Mapping Materials to Curriculum Standards for Design, Alignment, Audit, and Search

Alec Goncharow
Computer Science, UNC Charlotte
Charlotte, North Carolina

Matthew Mcquaigue
Computer Science, UNC Charlotte
Charlotte, North Carolina

Erik Saule
Computer Science, UNC Charlotte
Charlotte, North Carolina

Kalpathi Subramanian
Computer Science, UNC Charlotte
Charlotte, North Carolina

Jamie Payton
Computer & Information Sciences,
Temple University
Philadelphia, Pennsylvania

Paula Goolkasian
Psychological Science, UNC Charlotte
Charlotte, North Carolina

ABSTRACT

Computing proficiency is an increasingly vital component of the modern workforce, and computer science programs are faced with the challenges of engaging and retaining students to meet the growing need in that sector. However, administrators and instructors often find themselves either reinventing the wheel or relying too heavily on intuition, despite the availability of national curriculum standards. To address these issues, we present **CS Materials**, an open-source resource targeted at computing educators for designing and analyzing courses for coverage of recommended guidelines, and alignment between the various components within a course, between sections of the same course, or course sequences within a program. The system works by facilitating mapping educational materials to national curriculum standards.

A side effect of the system is that it centralizes the design of the courses and the materials used therein. The curriculum guidelines act as a lingua franca that allows examination of and comparison between materials and courses. More relevant to instructors, the system enables a more precise search for materials that match particular topics and learning outcomes, and dissemination of high quality materials and course designs.

This paper discusses the system, and analyzes the costs and benefits of its features and usage. While adding courses and materials requires some overhead, having a centralized repository of courses and materials with a shared structure and vocabulary serves students, instructors, and administrators, by promoting a data-driven approach to rigor and alignment with national standards.

CCS CONCEPTS

• **Human-centered computing** → **Information visualization**; • **Applied computing** → **Education**; • **Social and professional topics** → **Model curricula**; *Accreditation*; **Computer science education**; *Student assessment*.

KEYWORDS

curriculum guidelines; learning materials; alignment

ACM Reference Format:
Alec Goncharow, Matthew Mcquaigue, Erik Saule, Kalpathi Subramanian, Jamie Payton, and Paula Goolkasian. 2021. Mapping Materials to Curriculum Standards for Design, Alignment, Audit, and Search. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), March 13–20, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3408877.3432388

SIGCSE '21, March 13–20, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8062-1/21/03…$15.00
https://doi.org/10.1145/3408877.3432388

1 INTRODUCTION

Computer science skills are vital for promoting creative problem solving [1] and fostering participation in the emerging tech-focused global economy. In the U.S., growth in information technology and computer science jobs is projected at 13%, about twice the rate of all other professions [25]. In light of these needs, it is particularly timely and important to prepare undergraduate computer science students with high-quality learning experiences that promote retention in the major. Towards that goal, instructors are called upon to implement promising strategies in their courses, develop learning materials to increase student engagement, connect to potential career paths, and implement inclusive pedagogical techniques.

However, it is difficult for individual instructors to meet all these expectations. While instructors have discipline-specific expertise, very few have formal training in pedagogy. Many are not aware of evidence-based strategies for course design, nor do they pay sufficient attention to alignment between various course components (e.g., lectures, or assessments) within their own course, or their course's impact on downstream courses. High enrollment numbers make it harder for instructors to simultaneously pay attention to course structure and content. Additionally, administrators are often interested (or required) to assess their degree programs for internal review and external accreditation. This requires instructors to identify how their course content align with national CS and engineering education standards, which, although beneficial, adds another layer of complexity.

We believe that much of these problems can be alleviated by having courses (and materials within) explicitly mapped to national

standards such as the ACM 2013 CS curriculum guidelines [16]. We present in this paper **CS Materials** (https://cs-materials.herokuapp.com/), an open-source resource for designing and analyzing courses that rely on that principle. By mapping all learning materials in a class to national curriculum standards, **CS Materials** enables us to improve course design in a data-driven approach by checking coverage of topics of the class against recommended guidelines, by aligning different components of a course, by comparing sections of the same course, or across courses in a program.

With many materials mapped, the system can become a public resource of courses and materials. This will be of interest to educational researchers in the future. More immediately, it helps instructors publicize their well designed materials to impact the broader community and helps other instructors adopt materials that align well with the topics and learning outcomes of their courses.

In this paper, we are interested in studying the costs and benefits of mapping course materials against accepted curriculum guidelines, and will consider the following research questions:

RQ1: How much effort is required for an instructor to map course materials to a classification that embodies curriculum standards?
RQ2: Will mapping materials to curriculum guidelines lead to better course designs, in terms of topic coverage and alignment between the different elements within a course or between similar courses?
RQ3: Will searching for specific materials using classification tags that meet specific learning objecives be more effective when learning materials are mapped to curriculum guidelines?

2 RELATED WORK

We explore two primary areas of related work: (1) efforts to connect curriculum guidelines to learning materials and (2) repositories that support sharing and adoption of computer science educational materials. We summarize standards that serve as the "lingua franca" of CS education and address the limitations of existing repositories.

2.1 Curriculum Guidelines, Standards and Mapping Approaches

We are interested in understanding curriculum standards in computing, and their relationships to course design and content. The 2013 ACM computing curriculum guidelines [16], which are the current *de facto* standard for undergraduate CS degree programs, specify a 'redefined body of knowledge, a result of rethinking the essentials necessary for a Computer Science curriculum.' The guidelines provide exemplars of actual courses and programs that can be adopted by CS departments. The guidelines divide the body of knowledge into *knowledge areas*, with each containing a set of *topics* and *learning outcomes*. Learning outcomes are classified at 3 levels: *familiarity, usage,* and *assessment*. Sub-areas of computing have developed their own standards, such as parallel computing [21], cyber security [4], data science [3] and high school CS curriculum [9, 10] which could be used instead of or in addition to the ACM 2013 guidelines. Our system allows for incorporating multiple standards, and in the longer-term, we would define mappings to updates of a particular standard, to keep the content current.

In recent years, we are seeing increased offerings of online courses and programs. Quality Matters (QM) [23] is a rigorous standard that was developed to guide instructional design of online/blended/hybrid courses in a way to promotes student engagement, satisfaction, and learning outcomes. A key aspect of a QM certified course is *alignment*, that span the course objectives, materials and activities, all of which are student centered. Many of these principles form the bedrock of student motivation [18], and most also apply to face-to-face courses.

Tungare *et al.* [27] created a repository of computing course syllabi indexed by the ACM Computing Curriula 2001 [17], by using a web crawler to collect and analyze syllabi. A similar attempt involved analyzing learning outcomes of CS1 courses spanning 207 institutions and 30 countries [8]. Dragon and Mitchell [12] proposed a bottom up process of building concept maps relating CS concepts and learning resources, and extending them to program level objectives to assess student skills; our approach also has similar goals towards improved course design.

2.2 CS Learning Material Sources

Nifty Assignments are a set of over 120 assignments, collected since 1999 through an annual competition and presented at the ACM SIGCSE conference [22]. The selection is based on engagement, adoptability and scalability, and targeted at early courses (CS0, CS1, CS2). Nifty assignments now include metadata and many use real-world data, game interactions, and/or visualization. Many CS sub-communities followed the approach of Nifty, for instance, *Model AI* [2], *Groovy Graphics* [14], and *Peachy Parallel* [13]. *EngageCSEdu* is an NCWIT sponsored repository that provides introductory CS course materials, primarily engaging assignments targeted at CS0, CS1, and CS2 [19, 20]. The assignments are categorized by engagement practices to improve student inclusiveness, confidence and to broaden participation in computing. The repository has about 237 assignments with a competition for excellence [24], and submitted assignments are subject to peer review. *CS Unplugged* curates a set of activities that bring (without computers) a physicality to teaching computer science concepts [15]. There are 22 activities involving papers, blocks, crayons, and strings targeted at K-12 education, and publicized through science fairs and museums, and is localized in many parts of the world [28]. The CORGIS data repository [5, 6] is a large collection of tools, datasets and resources that can be used by educators as part of their programming assignments. The datasets range across a large number of disciplines and have been used in introductory courses, such as Computational Thinking by Bart *et al.* [7]. Using real-world datasets can be highly engaging in introductory courses. The work by Burlinson *et al.* as part of the BRIDGES project illustrates integrating real-world applications and data in data structures courses [11, 26].

2.3 Takeaways

On the one hand, the community has recognized the importance of well accepted curriculum guidelines, that designing courses is hard, and that alignments of topics and outcomes within modules and within a whole course leads to student satisfaction and improved outcomes. Yet, aligning courses and following curriculum guidelines is currently a difficult task. On the other hand, the community has crafted a number of assignment repositories to help improve the quality of courses; yet these repositories are hard to use because

identifying which assignment will integrate well in an instructor's course requires *essentially reading all of them.*

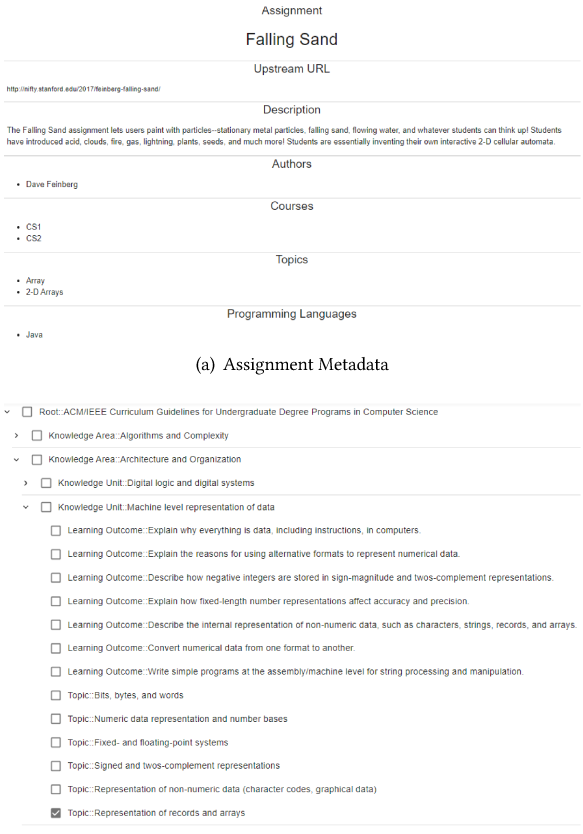
CS Materials solves both problems at once. It provides a tool to *design and analyze courses, while following curriculum standards.* The tool ensures that the modules within a course are well aligned. In the case of multiple courses, it helps ensuring different sections of a course cover the same topics and learning outcomes; or, in the case of a sequence of courses that the topic coverage of each class is clearly specified. As a side effect of aligning courses, the materials collected during the activity of course alignment become easily searchable, since they are naturally mapped to well-accepted ontologies in CS education. The materials become a resource that can be reused to achieve better course designs. The courses that are entered in the system also become case studies of how particular topics are taught across the world for education experts to study.

3 THE CS MATERIALS SYSTEM

Central to this project is the classification system for aligning learning materials to curriculum guidelines. Instructors can provide their materials for inclusion in the CS Materials system, classifying them according to multiple curriculum standards. This flexible approach to classification makes it possible for instructors to search for materials according to multiple criteria, as well as for instructors and administrators to consider mappings of courses and materials across multiple dimensions and guidelines.

3.1 Adding Materials

A common task for instructors is adding a new learning material (e.g., project, lecture slides) to the system and classifying it according to a given set of topics, learning objectives, or curriculum guidelines. A form guides the user to share basic information (e.g., title, authors, description) used to build the item's metadata. An example is shown in Figure 1(a) for mapping an assignment on recursion to the ACM classification. The leaf nodes in the tree list in Figure 1(b) represent topics in the curriculum guidelines/standards, and can be selected to indicate that the particular topic is covered by the assignment; a similar hierarchy is available for selecting learning outcomes.



(a) Assignment Metadata

(b) Editing Classification

Figure 1: Adding and Classifying Assignments

3.2 Coverage Views

Users can visualize the coverage of a course in terms of topics or objectives by viewing a *hit-tree*. The hit-tree is a tree representation, with items associated with the course highlighted in a subset of the ACM classification tree. Example courses and coverage of their associated items can be seen in Figure 2: nodes in orange are the selected topics or learning outcomes, those in blue represent nodes that are on the path to a selected node and the remaining gray nodes provide context to the selections. The nodes are sized according to the number of times the entry is selected throughout the course.

3.3 Alignment Views

Users can also explore material alignment, i.e., how well does the material align with the module or course objectives, between different materials within a course, such as assessments vs. lecture slides, and even materials between two sections of a course. For this task, we need ways to compare different sets of materials. We created an alignment tree for two sets of materials S_1 and S_2 which is a subtree of the ACM classification, where a node is in the alignment tree if a mapped item appears in any of the materials of S_1 or S_2. The nodes of the tree are scaled to the number of materials that are mapped against that item. The nodes are colored on a diverging color scale, where the color represents the difference between the relative number of items in S_1 that are mapped against that item and the same item in S_2. (Example in Figure 3.)

3.4 Search Views

A key application of the system is searching for learning materials. Educators will use the system to search for new materials if they are beginning to design a course, or look for similar or equivalent materials; for instance, they might want to find a different or more engaging project assignment, or a better slide set to explain a particular concept. Students can also find uses with the system to search for practice quizzes, assignments or exercises that reinforce course material, or for practice questions to better prepare for exams.

The first type of search we support can use standard keyword search, but *given that we index by topics and learning outcomes, we can perform a much richer search.* This is a significant advantage over existing systems, in the sense that we can narrow our search by exploiting classification terms. We use *similarity-based search*,

combining topics and learning outcomes at different levels of the classification hierarchy to find matching materials. Indeed, two materials m_1 and m_2 can be represented as a vector in a space where each dimension is an entry in the ACM/IEEE CS curriculum guidelines. Multiple distance metrics can be used to measure similarity, for instance, the classical cosine similarity is defined as: $\cos(m_1, m_2) = \frac{\sum_i m_1(i)m_2(i)}{||m_1||\ ||m_2||}$. This first type of search simply returns a list of materials ordered by similarity to the query.

The second type of search the system supports addresses the need of instructors who might want to find materials to integrate in their already classified course. To achieve this, the system performs a ranked search for all the tags of the class, and present a similarity graph of materials. From the query set S_1 (the existing class) and the results set S_2, we build a bipartite graph $G = (V = S_1 \cup S_2, E)$ where there is an edge between a material $m_1 \in S_1$ and a material $m_2 \in S_2$ if their similarity is high, e.g., if $\cos(m_1, m_2) \geq T$, for a threshold T. The bipartite graph is displayed to the user to identify relevant materials and where they could be used. (Example in Figure 5.)

3.5 Design

The CS Materials system is built as a web service. Our current implementation uses the ACM 2013 CS Curriculum Guidelines to classify assignments. The service is hosted on Heroku. The data is modeled relationally and is stored in a postgreSQL database. A Django web server provides a RESTful API to the service and serves webpages to provide the main interaction with the service. Webpages are made dynamic by the use of JavaScript, the system supports dynamic queries thanks to the jQuery library that enables asynchronous communication with the RESTful back end. Interactivity and visualization is made possible thanks to the D3 JavaScript library[9]. In the database, each assignment is associated with a title, authors, URL and description. The classifications are usually hierarchical and therefore they are represented with a key, the key of the parent, a string description, and type (separating topics and learning outcomes). Tags, items in the classification, dataset used, and authors are associated with an assignment using a many-to-many relationship (in join tables).

4 RESULTS: USAGE EXAMPLES

The authors have classified various materials including published nifty assignments and some of their classes. Author Subramanian classified a Data Structures course. Author Saule classified a different Data Structures course. Author Payton classified Software Development Projects, a capstone project course. External users of the system are currently in the process of classifying some of their courses. In total, over 300 materials have been entered. Our findings and an evaluation of the system are based on these experiences.

4.1 RQ1: Time to Classify Materials

The ACM classification is complex and extensive. It is necessary to go through the classification a few times to understand the distribution of topics and learning outcomes and their relevance to a course. For instance, a typical data structures course will map to topics from at least three Knowledge Areas. Basic data structure concepts, such as recursion and queues, are in the *Software Development Fundamentals* area. Core data structures topics, such as sorting, Big-Oh notations, and search trees, are in the *Algorithms and Complexity* area. Meanwhile, abstract data structures concepts, such as trees and graphs, are in the *Discrete Structures* area.

Classifying a course is a well invested day of work. The first few materials that are mapped to the classification scheme generally take more time as one learns about the curriculum guidelines. After the initial learning curve, mapping the materials goes much faster. Overall, mapping an entire class (say 10 slide decks, 6 assessments) takes on the order of 8 hours of work. We expect that the lessons learned from initial classification efforts will eventually result in a friendlier interface that will help reduce the time it takes to map a course to the curriculum guidelines; as the system gets populated with sufficient data, we will look into more automated techniques to further reduce the required effort. While it is not a small amount of work, building the mappings enables the other tasks that we describe in this paper, and should be considered as part of the course design/preparation process, that can lead to long-term benefits.

4.2 RQ2: Mapping Courses for Better Design

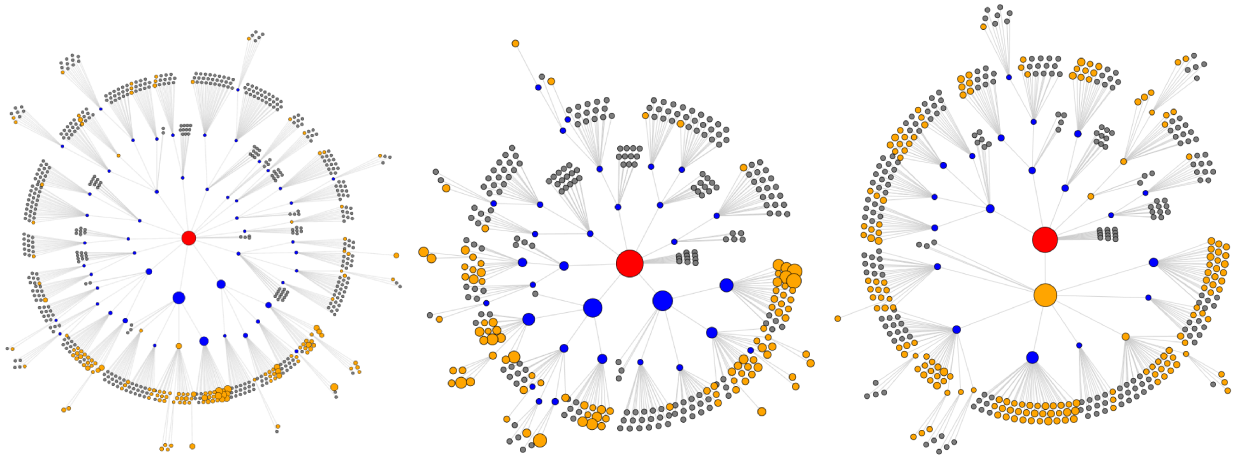
4.2.1 *Topic Coverage.*

A class may be covering many topics and still have room for more. The ACM curriculum guidelines are detailed and specific. As such, a typical course can cover many of the topics in the guidelines. For instance Author Payton's software development projects (Figure 2(c)) course covers most of the topics in the *Software Engineering* Knowledge Area, but also covers a fair number of entries from *Human Computer Interaction* and *Social Issue and Professional Practice*. Topics relating to secure software engineering could be covered in that class but probably would be difficult to integrate. Meanwhile, topics relating to ethics and professional practice could be integrated without much additional effort or class time. In this case, using the CS Materials system helped to highlight where ethics content could be woven into this course. Using this kind of information across courses, a curriculum committee may decide, for example, that a separate ethics class is not necessary and where ethics content can be integrated into other courses.

Classifying materials makes you think about design choices. While classifying, instructors notice topics and learning outcomes that the material or the class are not covering. The hit tree presented in Section 3.2 can also help the user to investigate the coverage of a particular course. This can lead to reflection on coverage gaps. For instance, classifying his data structure course made Author Subramanian (Figure 2(a)) realize that the outcomes *Choose the appropriate data structure for modeling a given problem* and *Compare alternative implementations of data structures with respect to performance* should have been central objectives and were not sufficiently emphasized.

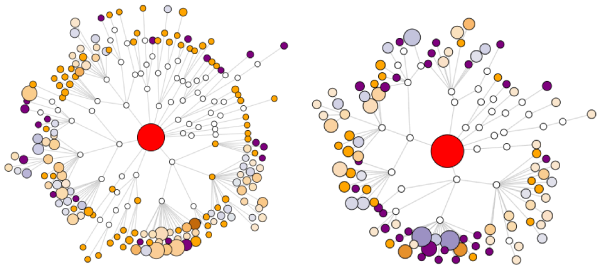
4.2.2 *Alignment Within and Between Courses.*

Figure 3(a) presents the alignment of the mappings of a data structures course between Author Saule and Author Subramanian's sections on a orange-white-purple diverging scale. Most of the heavy nodes are on core data structure topics (complexity, Big-Oh analysis, trees, graphs, abstract data types) and are light purple or light orange showing



(a) Author Subramanian's Data Structures Course (b) Author Saule's Data Structures Course (c) Author Payton's Software Devel. Projects Course

Figure 2: Classifying Learning Materials in 3 Courses: Orange nodes are selected topics/learning outcomes, blue nodes are on the path to selected nodes, and gray nodes provide context to the selections.



(a) Author Saule's VS Author Subramanian's Data Structures Courses (b) Authory Saule's Data Structures: Lecture vs Assessments

Figure 3: Alignment trees from different material sets

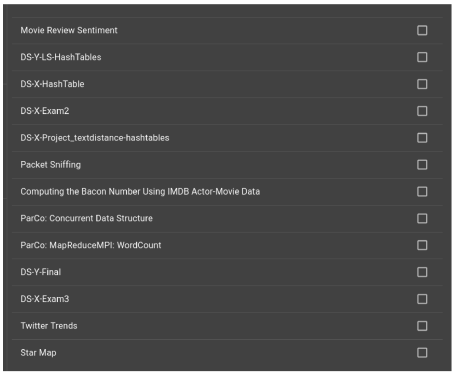


Figure 4: Search result for the Topics and Outcomes relating to Hash Tables.

that the two sections mostly cover the same topics and learning outcomes. Yet minor differences can be seen that are easily explained. Author Subramanian's section has many classification hits against API and visualization which come from using assignments based on a web-based visualization tool, and against proof techniques that are not covered by Author Saule's section. Conversely, Author Saule's section heavily hits the map abstraction, which is used as a motivating example, and empirical performance measurements, which are used to drive the importance of complexity throughout the course. This figure shows that the two sections of the same course are somewhat aligned but that better alignment could be derived. Such visualizations can induce conversations between the instructors in arriving at a more uniform curriculum that would be beneficial to students, as they enter their junior year.

Figure 3(b) presents the alignment of elements of Author Saule's data structure course: lecture notes against the assessments (assignments, projects, and exams). Most of the entries are lightly colored which shows that the assessment aligns fairly well with the lecture notes. The large nodes that are purple report to topics of empirical measurements that fit more naturally in assignments. This figure shows that the courses are reasonably well-aligned.

Checking alignment within a course can be beneficial for the instructor's own course, for instance, if a topic is covered at a level that is balanced with the assessment, or help identify topics that are not assessed. Learners also get clarity on topics to focus on and expectations on the associated assessment.

4.3 RQ3: Searching for Materials

The first search task that we demonstrate uses an explicit query of topics and learning outcomes. This type of search would be from a user who might be looking for materials on a specific topic and/or learning outcome. We perform a search using the two entries of the curriculum guidelines directly related to hash tables in "AL/Fundamental Data Structures". The result of that search is shown in Figure 4. The search returns various types of materials, assignments (some from Nifty) and projects, lecture, and exams

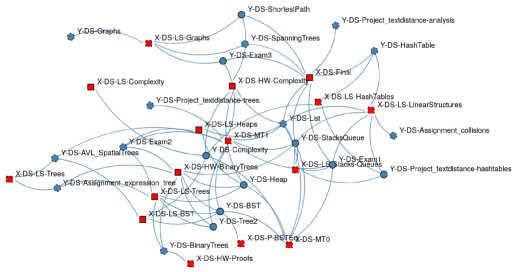


Figure 5: Similarity between Author Subramanian vs Author Saule's Data Structures Course

used in one of the authors' class. The returned results span classes on data structure classes and parallel computing. These results highlight the diversity of materials that can be retrieved by this mode of search. A keyword search would only have returned the three materials with "hash table", and would have missed other materials that discuss or use hash tables, without being primarily about hash tables. It is also to be noted that this search would not return materials covering topics similar in language such as cryptographical hashing or locality sensitive hashing, as the classification makes the difference between these different topics.

We performed a second search task where we searched for materials to integrate in one of the authors' Data Structure course. The search returned materials from a second author's Data Structure course. The system displays a similarity graph between the two sets of materials, as seen in Figure 5. We find similarities that should show: lectures on graphs being similar to lectures on graphs, spanning tree, and shortest path; a lecture on hash tables being similar to a lecture on hash tables and an assignment on hash tables; and an assignment on binary trees being similar to lectures on trees, binary trees, and binary search trees, and an assignment on expressions trees. The similarity graphs show relations between materials that cover similar topics, although not precisely the same topics. Similarity graphs can help highlight materials that a user might want to consider as replacements, for instance, a more engaging lecture or a different but equivalent assignment.

4.4 System Evaluation

We have begun conducting workshops of **CS Materials** system. To date, we have conducted two informational workshops with a total of 45 attendees. We followed this up with a hands-on session with 5 attendees to classify their courses so as to get initial feedback on the system features and interface. To date, the courses of the attendees have been partially classified and include 3 data structure courses, 1 on CS1 and 1 on algorithm analysis. The attendees comprised research universities (2), HBCU (1), Community college (1) and 4 year colleges (1). The attendees completed a survey that included 4 questions, 1) What benefits do you see in using the system?, 2) Can you identify any reasons for not using the system?, 3) How likely is it you will participate in classifying your entire course using the system?, and 4) Any other comments?

The **CS Materials** workshop was a success, with all 5 participating instructors identifying a number of benefits in using the system; understanding where they "fit in" in terms of concept coverage, finding gaps or room for improvement in materials and assessments, providing an opportunity to evaluate how well their course aligns with stated course objectives, ACM curriculum guidelines, and courses at other institutions. Beyond instructor level, advantages were also perceived at the departmental level when starting new programs of study or updating programs for accreditation. While participating in the project involves a significant time requirement and a learning curve, all 5 participants indicated that they would join in classifying their courses, and had very positive reactions in their introduction to the system, commenting "Great work" and "I think this is an amazing idea that couldn't come soon enough".

5 CONCLUSIONS AND FUTURE WORK

We have presented the design of a system and analysis tools that helps map instructors' course materials to national curriculum standards; once the materials are in the system, the visualization tools help with accessing, searching and comparing learning materials. The tools assists the instructors in checking alignment within their own course, compare two sections of the same course, and, in the long-term look at course sequences, which will be of interest to program coordinators as well as alignment with program goals. Our system is a means to 1) maintain course materials mapped to current curriculum guidelines, 2) improve course design for instructors, with well aligned course components, 3) rethink and revise courses, as a result of a deeper view of their own course content, and, 4) share resources for educators, who themselves contribute novel or engaging materials that other educators may find useful. While our system hosts learning materials of all types, it is primarily a course design and analysis system, rather than a passive repository.

Over 300 materials have been classified including three of our courses to the ACM 2013 curriculum guidelines and results to answer the three research questions: 1) In the current state, inputting and classifying a course takes about a day of work, a reasonable time-frame that can be spread over the course of a semester, 2) It is easy to understand the topic coverage of a course and alignment, leaving the decision to the instructor to make possible revisions, and, 3) The classification enables ontology based searching for specific materials that relate to a course, replace an existing material, or find an equivalent but more engaging material.

The key threat to the validity of these results will be insufficient data, i.e., the system will require sufficient number of courses for it to reach its full potential. We have begun conducting online workshops (in-person workshops are planned in the future) to disseminate **CS Materials** to the larger education community. We plan to provide incentives to instructors to get the system populated with an initial set of courses, at which point more automated tools can be brought in to help reduce the time taken to map courses. A peer review system will also be designed for vetting materials for consistency and robustness.

Future work include adding Bloom levels to improve the quality and granularity of the classification (ACM 2013 guidelines include these to an extent). We are looking into adding dimensions of engagement, such as active learning, unplugged activities, real world relevance, fun components, to enable searching for engaging materials. We will also investigate more sophisticated distance measures for building the similarity graph.

ACKNOWLEDGMENTS

This work was supported by grants from the National Science Foundation, Award Nos. OAC-1924057, CCF-1652442, DUE-1726809.

REFERENCES

[1] 2011. *Successful K-12 STEM education: Identifying effective approaches in science, technology, engineering, and mathematics.* National Research Council and others.
[2] AAAI. 2018. Model AI Assignments. http://modelai.gettysburg.edu/
[3] ACM Data Science Task Force. 2019. *Computing Competencies for Undergraduate Data Science Curricula (Draft).* Technical Report. ACM. available at http://www.cs.williams.edu/~andrea/DSTF/index.html.
[4] National Security Agency. 2018. *Centers of Academic Excellence in Cyber Defense (CAE-CD) – 2019 Knowledge Units.* Technical Report. NSA.
[5] A.C. Bart, E. Tilevich, S. Hall, T. Allevato, and C.A. Shaffer. 2014. Transforming Introductory Computer Science Projects via Real-time Web Data. In *Proc. of ACM SIGCSE*. 289–294.
[6] Austin Cory Bart. 2016. CORGIS Datasets Project: The Collection of Really Great, Interesting, Situated Datasets. https://think.cs.vt.edu/corgis/.
[7] Austin Cory Bart, Ryan Whitcomb, Dennis Kafura, Clifford A. Shaffer, and Eli Tilevich. 2017. Computing with CORGIS: Diverse, Real-world Datasets for Introductory C omputing. *ACM Inroads* 8, 2 (March 2017), 66–72.
[8] Brett A. Becker and Thomas Fitzpatrick. 2019. What Do CS1 Syllabi Reveal About Our Expectations of Introductory Programming Students?. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. ACM, New York, NY, USA, 1011–1017. https://doi.org/10.1145/3287324.3287485 http://doi.acm.org/10.1145/3287324.3287485.
[9] College Board. Fall 2014. *Computer Science A: Course Description.* College Board AP. https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-description.pdf
[10] College Board. Fall 2017. *AP Computer Science Principles, Including the Curriculum Framework.* College Board.
[11] David Burlinson, Mihai Mehedint, Chris Grafer, Kalpathi Subramanian, Jamie Payton, Paula Goolkasian, Michael Youngblood, and Robert Kosara. 2016. BRIDGES: A System to Enable Creation of Engaging Data Structures Assignments with Real-World Data and Visualizations. In *Proceedings of ACM SIGCSE 2016*. 18–23.
[12] Toby Dragon and Elisabeth Kimmich Mitchell. 2019. TECMap: Technology-Enhanced Concept Mapping for Curriculum Organization and Intelligent Support. In *Computer Supported Education*, Bruce M. McLaren, Rob Reilly, Susan Zvacek, and James Uhomoibhi (Eds.). Springer International Publishing, Cham, 191–213.
[13] EduHPC. 2018. Peachy Parallel Assignments. http://tcpp.cs.gsu.edu/curriculum/?q=peachy
[14] Groovy Graphics Assignments. Accessed July 2019. https://blog.siggraph.org/tag/groovy-graphics-assignments/.
[15] Computer Science Education Research Group. [n.d.]. https://csunplugged.org/en/.
[16] Joint Taskforce on ACM Curricula. 2013. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science.* ACM/IEEE Computer Society. https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf
[17] Joint Taskforce on Computing Curricula. 2001. *Computing Curricula 2001 Computer Science.* ACM/IEEE Computer Society. http://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2001.pdf
[18] B.D. Jones. 2009. Motivating Students to Engage in Learning: The MUSIC Model of Academic Motivation. *International Journal of Teaching and Learning in Higher Education* 21, 2 (2009), 272–285.
[19] Alvaro Monge, Beth A. Quinn, and Cameron L. Fadjo. 2015. EngageCSEdu: CS1 and CS2 Materials for Engaging and Retaining Undergraduate CS Students. In *Proc. of ACM SIGCSE (SIGCSE '15)*. 271–271. https://www.engage-csedu.org/
[20] NCWIT. 2018. https://www.engage-csedu.org/
[21] NSF/IEEE-TCPP Curriculum Working Group. 2012. *NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing : Core Topics for Undergraduates.* Technical Report. CDER. available at http://www.cs.gsu.edu/~tcpp/curriculum/sites/default/files/NSF-TCPP-curriculum-version1.pdf.
[22] Nick Parlante. 2018. Nifty Assignments. http://nifty.stanford.edu/
[23] QM. 2018. Quality Matters. https://www.qualitymatters.org/
[24] Gina Sprint and Andy O'Fallon. 2018. Engaging Programming Assignments to Recruit and Retain CS0 Students: (Abstract Only). In *Proc. of ACM SIGCSE (SIGCSE '18)*. 1093–1093.
[25] Alan Lacey Stella Fayer and Audrey Watson. 2017. U.S. Bureau of Labor Statistics Report on STEM Occupations: Past, Present, and Future. https://bit.ly/2mRxGQU.
[26] Kalpathi Subramanian. 2018. BRIDGES (Bridging Real-world Infrastructure Designed to Goal-align, Engage, and Stimulate),. http://bridgesuncc.github.io/
[27] Manas Tungare, Xiaoyan Yu, William Cameron, GuoFang Teng, Manuel A. Pérez-Quiñones, Lillian Cassel, Weiguo Fan, and Edward A. Fox. 2007. Towards a Syllabus Repository for Computer Science Courses. In *Proc. of ACM SIGCSE (SIGCSE '07)*. 55–59.
[28] Jean-Marc Vincent (Ed.). 2017. *L'informatique débranchée.* Edutions Pole.