLab Summer School. Annual SPLICE workshops are co-located with leading ALST conferences (EDM, AIED, LAK, L@S), where we help EDM researchers to engage in CER. At the annual SIGCSE and Tapia workshops, we will engage CS faculty and computing education researchers with EDM/analytics. These workshops will be conducted by project PIs and consultants. PI Barnes who will ensure that content relating to broadening participation and equity are integrated (i.e. anonymization, demographic disaggregation).

We leverage the long history of the LearnLab Summer Schools to reach out, engage, and educate both existing and new participants into Computing Education Research. Every year about 80 researchers, postdocs, PhD students, and educators have attended the week-long summer school for the last 20 years. Besides general lectures on advanced learning science and technology, attendees participate in a focused research track and execute a research project. The research tracks include educational data mining, intelligent tutor authoring, online course development, and computational modeling of learning. While participation has been across educational domains, for this project our goal is to have 10-20 computing education research participants each summer. These events not only help the community learn how to use the tools and services provided by SPLICE, but they also provide a great networking opportunity to aid broadening participation of young researchers. Each week-long project team involves a cross-disciplinary pair (e.g., one is a CS educator who brings data and goals and another is a machine learning PhD student wanting to demonstrate a newly developed algorithm) that is supervised by two mentors, also typically interdisciplinary. A great many of these projects lead to research publications.

The budget includes funding to engage 15-25 new participants each year in these planned workshops and summer school, with a focus on engaging diverse participants who might otherwise not be able to participate. PI Barnes will recruit graduate students and computer science faculty for these spots by leveraging the STARS Computing Corps (STARS), an NSF-funded Broadening Participation in Computing (BPC) Alliance that builds a community of diverse CS faculty and students doing BPC work. STARS includes 50 colleges and universities, many of which are minority-serving institutions, community colleges, and historically black colleges and universities. The majority of faculty and students in STARS are from groups that are traditionally underrepresented in computing, including women, Black/African Americans, Hispanic/Latinx, and people with disabilities. Each year, PI Barnes will coordinate two webinars, in spring (April) and fall (October), about SPLICE opportunities and a third workshop at the Tapia Celebration of Diversity in Computing to recruit diverse participants. Barnes will also leverage the STARS communication channels of social media, listserv, and newsletters, and website to share SPLICE opportunities with STARS. STARS BPC Scholars, a group of 5-6 faculty per year, will also be invited to apply for SPLICE opportunities, as the learning opportunities for SPLICE will enable them to enrich their ability to conduct research about the impact of educational interventions on broadening participation. Selection of travel scholarship recipients will be based on need and criteria including potential to (1) diversify the community, especially to include women, Black, and Hispanic participants, (2) contribute new data to the project, and (3) eventually be able to contribute to CER.

5 Project Outcomes and Evaluation

The overarching goal for this project is to advance the progress of data-driven and AI-enhanced CER through an infrastructure that empowers our target communities of CER and ALST researchers (see Section 2). Specifically we expect the infrastructure will enable a novel research agenda that aligns with existing CER priorities to deliver transformative scientific advancements (see Section 1 for example RQs within that agenda). Three pillars of that research agenda are:

1) Broadening Participation in Computing: Research on understanding and increasing the participation of underrepresented students in CS (women, students of color, first-generation students, students disabilities) has been a cornerstone of the CER community's efforts for decades [80, 66, 32, 20, 8]. CER interventions can dramatically improve retention and learning outcomes for students *generally* [58, 67]. However, researchers rarely have enough data from underrepresented groups to meaningfully detect how interventions affect them, though existing work suggests underrepresented student groups may experience different outcomes from interventions [87, 54]. SPLICE-Portal will enable researchers to pool data from across classrooms more easily, and more easily collect and analyze demographic data, to detect these effects and develop interventions that support the needs of specific underrepresented populations, informed by data in addition to theory (e.g., [19]). Researchers from this area including Nell O'Rourke [29] and Barbara Ericson [31, 1] have provided letters of collaboration detailing how they will use our infrastructure, and Co-PI Barnes has a proven track-record of research in this area [20, 5].

2) Large-scale Data-driven Models for Rich CS Data: Many CER researchers use data-driven algorithms to automate and adapt support for students (e.g., hints, feedback, examples), and to model student behavior to predict and describe their learning. In ALST research, the most effective of these algorithms are data-hungry, such as deep learning models (e.g., SAKT, a winner of the Riiid student modeling challenge [64]), which require thousands of examples to learn from. Adapting these models to CER has been met with limited success [83, 82], both because CER datasets often lack the necessary scale, and because existing algorithms may be domain-general, and fail to leverage the rich, structured, and time-series data available from CS datasets (e.g., program snapshots, compiler errors, interaction logs). To develop next generation, CS-specific models and algorithms, researchers need to a repository of high-quality CS datasets with comparable attributes, capturing this fine-grained detail. Our SPLICE-Portal will do just that, leading to novel algorithms, e.g. for detecting what knowledge components are being practiced in code [98, 77], or which equivalence classes exist within code [78]. These can be captured as reusable analytics components to enable the next round of advancements. Top researchers in this area (e.g. Sharon Hsiao [41, 40] and Bita Akram [2]) have provided letters of collaboration, and PIs Brusilovsky [13, 10], Koedinger [47], and Price [72] and coPI Barnes [42] all have extensive work in this area.

3) Generalizing What Works for CS Learning: Despite 50 years of CER, there are remarkably few interventions with robust evidence supporting their efficacy, and less than 2.5% of CER research constitutes a replication [35]. The next milestone in CER research will involve adapting interventions that have been successful in one setting (e.g. a lab study, or a single classroom) to work with diverse learners across many institutions and contexts. Data-driven interventions have the additional challenge of transfer learning: getting a model trained in once classroom (e.g. to predict when a student needs help) to work on a new population of learners, who may act differently – a grand challenge for Educational Data Mining [4]. To do so, researchers will need our proposed SPLICE-Portal infrastructure, to not only connect instructors and researchers, but also to facilitate the large-scale deployment, data collection and experimentation necessary to *consistently* scale research across institutions. Over 10 collaborators who develop and evaluate CER interventions have written letters of collaboration, indicating their intent to use SPLICE-Portal to replicate their research across contexts, and scale its impact.

Evaluation: To evaluate the infrastructure's success in promoting the above research agenda, we will use the following metrics (projected numbers in parentheses are based on our prior work): 1) the number of new SLC items contributed to the content hub (30); 2) the number of additional instructors using such tools in classrooms (40); 3) the number of new datasets contributed to the data hub (100); 4) the number of unique researchers downloading these datasets (300); 5) the number of analytics components contributed (20) and unique researchers using them (200); and 6) the number of papers published about a dataset, SLC, or analytic workflow hosted on SPLICE-Portal (50). These metrics can be measured automatically by SPLICE-Portal's internal analytics and logging, and tagged papers researchers contribute with SLC/workflows/datasets.

Our engagement and outreach efforts (see Sections 4 and 4) will enable a new generation of researchers (e.g. PhD students, instructors, researchers from other disciplines) to answer future research questions at the intersection of CSE and ALST, facilitated by SPLICE-Portal and access to study populations, data, and analytics workflows. We will measure the success of these efforts by tracking: the number of participants – especially new participants – at each of our 3 annual workshops (30+ each), the LearnLab summer school (10-20), and the CSEDM Data Challenge (30+). We will also track how many of these participants go on to use the infrastructure in their research or teaching (as measured above), or apply for our small grants to support this transition.

Ultimate evidence of our impact will be more qualitative: the number and quality of "success cases" that can be achieved. Key elements to a success case include infrastructure-enabled collaborative research, barriers overcome, and novel research insights. Here is a hypothetical example from collaborator Yetunde Folajimi, who recently took a tenure-track position at a small college, with research interests in AI, eLearning, the empowerment of youth and women, and more recently using novel technology-based learning approaches for computing education [24]. She now wants to investigate how online practice with SLC could help help graduates of urban schools to succeed in challenging introductory programming courses, and which type of SLC is most beneficial for these students. If successful, our infrastructure would enable Folajimi to **(1) find collaborators** who have developed relevant SLC (e.g., a practice environment like CodeWorkout [21], hints to support students [72], and Parsons problems to reduce cognitive load [23]) by attending a workshop (e.g. at SIGCSE, where she may already attend), or by joining a cohort of our STARS BPC Scholars with other women of color. SPLICE-Portal would help her **(2) overcome barriers** such as finding and installing SLC appropriate for her class through the content hub. It helps her collect detailed study data securely using the data hub and its Learning Record Store. Afterwards, she can use LearnSphere's existing workflows to analyze study data, measuring the effect of her intervention, disaggregated by demographic group. Documentation and use cases provided on SPLICE-Portal help her on each step of this work, and consultants assist her with uploading pre- and post-test data required by the workflow. Finally, working with her collaborators, Folajimi publishes her **(3) novel research insights** in ACM SIGCSE's Technical Symposium, demonstrating the value of online practice and identifying effective types of SLC for her target under-served population.

6 Qualifications of the Project Team

The project brings together an interdisciplinary team, comprised of the University of Pittsburgh, Virginia Tech, Carnegie Mellon University, and North Carolina State University faculty. The PIs represent an ideal mix of expertise, including extensive experience creating, evaluating, and integrating digital education software systems, leveraging learning data, working with the CER communities, and building communities of practice, as detailed below. All investigators have strong records of interdisciplinary research and have a proven track record of collaboration.

Peter Brusilovsky is a Professor of Information Science and Intelligent Systems at the University of Pittsburgh, where he directs the Personalized Adaptive Web Systems (PAWS) lab. He has been working in the field of adaptive educational systems and user modeling for more than 30 years. With support from NSF and DoD ADL Lab, PAWS developed several broadly used personalized learning systems for Java, SQL, and Python programming. Brusilovsky has extensive experience with learning infrastructures and standards. He developed the KnowledgeTree architecture [10] and collaborated with DoD ADL Lab on components of the Total Learning Architecture [55].

Kenneth Koedinger is a professor of Human Computer Interaction at CMU. He has contributed new educational techniques and technologies and has produced basic cognitive science research results on the nature of STEM thinking and learning. Dr. Koedinger has contributed to and led many large-scale research infrastructure efforts including LearnLab.org, a Science of Learning

	2022 Fall	2023 Spring	2023 Summer	2023 Fall	2024 Spring	2024 Summer	2024 Fall	2025 Spring	2025 Summer
Development stages	First iteration design and development	First iteration release and pilot testing	First iteration evaluation and result analysis	2nd iteration design and development	2nd iteration release and pilot testing	2nd iteration evaluation and result analysis	3td iteration design and development	3rd iteration release and pilot testing	3rd iteration evaluation and result analysis
Development milestones	Project website	LTI support Live catalog	ACOS server	LRS & Caliper support	SPLICE-Portal	Model integrators	Resource recommenders	Collaborator recommender	
Community engagement milestones	Call for small grant proposals	SIGCSE '23 meeting WG formation	CSEDM '23 meeting WG formation	2nd call for small grant proposals	SIGCSE '24 meeting WB reports	CSEDM '24 meeting WG reports	3rd call for small grant proposals	SIGCSE '25 meeting WG reports	CSEDM '25 meeting WG reports
Outreach events	TAPIA '22 Fall webinar	SIGCSE '23 Spring webinar	CSEDM '23 Summer School	TAPIA'23 Fall webinar	SIGCSE '24 Spring webinar	CSEDM '24 Summer School	TAPIA'24 Fall webinar	SIGCSE '25 Spring webinar	CSEDM '25 Summer School

Figure 1: The timeline of development, community engagement, and outreach milestones.

Center funded by NSF for 10 years and about $50M. He also led the development of DataShop and LearnSphere, which together are the world's largest infrastructures for sharing educational technology data and analytic routines.

Clifford A. Shaffer is Professor of Computer Science and Associate Department Head for Graduate Studies at Virginia Tech. He leads the OpenDSA project that provides infrastructure and materials for teaching a variety of CS courses related to Data Structures and Algorithms, formal languages, and concepts for compilers and translators. The current proposal will benefit from OpenDSA solutions to several challenges, including how to integrate educational software components, how to collect and process student analytics data, and how to build a community of practice around developing and using the software.

Thomas Price is an Assistant Professor of Computer Science at NC State University, where his research focuses on developing and evaluating novel, data-driven support for students learning to program. He is a lead organizer of the CSEDM Workshop (2018-2021), and founder of the CSEDM Data Challenge (2019-2021), which both bring together researchers in the CER and ALST communities. He is an organizer of the ProgSnap2 standard for logging programming process data.

Tiffany Barnes is a Distinguished Professor of Computer Science at NC State University and Distinguished Member of the Association of Computing Machinery (ACM). Barnes has researched artificial intelligence in education, educational data mining, computer science education, and broadening participation for more than 20 years. She is a pioneer in data-driven intelligence for learning environments, including logic and programming. Dr. Barnes has established and grown many communities, as Founding Co-Director of the NSF-funded STARS Computing Corps Broadening Participation in Computing Alliance, and founding member and current President of the International Educational Data Mining Society, and a representative on the International Alliance to Advance Learning in the Digital Era.

7 Work Organization and Project Schedule

Our project will be organized around two hubs: the content hub and the data hub. We will release an initial version of the infrastructure in the first year of the project using the SLC and data collections developed under SPLICE, and will be incrementally adding the components of the infrastructure (Section 4) over the duration of the project. Implementation efforts will be tightly integrated with our community engagement efforts. The extendable, open, and standard-based architecture of the infrastructure and its core standards are designed to support community contribution. Following our successful community building experience during SPLICE, we will engage active members of our research communities to contribute new types of SLC, content-support tools, datasets, reusable analytics, workflows, and services specific to questions of their interest. The distributed component-based nature of the infrastructure along with commitment of project members and collaborators to maintain various components of the infrastructure such as the DataShop and SLC servers will ensure sustainability of the infrastructure. The use of LTI and Caliper standards alsi turns LMS and other sustainable services maintained by participating universities into infrastructure components.

Development efforts will be informed by continuous pilot use and evaluation efforts (see Section 5). We will recruit pilot users of the infrastructure through the community engagement workshops focused on the SIGCSE and ALST communities and other means as as specified in Section 4. The results of the pilots will be discussed with our key contributors from both communities at the working meetings, included in the schedule of key community workshops. The workshops will also be used to solicit feedback from key contributors and the Advisory Board.

Pilot results and group discussions will serve as an input for the next design cycle to refine existing components and design new components. We expect to run three iterations of the design-pilot cycle, with each Fall focused on design, Spring and early Summer focused on community engagement and pilot use, and late Summer focused on reflecting on lessons learned from the previous stages. A final summary meeting to examine lessons learned during the entire project will be held in Summer 2025. The schedule of major development, evaluation, and community engagement activities is shown in Figure 1.

The design and development of the data hub components will be lead by Koedinger, the architect of DataShop, who has extensive experience in data analytics. The design and development of the content hub will be lead by Shaffer, one of the leading CER tool developers. Price and Brusilovsky will focus on the connections between the data and the content side of the infrastructure. They will work on the central SPLICE-Portal while also supporting design and development of content and data components correspondingly. All PIs will participate in community engagement and outreach efforts according to their community involvement and past engagement experience. Shaffer will focus on organizing SIGCSE workshops and connection with SIGCSE community, Price will focus on organization of CSEDM workshops and connections with ALST community. Koedinger will lead the organization of Summer Schools which will be hosted annually at CMU. Brusilovsky will lead the the engagement of collaborators through small grants and subcontracts. Community outreach efforts will be coordinated by Barnes, the Community Outreach Director, who will also organize several annual events. The overall project coordination will be provided bythe lead PI Brusilovsky. The PIs will also participate in the CCRI Virtual Organization (CCRI-VO) and will select representatives for CCRI community PI meetings. Information about developed resources, tools and other infrastructure components, as well as community outreach meetings, will be regularly shared with both the contributing project communities and with CCRI-VO.

8 Intellectual Merit

SPLICE will facilitate scientific advances in Computing Education Research including: 1) understanding and breaking barriers to scale in instructor adoption of applications of research-based best practices and innovative learning technologies, 2) using high volume/quality learner data to unlock mysteries of human learning in the complex context of computing education and to produce innovations that optimize student learning effectiveness, efficiency, and engagement, and 3) developing new algorithms that produce better learning analytics or better automated learning support. These scientific advances cross disciplinary boundaries, come from technical and social scientists, and will be disseminated through the SPLICE infrastructure and community as well as usual publication.

9 Broader Impacts

SPLICE will have direct and immediate impact on hundreds of researchers during the proposal phase and beyond. It will reduce barriers to educational innovation and support scientific discoveries in the many scientific communities that contribute to computing education research but where many researchers are currently isolated into separate silos. The discoveries/innovations enabled by SPLICE will in turn help tens of thousands of students in the strategically important field of CS.

10 Results from Prior NSF Support

Brusilovsky is PI for *Collaborative Research: CSEdPad: Investigating and Scaffolding Students' Mental Models during Computer Programming Tasks to Improve Learning, Engagement, and Retention* (Collaborating PIs: V. Rus (U. Memphis) (IIS-1822752, 2018-2022, $250,519). The preliminary results are presented in [38, 75]. **Intellectual Merit** The project is one of the first attempts to systematically study and model student comprehension of worked code examples. **Broader Impacts** Over the course of the project we will impact several hundred students in introductory programming classes by providing advanced learning support for example comprehension.

Shaffer has been PI for a series of NSF grants that supported creation and development of the OpenDSA eTextbook system (DUE-1139861, IIS-1258571, 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, 01/01/2015–12/31/2017. **Intellectual Merit** The project contributes to the OpenDSA eTextbook and its assessment with partners including Virginia Tech, Aalto University, Duke University, and U Wisconsin, among others. Publications include [18, 27, 26, 25, 33, 44, 45]. **Broader Impacts** include dissemination of algorithm visualizations, interactive problems, and eTextbooks to tens of thousands of CS students.

Koedinger's prior NSF support includes **LearnSphere (CISE-ACI-1443068, 2015-2020, $5M)**. **Intellectual Merit** LearnSphere provides data infrastructure building blocks to integrate the sharing and use of educational data and learning analytic methods. It has facilitated discoveries such as the six times bigger relationship to learning outcomes from online active doing with feedback than from reading online text or watching online videos [49, 50]. Other publications include [9, 43, 48, 53, 95, 77, 79]. **Broader Impact** LearnSphere's DataShop stands as the world's largest open repository for educational technology data. DataShop contains over 3000 educational technology datasets and has supported over 250 data mining or secondary data analysis studies. Directly relevant to this project, LearnSphere's DataShop already contains over 50 computer science education datasets including over 20 million data points contributed by some 75,000 students.

PI Price and **Co-PI Barnes** are PI and Co-PI for NSF Award #1917885: Data-driven Support for Creative, Open-ended Programming (08/14/2019 - 08/13/2022, $749,999). *Intellectual Merit*: This project develops novel, data-driven help features (e.g., hints and subgoals) to support students working on creative programming tasks (e.g., making apps and games), where there is no instructor solution. *Broader Impacts*: The developed technologies will augment programming environments already used in hundreds of AP Computer Science Principles classrooms that focus on open-ended, creative projects. *Publications*: Publications include results on example-based feedback [93, 91], automated assessment of open-ended projects [92, 94, 90] and planning open-ended projects [62].

References

[1] Rick Adrion, Renee Fall, Barbara Ericson, and Mark Guzdial. Broadening access to computing education state by state. *Communications of the ACM*, 59(2):32–34, 2016.

[2] Bita Akram, Wookhee Min, Eric Wiebe, Bradford Mott, Kristy Elizabeth Boyer, and James Lester. Improving stealth assessment in game-based learning with lstm-based analytics. In *International Conference on Educational Data Mining*, 2018.

[3] Uri Alon, Meital Zilberstein, Omer Levy, and Eran Yahav. code2vec: Learning distributed representations of code. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–29, 2019.

[4] Ryan S Baker, Peter Brusilovsky, Dragan Gasevic, Neil T Heffernan, Mykola Pechenizkiy, and Alyssa Friend Wise. Grand challenges for edm and related research areas. In *EDM*, page 15, 2015.

[5] Tiffany Barnes and George K Thiruvathukal. The need for research in broadening participation. *Communications of the ACM*, 59(3):33–34, 2016.

[6] Brett A Becker. An effective approach to enhancing compiler error messages. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pages 126–131, 2016.

[7] Sylvia Beyer. Why are women underrepresented in computer science? gender differences in stereotypes, self-efficacy, values, and interests and predictors of future cs course-taking and grades. *Computer Science Education*, 24(2-3):153–192, 2014.

[8] Jennifer M Blaney. Broadening participation in computing: The role of upward transfer. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, pages 254–260, 2020.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Hung Chau, Jordan Barria-Pineda, and Peter Brusilovsky. Course-adaptive content recommender for course authoring. In Viktoria Pammer-Schindler, Mar Pérez-Sanagustín, Hendrik Drachsler, Raymond Elferink, and Maren Scheffel, editors, *13th European Conference on Technology Enhanced Learning, EC-TEL 2018*, Lecture Notes in Computer Science, pages 437–451. Springer, 2018.

[15] Hung Chau, Igor Labutov, Khushboo Thaker, Daqing He, and Peter Brusilovsky. Automatic concept extraction for domain and student modeling in adaptive textbooks. *International Journal of Artificial Intelligence in Education*, 31(4):820–846, 2021.

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

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

[18] 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.

[19] Kimberle Crenshaw. Mapping the margins: Intersectionality, identity politics, and violence against women of color. *Stanford Law Review*, 43(6):1241–1299, 1991.

[20] Teresa Dahlberg, Tiffany Barnes, Kim Buch, and Audrey Rorrer. The stars alliance: Viable strategies for broadening participation in computing. *ACM Transactions on Computing Education (TOCE)*, 11(3):1–25, 2011.

[21] Stephen H. Edwards and Krishnan Panamalai Murali. Codeworkout: Short programming exercises with built-in data collection. In *2017 Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE'17*, pages 188–193. ACM Press, 2017.

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

[23] Barbara J. Ericson, James D. Foley, and Jochen Rick. Evaluating the Efficiency and Effectiveness of Adaptive Parsons Problems. In *Proceedings of the International Computing Education Research Conference*, pages 60–68, 2018.

[24] Yetunde Folajimi, Britton Horn, Jackie Barnes, Amy Hoover, Gillian Smith, and Casper Harteveld. A cross-cultural evaluation of a computer science teaching game. In *Games, Learning, and Society Conference*, Lecture Notes in Computer Science. ETC Press, 2016.

[25] 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.

[26] 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.

[27] 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.

[28] Ge Gao, Yang Shi, and Thomas W. Price. Early Performance Prediction using Interpretable Patterns in Programming Process Data. In *Proceedings of the ACM Technical Symposium on Computer Science Education*, pages 1–7, 2021.

[29] Jamie Gorson and Eleanor O'Rourke. Cs1 student assessments of themselves relative to others: The role of self-critical bias and gender. In *Proceedings of the 15th International Conference of the Learning Sciences-ICLS 2021.* International Society of the Learning Sciences, 2021.

[30] Luke Gusukuma, Austin Cory Bart, Dennis Kafura, and Jeremy Ernst. Misconception-Driven Feedback: Results from an Experimental Study. *Proceedings of the 2018 ACM Conference on International Computing Education Research - ICER '18*, pages 160–168, 2018.

[31] Mark Guzdial and Barbara Ericson. Georgia computes! an alliance to broaden participation across the state of georgia. *ACM Inroads*, 3(4):86–89, 2012.

[32] Mark Guzdial, Barbara J Ericson, Tom McKlin, and Shelly Engelman. A statewide survey on computing education pathways and influences: factors in broadening participation in computing. In *Proceedings of the ninth annual international conference on International computing education research*, pages 143–150, 2012.

[33] 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.

[34] Brian Hanks, Sue Fitzgerald, Renée McCauley, Laurie Murphy, and Carol Zander. Pair programming in education: A literature review. *Computer Science Education*, 21(2):135–173, 2011.

[35] Qiang Hao, David H Smith IV, Naitra Iriumi, Michail Tsikerdekis, and Andrew J Ko. A systematic investigation of replications in computing education research. *ACM Transactions on Computing Education (TOCE)*, 19(4):1–18, 2019.

[36] Ryan Hardt and Kamil Akhuseyinoglu. Database query analyzer integration. In *Sixth SPLICE Workshop "Building an Infrastructure for Computer Science Education Research and Practice at Scale" at ACM Learning at Scale 2020*, 2020.

[37] Alexander Hicks, Kamil Akhuseyinoglu, Clifford Shaffer, and Peter Brusilovsky. Live catalog of smart learning objects for computer science education. In *Sixth SPLICE Workshop "Building an Infrastructure for Computer Science Education Research and Practice at Scale" at ACM Learning at Scale 2020*, 2020.

[38] Roya Hosseini, Kamil Akhuseyinoglu, Peter Brusilovsky, Lauri Malmi, Kerttu Pollari-Malmi, Christian Schunn, and Teemu Sirkiä. Improving engagement in program construction examples for learning python programming. *International Journal of Artificial Intelligence in Education*, 30(2):299–336, 2020.

[39] Roya Hosseini, Kamil Akhuseyinoglu, Andrew Petersen, Christian D. Schunn, and Peter Brusilovsky. Pcex: Interactive program construction examples for learning programming. In *Proceedings of the 18th Koli Calling International Conference on Computing Education Research*, pages 5:1–5:9. ACM, 2018.

[40] 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.

[41] I-Han Hsiao, Peter Brusilovsky, and Sergey Sosnovsky. Web-based parameterized questions for object-oriented programming. In *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, pages 3728–3735. Association for the Advancement of Computing in Education (AACE), 2008.

[42] Wei Jin, Tiffany Barnes, John Stamper, Michael John Eagle, Matthew W Johnson, and Lorrie Lehmann. Program representation for automatic hint generation for a data-driven novice programming tutor. In *International conference on intelligent tutoring systems*, pages 304–309. Springer, 2012.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] Kenneth R. Koedinger and Albert Corbett. Cognitive Tutors: Technology Bringing Learning Sciences to the Classroom. In *The Cambridge handbook of: The learning sciences.*, pages 61–77. Cambridge University Press, 2006.

[48] 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.

[49] 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.

[50] 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.

[51] Anastasia Kovalkov, Benjamin Paassen, Avi Segal, Kobi Gal, and Niels Pinkwart. Modeling creativity in visual programming: From theory to practice. *International Educational Data Mining Society*, 2021.

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

[53] 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.

[54] Alex Lishinski, Aman Yadav, Jon Good, and Richard Enbody. Introductory Programming: Gender Differences and Interactive Effects of Students' Motivation, Goals and Self-Efficacy on Performance. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, pages 211–220, 2016.

[55] Tomasz Loboda, Julio Guerra, Roya Hosseini, and Peter Brusilovsky. Mastery grids: An open source social educational progress visualization. In Sara de Freitas, Christoph Rensing, Pedro J. Muñoz Merino, and Tobias Ley, editors, *9th European Conference on Technology Enhanced Learning (EC-TEL 2014)*, volume 8719 of *Lecture Notes in Computer Science*, pages 235–248, 2014.

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

[57] Hamza Manzoor, Kamil Akhuseyinoglu, Jackson Wonderly, Peter Brusilovsky, and Clifford A. Shaffer. Crossing the borders: Re-use of smart learning objects in advanced content access systems. *Future Internet*, 11(7):160, 2019.

[58] Lauren E Margulieux, Briana B Morrison, and Adrienne Decker. Reducing withdrawal and failure rates in introductory programming with subgoal labeled worked examples. *International Journal of STEM Education*, 7:1–16, 2020.

[59] Samiha Marwan, Ge Gao, Susan Fisk, Thomas W. Price, and Tiffany Barnes. Adaptive Immediate Feedback Can Improve Novice Programming Engagement and Intention to Persist in Computer Science. In *Proceedings of the International Computing Education Research Conference*, pages 194–203, 2020.

[60] Samiha Marwan, Joseph Jay Williams, and Thomas Price. An Evaluation of the Impact of Automated Programming Hints on Performance and Learning. In *Proceedings of the International Computing Education Research Conference*, pages 61–70, 2019.

[61] Samiha Marwan, Yang Shi, Ian Menezes, Min Chi, Tiffany Barnes, Thomas W Price, and North Carolina. Just a Few Expert Constraints Can Help : Humanizing Data-Driven Subgoal Detection for Novice Programming. In *Proceedings of the International Conference on Educational Data Mining*, pages 1–13, 2021.

[62] Alexandra Milliken, Veronica Catete, Amy Isvik, Tiffany Barnes, Wengran Wang, Thomas W. Price, Chris Martens, Neeloy Gomes, and Sarah Martin. PlanIT! A new integrated tool to help novices design for open-ended projects. In *Proceedings of the ACM Technical Symposium on Computer Science Education*, pages 1–7, 2021.

[63] Aya Nabil, Mohammed Seyam, and Ahmed Abou-Elfetouh. *Deep Neural Networks for Predicting Students' Performance*, page 1380. Association for Computing Machinery, New York, NY, USA, 2021.

[64] Shalini Pandey and George Karypis. A self-attentive model for knowledge tracing. *arXiv preprint arXiv:1907.06837*, 2019.

[65] 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.

[66] Joan Peckham, Lisa L Harlow, David A Stuart, Barbara Silver, Helen Mederer, and Peter D Stephenson. Broadening participation in computing: issues and challenges. *ACM SIGCSE Bulletin*, 39(3):9–13, 2007.

[67] Leo Porter, Cynthia Bailey Lee, and Beth Simon. Halving fail rates using peer instruction: a study of four computer science courses. In *Proceeding of the 44th ACM technical symposium on Computer science education*, pages 177–182, 2013.

[68] James Prather, Raymond Pettit, Brett A. Becker, Paul Denny, Dastyni Loksa, Alani Peters, Zachary Albrecht, and Krista Masci. First Things First. In *Proceedings of the ACM Technical Symposium on Computer Science Education*, pages 531–537, 2019.

[69] Thomas Price, Yang Shi, Bita Akram, Peter Brusilovsky, Kenneth R. Koedinger, and Juho Leinonen. CSEDM Data Challenge, 2021.

[70] Thomas W. Price. CSEDM Data Challenge, 2019.

[71] Thomas W Price, Ge Gao, David Hovemeyer, Kelly Rivers, Austin Cory Bart, Ayaan M Kazerouni, Stephen H. Edwards, Luke Gusukuma, Brett A Becker, Andrew Petersen, and David Babcock. ProgSnap2: A Flexible Format for Programming Process Data. In *Proceedings of the International Conference on Innovation and Technology in Computer Science Education*, pages 1–7, 2020.

[72] Thomas W. Price, Rui Zhi, and Tiffany Barnes. Evaluation of a Data-driven Feedback Algorithm for Open-ended Programming. In *Proceedings of the International Conference on Educational Data Mining*, pages 192–197, 2017.

[73] Behnam Rahdari, Peter Brusilovsky, Dmitriy Babichenko, Eliza Beth Littleton, Ravi Patel, Jaime Fawsett, and Zara Blum. Grapevine: A profile-based exploratory search and recommendation system for finding research advisors. In *83rd Annual Meeting of the Association for Information Science & Technology*, 2020.

[74] Pathmeswaran Raju and Vian Ahmed. Enabling technologies for developing next-generation learning object repository for construction. *Automation in Construction*, 22:247–257, 2012. Planning Future Cities-Selected papers from the 2010 eCAADe Conference.

[75] Zak Risha, Jordan Barria-Pineda, Kamil Akhuseyinoglu, and Peter Brusilovsky. Stepwise help and scaffolding for java code tracing problems with an interactive trace table. In *21st Koli Calling International Conference on Computing Education Research*. ACM, November 2021.

[76] Kelly Rivers. *Automated Data-Driven Hint Generation for Learning Programming*. Phd, Carnegie Mellon University, 2017.

[77] 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.

[78] Kelly Rivers and Kenneth R. Koedinger. A Canonicalizing Model for Building Programming Tutors. In *Proceedings of the International Conference on Intelligent Tutoring Systems*, pages 591–593, 2012.

[79] 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.

[80] Allison Scott, Alexis Martin, Frieda McAlear, and Sonia Koshy. Broadening participation in computing: Examining experiences of girls of color. In *Proceedings of the 2017 ACM conference on innovation and technology in computer science education*, pages 252–256, 2017.

[81] 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.

[82] Yang Shi, Ye Mao, Tiffany Barnes, Min Chi, and Thomas W Price. More with less: Exploring how to use deep learning effectively through semi-supervised learning for automatic bug detection in student code. *International Educational Data Mining Society*, 2021.

[83] Yang Shi, Krupal Shah, Wengran Wang, Samiha Marwan, Poorvaja Penmetsa, and Thomas W. Price. Toward Semi-Automatic Misconception Discovery Using Code Embeddings. In *Proceedings of the 11th International Learning Analytics and Knowledge Conference (LAK21)*, pages 1–6. Association for Computing Machinery, 2021.

[84] Teemu Sirkiä and Lassi Haaranen. Improving online learning activity interoperability with Acos server. *Software: Practice and Experience*, 47(11):1657–1676, 2017.

[85] 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.

[86] Chun-Hua Tsai and Peter Brusilovsky. Exploring social recommendations with visual diversity-promoting interfaces. *ACM Transactions on Interactive Intelligent Systems*, 10(1):Article 5, 2020.

[87] Alexandria Katarina Vail, Kristy Elizabeth Boyer, Eric N Wiebe, and James C Lester. The Mars and Venus Effect: The Influence of User Gender on the Effectiveness of Adaptive Task Support. In *Proceedings of the Conference on User Modeling, Adaptation and Personalization*, pages 265–276, 2015.

[88] Kurt VanLehn. The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education*, 16(3):227–265, 2006.

[89] Kalyan Veeramachaneni, Franck Dernoncourt, Colin Taylor, Zachary Pardos, and Una-May O'Reilly. Moocdb: Developing data standards for mooc data science. *AIED 2013 Workshops Proceedings Volume*, 1009:17, 01 2013.

[90] Wengran Wang, Gordon Fraser, Chris Martens, and Thomas Price. Execution-Trace-Based Feature Engineering To Enable Formative Feedback on Visual , Interactive Programs. In *The 5th CSEDM Workshop on Educational Data Mining in Computer Science Education*, pages 1–10, 2021.

[91] Wengran Wang, Archit Kwatra, James Skripchuk, Neeloy Gomes, Alexandra Milliken, Chris Martens, Tiffany Barnes, and Thomas W. Price. Novices' Learning Barriers When Using Code Examples in Open-Ended Programming. In *Proceedings of the International Conference on Innovation and Technology in Computer Science Education*, pages 1–7, 2021.

[92] Wengran Wang, Yudong Rao, Yang Shi, Alexandra Milliken, Chris Martens, Tiffany Barnes, and Thomas W. Price. Comparing Feature Engineering Approaches to Predict Complex Programming Behaviors. In *Proceedings of the 4th Workshop on Educational Data Mining in Computer Science Education (CSEDM) at EDM'20*, pages 1–7, 2020.

[93] Wengran Wang, Yudong Rao, Rui Zhi, Samiha Marwan, Ge Gao, and Thomas W. Price. Step Tutor: Supporting Students through Step-by-Step Example-Based Feedback. In *Proceedings of the International Conference on Innovation and Technology in Computer Science Education*, pages 391–397, 2020.

[94] Wengran Wang, Chenhao Zhang, Andreas Stahlbauer, Gordon Fraser, and Thomas Price. SnapCheck: Automated Testing for Snap Programs. *Proceedings of the International Conference on Innovation and Technology in Computer Science Education*, pages 1–7, 2021.

[95] 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.

[96] Joseph J. Williams and Tania Lombrozo. The role of explanation in discovery and generalization: Evidence from category learning. *Cognitive Science*, 34(5):776–806, 2010.

[97] Joseph Jay Williams, Tania Lombrozo, and Bob Rehder. The hazards of explanation: Overgeneralization in the face of exceptions. *Journal of Experimental Psychology: General*, 142(4):1006–1014, 2013.

[98] Michael Yudelson, Roya Hosseini, Arto Vihavainen, and Peter Brusilovsky. Investigating Automated Student Modeling in a Java MOOC. In *Proceedings of the International Conference on Educational Data Mining*, pages 261–264, 2014.

[99] Rui Zhi, Min Chi, Tiffany Barnes, and Thomas W Price. Evaluating the Effectiveness of Parsons Problems for Block-based Programming. In *Proceedings of the International Computing Education Research Conference*, pages 51–59, 2019.