OCTAL: The Online Course Tool for Adaptive Learning

Daniel Armendariz
Dan Garcia, Ed.
Armando Fox, Ed.

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2014-76
http://www.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-76.html

May 15, 2014

OCTAL: The Online Course Tool for Adaptive Learning

by Daniel Armendariz

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Senior Lecturer SOE Dan Garcia
Research Advisor

(Date)

* * * * * * *

Professor Armando Fox
Second Reader

(Date)

Abstract

The Online Course Tool for Adaptive Learning (OCTAL) is a tool that combines an exercise system with a concept map to allow learners to explore an underlying prerequisite structure of topics. An algorithm that estimates a learner's level of mastery highlights concepts in the graph to provide the user with a metacognitive hint about their progress through the material. Learners are guided by the prerequisite structure and knowledge inference but may navigate freely through the graph. We intend OCTAL to be a formative assessment tool that is not tied to any specific course or subject and provide authoring tools for content designers to create material. Toward the goal of being usable in a number of online courses, OCTAL has support to be embedded within online learning platforms such as edX.

Students enroll in online courses with different learning goals and, as a result, may wish to pursue their own paths through the material. OCTAL presents the underlying prerequisite structure of the material to allow learners the opportunity to decide whether or not deviation from the expert-defined path would be beneficial for their understanding. This allows students to metacognitively consider their level of mastery in a course's advanced concepts by exploring exercises without limitation and may therefore be useful to help answer the question "will this course be useful?" Similarly, for those students enrolled in a course, it allows them to decide how to prioritize consumption of content and discover which concepts they may reasonably skip, if necessary.

In order to study the benefits of metacognition with OCTAL, we authored a concept map and question set for topics from UC Berkeley's CS10: The Beauty and Joy of Computing. We presented the tool to the students of the course between their first and second midterm exams during the spring of 2014. In the study, we found no statistically significant changes in metacognition among participants who used the tool. However, analysis of participant usage of the tool reveals differences in the way learners approach concepts presented to them in a list versus in a graph. In particular, while users often followed a list of concepts in-order, learners that navigated a graph explored concepts in clusters.

Contents

1 Introduction

With the proliferation and maturity of many types of online courses from OpenCourseWare (OCW) to Massively Open Online Courses (MOOCs), several closely related but distinct concerns are apparent. First, there is a need for scalable tools to address the effective management of an enormous student population. Second, there is a need to offer students an individualized learning experience in an environment where interaction between individual students and their instructors is all but impossible.

Specifically, given the demonstrated efficacy of one-on-one tutoring and mastery learning [5], it is important to incorporate these elements into the ecosystem of online courses in a manner that is constructive to student learning. Intelligent Tutoring Systems (ITS) are automated systems that aim to provide exactly these benefits and have been used for a number of years in other educational contexts. ITS implementations are often painstakingly developed for very specific topics, such as elementary arithmetic or physics. We introduce the Online Course Tool for Adaptive Learning (OCTAL) as a framework for content to leverage both mastery learning and metacognitive prompting [18]. The framework allows course staff, instructors, or other content designers to author and deploy material to learners. We intend the tool to augment but not necessarily inform the design of a variety of existing and future online courses.

Computerized Adaptive Testing (CAT) and ITS have been the subject of active research since the 1970s. Now essential for many applications, such as the Graduate Record Examination (GRE) and Test of English as a Foreign Language (TOEFL), the need for accurate and robust adaptive assessment has led to a body of research that has approached the problem from a number of angles. Related work in latent knowledge estimation, fundamental to intelligent tutoring systems designed for real-time adaptive learning, have similarly offered a number of solutions for building and utilizing coherent models of student knowledge [10]. Meanwhile, research in education technology has investigated the benefits of scaffolding and metacognition in learning [39].

OCTAL falls at the intersection of these fields. We investigate benefits to a learner's metacognitive awareness through estimation of latent knowledge traits and displaying an expert-defined structure of concept dependencies. Although popular CAT algorithms such as Item Response Theory (IRT) provide accurate estimation of latent knowledge traits at a given point in time, we also intend to provide continuously updating mastery estimates that change as the user learns. We have therefore chosen to model student knowledge using a Bayesian Inference Network (BIN), a model of latent knowledge that supports hierarchical structures and changes to latent knowledge [1, 8].

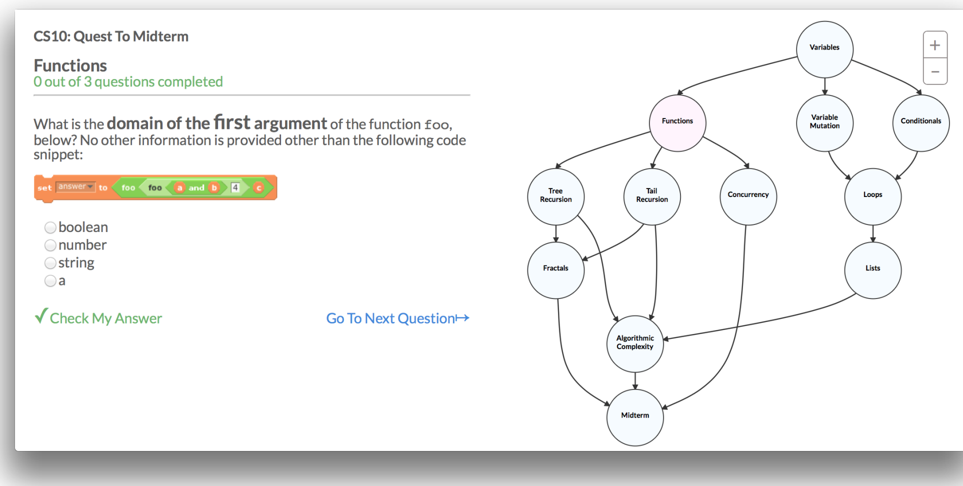

Figure 1: OCTAL user interface with concept graph and exercise.

We intend OCTAL to be used as a tool for formative self-assessment of students participating in any course whose topics lend themselves to a dependency structure (Figure 1). At a coarse level, OCTAL involves the integration of a number of concepts well studied in the literature. First, we presuppose a hierarchy of concepts comprise a given course; the construction of this hierarchy is itself a field of study and is outside the scope of this work. We therefore intend an instructor or expert in the field to create a concept graph in OCTAL before using the tool as part of their instruction. Second, we maintain a student model for each learner, including features about their responses to assessment items offered by our tool and other metadata. Third, we utilize a predictive algorithm incorporating this hierarchy of concepts and these observed features into a BIN in order to make predictions about a student's current knowledge state. Finally, the combination of these factors is intended to provide metacognitive hints toward the underlying structure of a sequence of concepts.

We built a number of features into OCTAL so that it may be used outside the scope of a research project and embedded in real-world courses. Included in these features are tools to create content: graphical editors allow content creators to design concept graphs and exercises without requiring advanced technical knowledge. In addition, OCTAL supports the IMS Global Learning Consortium's Learning Tools Interoperability (LTI) specification, which allows it to be embedded within learning platforms, such as edX, with relative ease. These features are described in detail in Sections 4 and 5.

2 Related Work

2.1 Concept Maps

Novak and Cañas describe the specifics of a concept map in their 2006 technical report [20]. A concept map is a graph with nodes representing concepts (each node is typically surrounded by a square or circular border with the concept name within) and edges linking nodes with a relationship. In the map, the edges typically contain a "linking word" or "linking phrase" that describe the relationship between the nodes. In essence, a basic sentence providing semantic meaning can be formed by reading the concept name of a node, the linking phrase of its edge to a subsequent node, and the concept name of that connected node. Further, the hierarchy must be structured such that the least specific concepts are at the top with increasing specificity towards the bottom. A sample concept map is shown in Figure 2.

There are several ways that OCTAL differs from this definition, and we therefore use the term "concept graph" instead of "concept map" to refer to our implementation. The most significant change is the elimination of linking words between nodes. We justify this because, according to Novak and Cañas, "it is best to construct concept maps with reference to some particular question we seek to answer" [20]. Our graphs are intended to include coarse concepts in a knowledge domain, rather than those seeking to answer a specific question, and primitive linking words would therefore be too restrictive for complex relationships.

The Concept Mapped Project-based Activity Scaffolding System (CoMPASS) displays a concept map and associated text for a selected node (Figure 2) [23]. The CoMPASS concept map is displayed on a website using a fisheye technique to provide visual priority to the selected node and decrease the visual impact of nodes in the graph as their degree increases from the focus. This display provides clear relationships between the selected node and its immediate neighbors, but it can cause confusion in understanding the placement of the node in the global context [31]. One notable differentiation between CoMPASS and OCTAL is that a selected node in the concept map displays further information rather than an exercise.

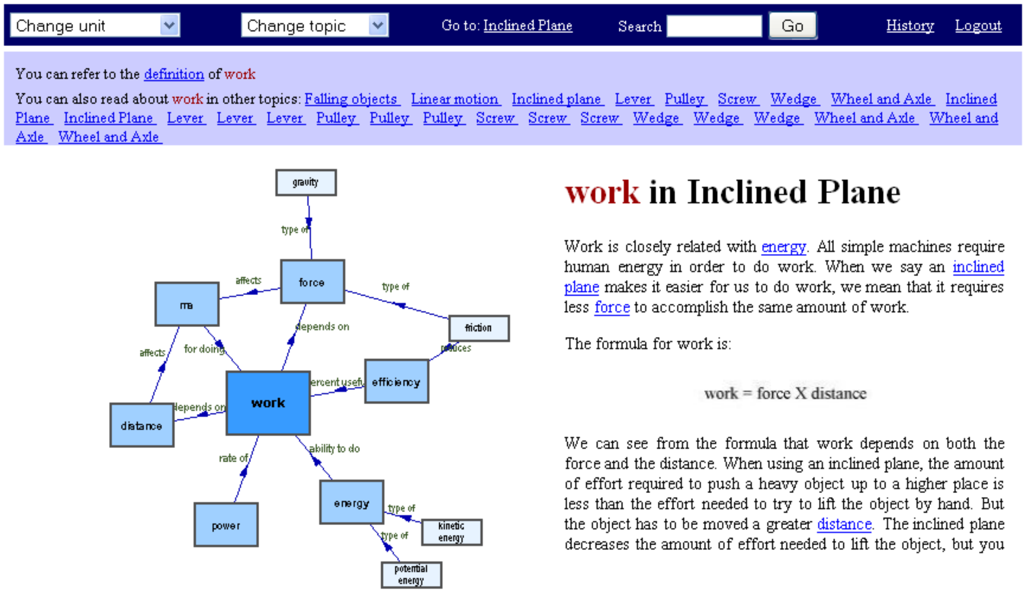

Figure 2: CoMPASS screenshot displaying a concept map and related text.

Several research projects have emerged from the CoMPASS project. Puntambekar, et al., divided a group of middle school science students and provided one set with CoMPASS and the other group with an outline of the same topics. Although the two groups displayed similar fact retention, they found the group that navigated the material with a concept map to have a deeper understanding of the content [24]. Hübscher and Putambekar performed a case study with the system, attempting to make design choices from the context of pedagogy improvements over technical constraints [19]. Embedding the CoMPASS project into a middle school science curriculum resulted in students having a better understanding of the relationships between concepts [25], though the benefit arose from discussion and reflection and not simply providing a visual display that showed relationships.

Romero and Ventura describe additional educational concept graph technologies in their review paper of modern technology in educational data mining [30], though the automatic creation of concept graphs is outside the scope of this report. From an education design perspective, concept maps are widely applied in science education [6] and are consistent with several learning theories including Anderson's knowledge representation [2], Duffy et al.'s constructive learning [11], and Novak's meaningful learning [20].

Navigation and display of hierarchical graphs is also well-studied [12]. OCTAL itself, however, relies on `kmap`, a graph display library developed by Colorado Reed at UC Berkeley and used in the open-source project Metacademy [29]. We provide additional detail on this technical dependency in Section 5.1.

2.2 Knowledge Estimation

One oft-employed model of student knowledge estimation is Item Response Theory [33]. Employing the Item Response Function (IRF), and a prior distribution of latent knowledge over a population, IRT serves as a means to summatively assess a student's latent knowledge relative to that population. Typically, the algorithm iteratively selects maximally informative assessment items until a latent trait has been estimated within some small bound of error. The IRF is a 3-parameter sigmoid function unique to each assessment item and includes an assignation of item difficulty, discriminability, and the probability of an uninformed but correct guess. Other less popular variants include 2- and 1-parameter models which are derived by omitting the probability of guessing and item discriminability, respectively [17]. Additionally, multidimensional IRT has been shown to perform well in the simultaneous measurement multiple latent traits [26].

One limitation to Item Response Theory, however, is that it models a student's knowledge at a particular point in time; there is not, generally, an allowance made for learning that occurs during an assessment. While IRT has been shown to be well-suited for measuring multiple dimensions of student ability, such as vocabulary skill and reading comprehension, it is not easily adapted to measuring a hierarchy of traits. The Assessment and Learning in Knowledge Spaces (ALEKS) system is an example of an IRT implementation that also employs a hierarchical knowledge structure, adapts the IRF to search through a space of all possible student positions in the hierarchy and iteratively attempts to select the most probable knowledge state for a student [13].

Another frequently used model of student learning employed in intelligent tutoring systems is Bayesian Knowledge Tracing (BKT), first described by Corbett and Anderson [10]. BKT relies on the strong assumption that knowledge of a topic is binary—either known or unknown. It calculates the posterior probability that a student knows a concept based on observed data and a four-parameter model that strongly impacts its predictions. The four parameters are defined as follows:

- $P(L_0)$: The prior probability that a student understands a concept before being assessed.
- $P(G)$: The probability of guessing, or responding correctly to a question given no knowledge of a concept.
- $P(S)$: The probability of slipping, or responding incorrectly given that a student does know a concept.
- $P(T)$: The probability that a student will acquire knowledge of a concept after each assessment.

These parameters are usually generalized over the entire model, though efforts have been made to analyze the efficacy of their contextual assignment [4]. The probability that a student has attained mastery of a concept is recalculated after each question is answered, using Bayesian inference. This process is repeated until some condition is fulfilled, typically until a threshold probability of mastery is achieved.

BKT is intended to be a formative assessment tool to approximate mastery learning. Although it is based on a simple model, it has been shown to perform consistently well even when compared to more complex models such as Performance Factors Analysis [14]. However, BKT is traditionally implemented to measure only a single trait and must be adapted in order to model a knowledge over a hierarchy of concepts. To this end, we have drawn upon features of BKT, but with an approach tailored to the use of a concept graph, as detailed in Section 5.6.

3 Design

3.1 Principles

OCTAL's design has been guided and informed by several principles, described below.

3.1.1 Scale and Reach

The issue of scale is often fairly well-defined in the context of large online courses, including MOOCs. Frequently, issues of scale center on the vast number of students participating in the course. Coburn, however, proposes a broader definition of scale that contains four dimensions: depth, sustainability, spread, and shift in reform ownership [7].

In this definition, *spread* (or the overall impact in numbers of students or institutions) remains an integral part. However, she argues that to be "at scale", an innovation must also effect deep change in practice (*depth*), be easily implemented and reproduced in other curricula (*sustainability*), and become naturally embedded in a curriculum (*shift in reform ownership*).

These ideas have similar meaning in Computer Science under the term "reach," and our design has been informed by these concepts. Several design principles and implementation details, as shown below, attempt to broach this subject: we intend that the tool be used by courses and implemented in future MOOCs.

3.1.2 Openness and Integration

OCTAL is open-source and built upon open-source tools (Section 5.1). It is our hope that others may contribute to the tool and modify it for their own purposes. In addition, we intend usage to be as unrestricted as possible. The system does not enforce a specific order that students proceed through concepts or exercises.

Any user may also contribute by authoring content. A single OCTAL unit represents a concept graph and the set of all exercises that correspond to that graph. Units are accessible by anyone with a URL, and by default every unit is shown on a list of all available units. Content authors may elect to hide the unit from public listing, however. Further, no user credentials are required to use the tool or to contribute content to the system.

By default, OCTAL can be used as a stand-alone tool. However as OCTAL supports IMS Global Learning Consortium's Learning Tools Interoperability (LTI) protocol as a provider, it can be integrated into Learning Management Systems (LMS) that are LTI consumers, such as edX.

3.1.3 Unique Learning Paths

A main guiding principle behind the concept graph is that each learner may wish to pursue their own path through the material. Students may or may not learn the material best in the linear order that an instructor presents it, as learners bring unique understandings and misconceptions [3]. Presenting the underlying prerequisite structure of the material allows learners the opportunity to decide whether or not deviation from the expert-defined path would be beneficial for their understanding.

A side-effect of this principle is that OCTAL may be useful for learners asking the question "will this course be useful?" Students may gauge their level of mastery in a course's advanced topics by exploring the exercises in those concepts without limitation and in the order they wish. Similarly, it allows students to decide how to prioritize consumption of content and discover which concepts they may reasonably skip, if necessary.

3.1.4 Scaffolding

Although OCTAL provides learners the opportunity to navigate the material in any manner they wish, the tool also provides metacognitive hints through knowledge estimation. By displaying an estimated level of mastery for concepts, an alternative learning path is suggested and helps scaffold a learner's understanding as they work through the material. OCTAL aims to show learners the expected path that targets their Zone of Proximal Development

(ZPD). Placing learners in their ZPD allows them to engage in more advanced thinking and activities [37].

Although Vygotsky never coined the term "scaffolding", the idea of the ZPD provided the framework for the term. Early scaffolding was defined as an adult guiding a learner through tasks just beyond that learner's capacity [39]. Its definition has evolved to mean a specific set of requirements: a scaffolding technology must continuously assess a learner, the scaffolding must fade away over time, and the learner must be an active participant in the learning process [19, 21, 35]. However, given logistical and conceptual difficulties in emulating a human tutor, it is useful to split scaffolding into "hard" and "soft" types; the former being technology-mediated and the latter provided by experts [34]. More specifically, scaffolds can be either explicit (more constrained) or tacit (less constrained) [16].

OCTAL falls in the categories of "hard" and "tacit" scaffolds since it allows the learner to navigate the concept graph freely despite the metacognitive hints. The scaffolding is present but optional for learners; we therefore implement the first and third requirements from the strict definition above, but not the second.

3.1.5 Formative Assessment

OCTAL is not intended to assess students as part of their grade. Learners are encouraged to honestly respond to exercises, irrespective to the number of times they get it wrong.

This principle has several implications. For instance, an early technical decision resulted in exercises marked for correctness on the front end, where savvy users could manipulate the code and send falsely correct attempt events to the server. Although this would allow a learner to "cheat" the system, the result would have no bearing on a grade and would therefore only impact the knowledge estimates provided to the user.

There is no facility for instructors or course staff to view a learner's progress through a graph and unit. However, this may be a possible future enhancement to the system.

3.2 Limitations

Although we strive to abide by these principles in as many ways as possible, there are several limitations to the system.

For every exercise attempt, the knowledge inference algorithm requires a tuple of the selected concept and a correctness value. In principle this allows for a wide variety of question types, but the current implementation allows only for multiple choice and true/false.

In addition, a third party exercise system as part of a LMS would, in theory, be able to supply the knowledge inference algorithm with the needed data by performing a request to the OCTAL server with a concept and correctness value. However, this facility is not implemented and exercises are therefore required to be input into OCTAL's content editor.

Students' optimal learning paths may differ greatly from the expert-defined graph. There is currently no way of specifying multiple prerequisite structure for material without defining separate units for each alternative.

Finally, OCTAL is also designed to be used by individual learners and therefore has no built-in capability for collaborative efforts.

3.3 Cultural Considerations

The population of students taking online courses varies wildly. However, in order to maintain focus while implementing OCTAL, we have made several assumptions regarding its use.

First, the tool's premise depends on the visual display of a concept graph. This may not be reasonable for users that require the use of a screen reader, such as those with visual impairments. For this reason, we consider accessibility support to be poor.

We have attempted to make considerations for users with differences in color perception. Concepts that the knowledge inference algorithm considers mastered, for instance, are not only highlighted in green but also given a thicker border.

There is no support to filter, flag, or revoke offensive content. This initial version trusts content authors to be respectful.

The website and instructions assume an understanding of English. Content authors may submit graphs and content in other languages, but the site itself, including usage instructions, has no support for localization.

4 User Interface

OCTAL is an HTML- and JavaScript-based application hosted on its own independent server. The user interface is informed by the needs for four different types of users: *visitors*, *learners*, *participants* in research studies, and *content authors*. A single OCTAL unit represents a concept graph and the set of all exercises that correspond to that graph. The interfaces for each of these types of users is described further below.

4.1 Visitor

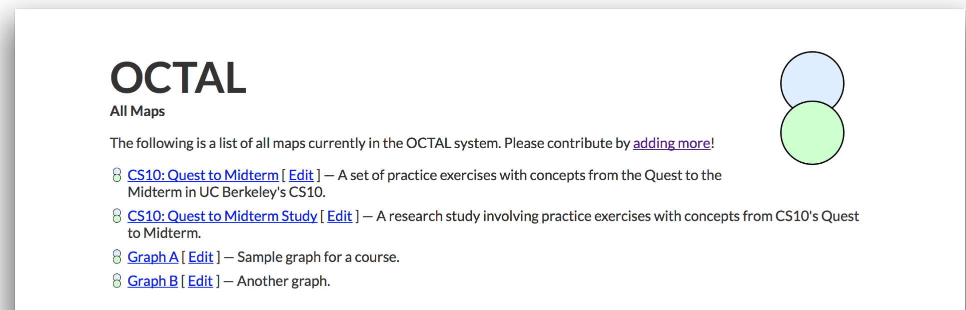

Figure 3: User interface listing all public units.

Users that visit the website without having been provided a direct link to an OCTAL unit are presented with a list of publicly-listed units (Figure 3). Clicking on any name allows a visitor to transition to a learner (Section 4.2) for that unit.

4.2 Learner

The primary interaction a user will have with OCTAL is as a *learner*[1]. These users have been given a direct link to the unit (for example, through a course or online tutorial) or have selected a unit from the unit list (Section 4.1). Upon loading the page, learners are presented with welcome text and a two-dimensional directed acyclic graph that represents the concept graph.

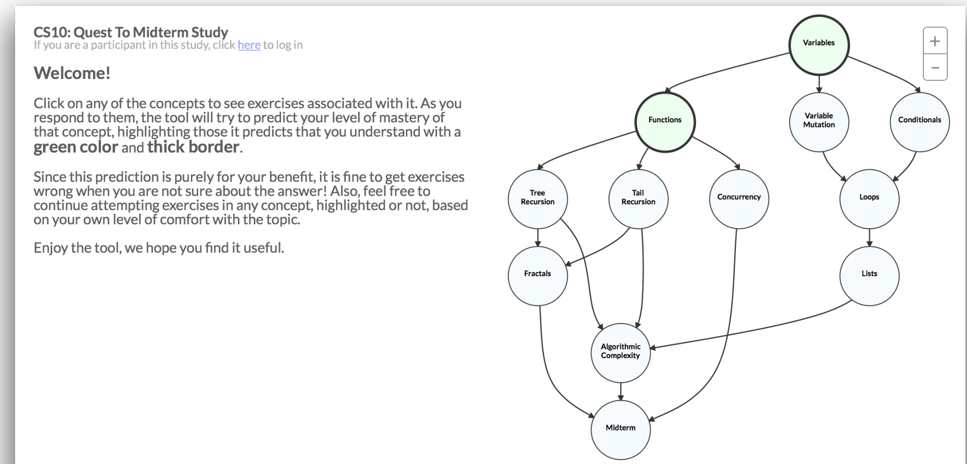

Figure 4: OCTAL interface with concept graph, welcome text, and inferred knowledge calculated from this user's prior exercise attempts.

A node represents a single concept and directed edges indicate the hierarchy of prerequisites. Prerequisite root nodes are placed at the top and post-requisite leaves placed at the bottom with edges pointing from a prerequisite to its post-requisite. Post-requisites are always placed below their prerequisite nodes. Outside of these restrictions, the placement of nodes in the graph is a result of `kmap` attempting to maximize visual clarity of the graph. The library arranges the nodes to minimize edge intersections but maintain reasonable viewable dimensions that can be displayed in a browser viewport. A consequence of this priority on visual clarity is that node placement in horizontal and vertical axes are not intended to convey structure

[1]We use the term *learners* instead of *students* since the types of exercise graphs created in OCTAL may include those outside of the scope of a class or course.

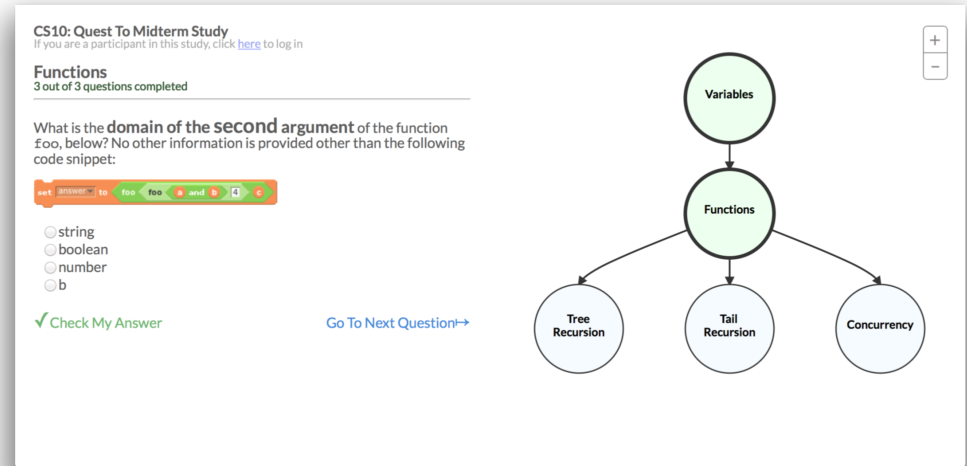

Figure 5: Exercise interface for a selected concept showing a subgraph of nodes with degree 1 from that concept.

or context. In other words, nodes found on the same vertical placement are not necessarily intended to provide contextual clues as to their relative difficulty or importance to others in the same row. Likewise, placement on the horizontal axis has no intended meaning.

Learners are allowed unrestricted access to any concept in the graph. Clicking on a concept results in a zooming animation to a subgraph of the selected node and all nodes with degree one from it (the set of immediate pre- and post-requisites). The introductory text is replaced by an exercise from the bank of exercises defined for that concept. In addition, learners are presented with text indicating the number of remaining incomplete questions and an option to skip the current exercise, if more than one incomplete exercise remains in the set (Figure 5).

We define "incomplete" exercises as those that have not yet been correctly answered. When a learner completes an exercise, it is not shown again to the user until they have completed all remaining questions in that concept. If a learner selects "Go To Next Question" another exercise will be randomly displayed from the remaining set of incomplete exercises.

Learners check their work by selecting a response and clicking the "Check My Answer" button. Text will appear directly above the button that indicates whether or not the answer was correct. Learners may then choose to proceed to the next question by selecting "Go To Next Question" or attempt the current exercise again. Currently, the learner receives no feedback other than correctness. OCTAL does not support answer hints to ensure that the knowledge estimation algorithm (Section 5.6) receives an accurate stream of attempts that is not influenced by other factors.

Every question attempt causes a request to the server that records the attempt's correctness and performs an updated knowledge inference computation, based on that described in Section 5.6. The result of the computation contains a list of concepts considered mastered and is passed from the server back to the client. Those nodes contained within the list are subsequently highlighted in a green color and the border given a thicker stroke to indicate to the learner that the concept is mastered.

Learners may proceed to subsequent exercises at any time during this process. Alternatively, the learner may navigate the graph by clicking on the current concept to zoom the graph out and select another. The knowledge inference provides a visual clue as to where learners may want to pursue further exercise in the current concept, advance to post-requisites, or retreat to prerequisites. Users are not limited, however, and may continue with exercises in the selected concept or select another concept at any time.

4.3 Participant

OCTAL units may be defined by content authors to be part of a research study. In this case, users that interact with the unit become *participants*.

When a user first visits the unit they are forced to either identify themselves as not a participant or log in with a participant identifier (generally supplied offline after a consent form has been signed) in order to access the tool (Figure 6). Non-participants are immediately forwarded to the unit and interact with it in an identical manner as learners (Section 4.2). If a user instead identifies as a participant and enters a valid participant ID, they are forwarded to a URL hosting a pre-survey, if the unit author has defined one. Upon completion of the survey, or if none was defined in the unit, a link returns them to the OCTAL unit where they may begin to interact with the tool. Half of these participants see the UI in the same manner as defined in Learners (Section 4.2), the other half are provided a linear view.

The linear view behaves in a similar manner as the interface presented in

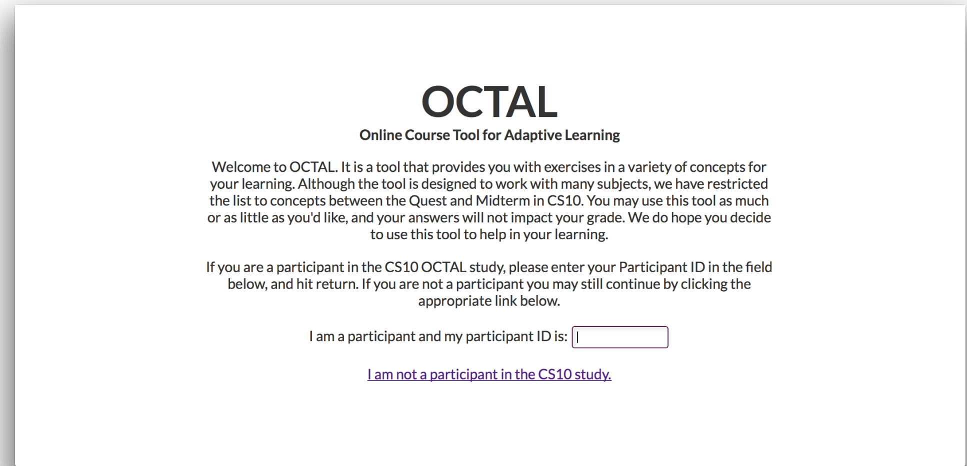

Figure 6: Participant login form.

Section 4.2, except the two-dimensional graph is collapsed into a linear list of concepts and is found on the left side of the screen (Figure 7). Knowledge inference continues to highlight the concepts predicted to be mastered, but the underlying structure of the graph is hidden from these participants.

Participants that visit the unit after a content author specifies the study as "completed" are forced to another third-party URL that hosts a post-survey, if the unit author has defined one. Upon completion of the survey, or if none was defined, learners are forwarded back to the OCTAL site where a completion page thanks them for their participation. They may continue with the unit as learners once they have completed the post-survey and view the study completion page.

4.4 Content Author

Any user may elect to be a *content author* to create an OCTAL unit. Designing a unit involves first specifying metadata and designing its concept graph (Appendix D.1) and then specifying a set of exercises (Appendix D.2).

The user interface is a standard HTML form with fields for the graph name, a description, an option to hide the unit from the units list (which

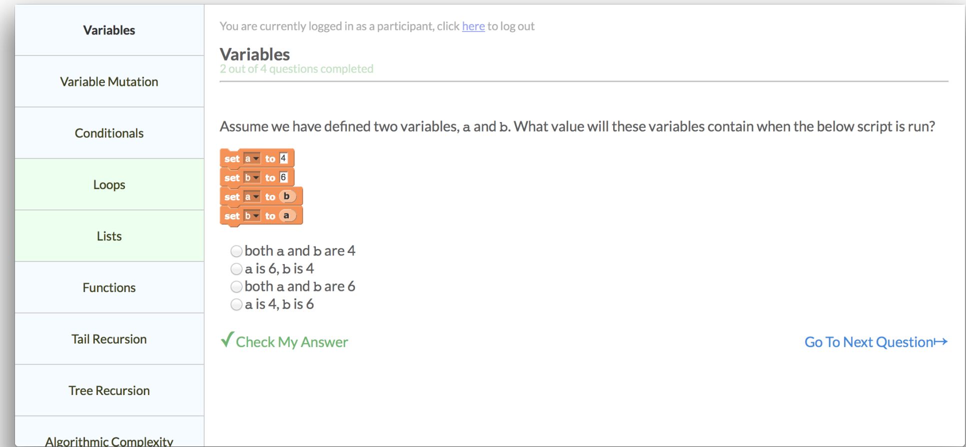

Figure 7: Linear concept view.

will hide the unit from *visitors*, as in Section 4.1), and an option to allow the unit to be a part of a research study (making users of the unit *participants*, Section 4.3). Finally, the graph editor, provided by `kmap`, allows content authors to visually create the concept graph for the unit (Figure 8). Shift-clicking creates nodes and shift-click-and-drag creates directed edges between them. Authors may change concept names and delete concepts from the graph. An option is provided to optimize the graph placement so that the content author can get a preview of how the concepts will be arranged to a learner. Advanced content authors may instead elect to input the graph as valid JavaScript Object Notation (JSON) through a checkbox on the page. The expected JSON structure is defined in Section 5.4.

Content authors may also specify the unit as being part of a research study. In this case, an additional form appears that allows them to include (optional) pre-survey and post-survey URLs and a list of comma-separated identifiers for participants. The action of the surveys is described in *Participants*, Section 4.3. The last identifier in the list of participant IDs is used as a special-case representing a non-participant.

OCTAL generates a secret key for every new unit and displays this key at the bottom of the form. Content authors must use this secret key in order to edit the unit and its exercises in the future. *Content authors* may share that key with other users (say, other course staff) to delegate responsibility

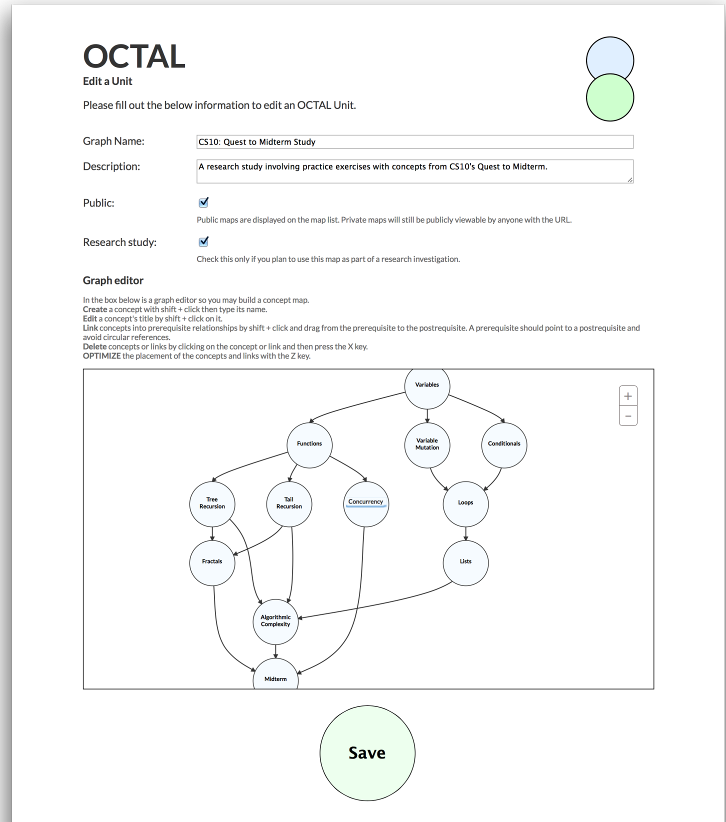

Figure 8: OCTAL unit creation and editing form.

for the content.

Upon submission, the form reminds the author to record the key. A second click submits the form, and the author is forwarded to the unit on success or, if a validation error occurred, is returned to the creation page with a request to correct the errors.

Editing the unit shows a similar user interface as the creation form with some small differences. First, the content author must enter the unit's secret key before being shown the editing form. Second, if the unit is part of a research study, the participant list field is disabled by default. The page warns users that changing the list will cause all participant IDs to be removed from the server before being recreated, but content authors may click through

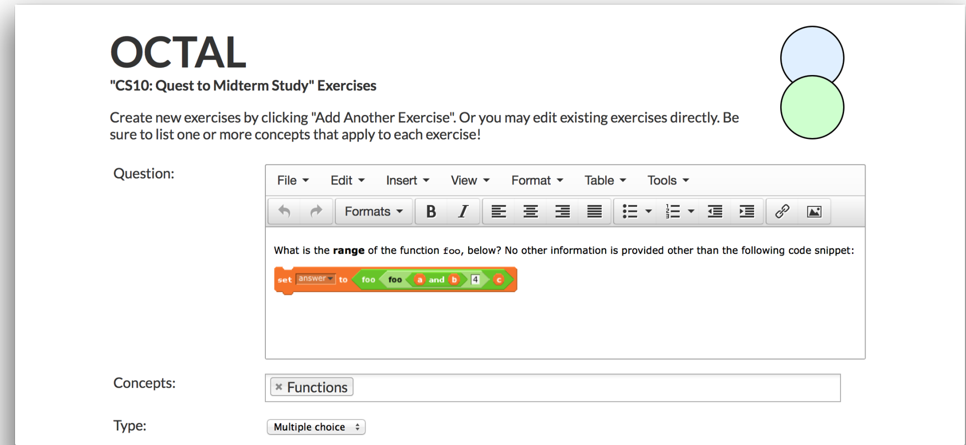

Figure 9: Exercise authoring and editing.

the warning to modify the list. Third, the option to add a graph via JSON is removed and all edits must be done using the graphical editor. Finally, the unit's key is not displayed to the author.

Adding and modifying exercises uses a separate form that can only be accessed once a graph has been created and requires the content author to enter the graph's secret key. In this interface, content authors may use a rich text editor interface to design the exercises (Figure 9). The rich text is supported by HTML, and advanced editors may elect to edit this markup directly. Each exercise is specified a type (multiple choice or short answer, though the OCTAL front-end currently only supports multiple choice). Correct answers are input by a text field, which also supports HTML. Distractors must also be specified for multiple choice questions.

Each exercise must be associated with at least one concept. A text field supports listing concepts in a system similar to tags: it auto-completes concept names and allows the author to delete or add multiple concepts at a time.

5 Implementation

OCTAL is written in Python using the Django framework on the back end and HTML5 and JavaScript with the Backbone framework on the front end. The knowledge estimation algorithm is implemented with support from the PyMC library.

5.1 Metacademy and `kmap`

Metacademy is an open-source project created by Colorado Reed from UC Berkeley and Roger Grosse from MIT [28]. It is a community-driven tool meant to create and display hierarchical concept maps for all domains of knowledge [29]. OCTAL is a natural extension to this, being an exercise system that itself leverages a dependency graph.

Metacademy is a Python project built with the Django framework on the back end with JavaScript and the Backbone framework on the front end. The graph visualization in Metacademy is provided by Colorado Reed's excellent and open source [27] `kmap` library, written in JavaScript and utilizing the Data-Driven Documents (D3) library.

Early versions of OCTAL heavily relied upon Metacademy and were implemented as a Django app and a single additional Backbone view to provide an exercise system built on top of Metacademy's infrastructure. The latest version of OCTAL continues to rely on several important aspects of the Metacademy codebase:

- `kmap`: the graph visualization library.
- Metacademy's graph creator: a single Backbone view that extends `kmap` to provide node and edge creation and manipulation.
- Metacademy's code structure and design serve as an influence.

Other than the above items, OCTAL uses original code for its back and front ends.

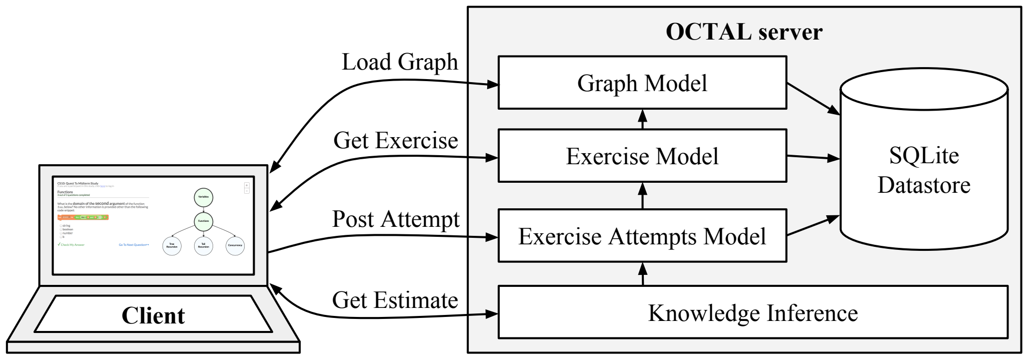

Figure 10: Data flow and system architecture. A learner loads a graph and the system will dynamically load exercises through AJAX requests while the user navigates. The results of an exercise attempt are sent via AJAX POST to the server. A second query requests an updated knowledge estimation computation from the server and nodes are highlighted based on the result.

5.2 Data Flow

As learners complete exercises, the front-end evaluates the answers for correctness, and submits an exercise ID, a correctness value, and a concept ID to be stored on the server. We elected to perform correctness evaluation on the front-end for simplicity since OCTAL is not intended for summative assessment. This would also allow OCTAL to be enhanced in the future to enable more advanced types of exercises whose answers can be evaluated in JavaScript.

The back-end stores the submitted values along with timestamp information and a user ID in a database. With Django's `lazysignup` package, we can maintain IDs even for users that have not registered for the site. If the unit is part of a research study, we instead record the participant ID. After recording the values, the back end passes a user's full set of responses over all concepts in a graph to the knowledge estimation algorithm (Section 5.6) for processing. We record only the first attempt per requested exercise to eliminate accidental or intentional duplicate submissions. In other words, OCTAL will only accept at most one attempt for every viewed exercise, and will recycle IDs for unused attempts. This ensures an accurate stream of exercise attempts and ensures the validity of the knowledge estimation.

5.3 Access Control

We elected to use a randomly generated key system over a login scheme with access control lists because it lowers the barrier to entry for new users to participate (no login necessary) and would allow a teaching staff to share editing duties without dealing with a complicated privileges system. Since we expect the primary means of discovery to be a content author sharing a unit's direct link to a learner, we do not expect spam to have a large impact on the learners experience. However, it may be necessary in the future to rate-limit graph creation or use human verification techniques (such as CAPTCHAs) to prevent denial-of-service attacks on the system.

Secrets are randomly generated 16-character strings made up of symbols and unambiguous alphanumeric characters and stored as part of a unit's metadata.

5.4 Graph

The graph is stored as an adjacency list in a Django model. A concept is defined by its `graph ID`, `title`, a `tag`, and its `dependencies`. Each concept is assigned to a specific graph through the `graph ID`. A graph's set of concepts are therefore queried by finding all matching concepts for a given `graph ID`.

Since OCTAL URLs frequently contain `tags` to refer to specific concepts, they are `title` strings converted to be compatible with HTTP GET queries. The `title` string is lowercased with symbols removed and whitespace replaced with underscores. A `dependencies` field recursively refers to one or more other unique concepts.

The `kmap` library accepts an adjacency list with additional metadata for each concept. This structure is documented in Figure 11, and can be used to input a graph as JSON when creating an OCTAL unit (Section 4.4).

The data in `id` and `tag` fields are required for the server to perform consistency checks for orphaned nodes, cycles, missing or bad concept ID references, or invalid strings. However, once the graph is validated the server overwrites these values: the `id` becomes associated with the primary key of the model and the `tag` is generated from the `name`, as described above.

```
[
  {
    "id": "<id>"
    "name": "<Concept Name>",
    "tag": "concept_name",
    "dependencies":
    [
      { "source": "<id of other concept>"},
      ...
    ]
  },
  ...
]
```

Figure 11: JSON graph adjacency list definition, including metadata for each concept.

5.5 Exercises and Attempts

Exercises, exercise answers, and exercise attempts are stored in separate Django models.

A single exercise attempt refers to a specific user attempting one exercise within a unit. An exercise attempt is generated when a learner requests an exercise (by clicking on a concept or completing an exercise and requesting a subsequent), with a flag (initially false) indicating the attempt's submission status. When a learner submits an attempt, the correctness value is stored and the flag is changed. Exercise attempt IDs may be recycled if a user requests the same exercise without having submitted a prior attempt. This scheme ensures that learners cannot submit an attempt multiple times or without first seeing the exercise.

5.6 Knowledge Estimation Algorithm

We employ a Bayesian Inference Network (BIN) graphical model of student knowledge to incorporate the concept graph into our predictive model. One notable strength of such a model when compared to BKT or IRT is its abil-

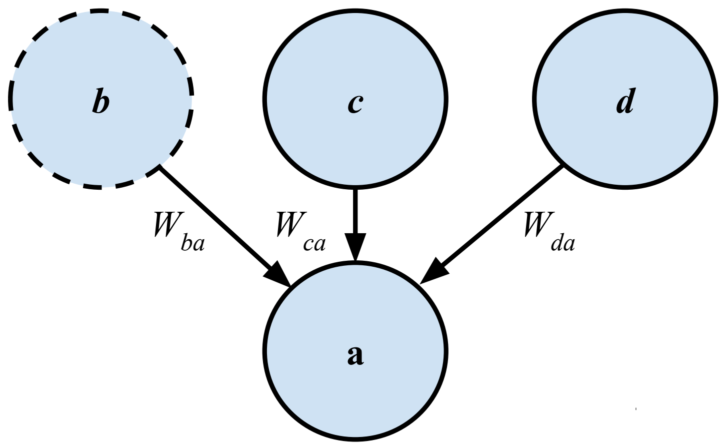

Figure 12: Dependency graph of concepts in which b, c, and d are prerequisites to a, with each edge weight W. Concept b is not learned.

ity to incorporate the dependency structures we require. Conceptually, our model for hierarchical knowledge tracing borrows the parameters for probability of guess, $P(G)$, and slip, $P(S)$, from Bayesian Knowledge Tracing, but eschews $P(T)$, the probability of knowledge acquisition, and replaces the initial learned probability $P(L_0)$ for a concept a with Equation 1.

$$P(L_{a_0}) = (P_{max} - P_{min}) \frac{\sum\limits_{i \in C} W_{ia}\varphi(i)}{\sum\limits_{i \in C} W_{ia}} + P_{min} \tag{1}$$

Where P_{max} and P_{min} are globally defined as the maximal and minimal prior probability of learning, respectively, with initial untrained values set at $P_{max} = 0.5$ and $P_{min} = 0.05$. Further, each edge between concepts i and a have weight W_{ia}. However, this equation is simplified in practice because OCTAL currently has no capability to specify edge weights; this is discussed further below. The activation function $\varphi(i)$ is defined in Equation 2.

$$\varphi(i) = \begin{cases} 1, & \text{if } P(L_i) > T_L \\ 0, & \text{if } P(L_i) < T_L \end{cases} \tag{2}$$

Where $T_L = 0.75$ as an untrained global threshold indicating the probability that a concept is learned. Equation 1 ensures $P_{min} \leq P(L_{a_0}) \leq P_{max}$ while each node in the set of prerequisite concepts C contributes a variable amount to the estimate that a student understands concept a prior to any observations. The activation function reveals our assumption that a knowledge state for a concept is binary: either learned or not.

Although the algorithm supports edge weights, there is no capability within OCTAL for a content author to define the relative weights. As a result, in practice, all edge weights remain the same and the estimation is reduced to Equation 3.

$$P(L_{a_0}) = (P_{max} - P_{min})\frac{\sum\limits_{i \in C} \varphi(i)}{n} + P_{min} \tag{3}$$

Where n is the number of prerequisites to concept a. As an example, given the graph in Figure 12, $P(L_{a_0}) = 0.35$, as shown in Equation 4.

$$P(L_{a_0}) = (P_{max} - P_{min})\frac{2}{3} + P_{min} = 0.35 \tag{4}$$

These probabilities propagate throughout the entire graph with each observation. We estimate learning over the graph using a Markov Chain Monte Carlo (MCMC) model, an algorithm for iteratively sampling a distribution of interdependent random variables [38]. The results of this sampling yield an approximate distribution for the latent knowledge associated with each of the concepts in the graph. The concepts whose estimated probabilities of learning exceed T_L are then presented explicitly to the student on the decorated knowledge graph as described in *Learners* (Section 4.2) and shown in Figure 4. The group of learned concepts grows or shrinks accordingly as information propagates through the graph with additional student responses.

5.7 Learning Tools Interoperability

IMS Global Learning Consortium's Learning Tools Interoperability (LTI) support allows third party tools (such as OCTAL) to more easily integrate with other learning applications, such as the edX platform [9]. OCTAL implements the LTI 1.0 Producer specification so that it might be embedded within an edX course.

There are several limitations, however:

- The edX implementation as an LTI Consumer may not be complete. According to a Piazza LTI document, edX's LTI implementation only functions properly on published courses and not on Studio [22].

- LTI integration requires OCTAL to be opened in its own tab rather than fully integrated as an iframe in a page. OCTAL (specifically, the Django framework) relies on cookies to maintain session state. Modern web browsers block cookies on third-party websites in an iframe, so OCTAL's exercise view, exercise attempt, and knowledge inference requests do not function. Additionally, the current OCTAL server does not support HTTPS, so users will receive security errors if OCTAL is embedded in this manner.

6 Metacognition Study

We believe OCTAL provides learning benefits by improving a learner's metacognitive awareness of their progression through a course's topics. In other words, does seeing an expert-defined concept dependency graph improve a learner's ability to understand which topics they do and do not yet understand? We ran a study[2] on the metacognitive benefits of OCTAL in the spring of 2014 to answer this question.

The study involves pre- and post-surveys on metacognition and participants were divided into two groups: those who saw the concept dependency graph and its full structure, and those for whom the underlying structure was hidden and shown only a list of concepts.

Although OCTAL is intended to be a platform for any content that lends itself to a prerequisite structure, we designed a concept graph and set of exercises geared for UC Berkeley's CS10: The Beauty and Joy of Computing. Specifically, we targeted concepts and ran our protocol for three weeks between the first and second exams, the Quest and Midterm.

The state-of-the-art in evaluating metacognition involves think-aloud studies in which participants vocalize their thought process as they interact with a tool [36]. This was infeasible, however, since we intended our study to be available to the entire class of 238 enrolled students. As a result, we elected to implement pre- and post-surveys that evaluate a participant's metacognitive state before and after using the tool. Although less ideal than think-aloud studies, this form of evaluation is still considered reliable and intercorrelated [32].

We adapted the metacognitive survey from Shaw, et al. to limit the size of the survey and restrict the questions to those most relevant: metacognition regarding learning [32]. The contents of the pre- and post-surveys is available in Appendix A. The questions on the pre-survey and post-survey are identical, except for some limited biographical information questions included on the post-survey.

We visited each section in CS10 to inform students of the study and distribute consent forms. Consenting students received a card with a randomly-generated participant identification number and a URL to access the OCTAL

[2] UC Berkeley Committee for Protection of Human Subjects approval #2014-01-5967.

study. No correlation between the consent form and participant ID number exists.

Upon visiting the OCTAL URL, participants were asked to enter their participant ID and were immediately forwarded to the metacognitive pre-survey. Upon its completion participants were allowed access to the tool. In order to study the metacognitive benefits of displaying the underlying concept structure, half of the students were shown the concept graph in the expected two-dimensional display. This group is referred to as the "graph" group. The other half of the participant IDs, the "linear" group, were not provided the underlying structure and therefore presented only a list of the concepts.

In each case, participants were allowed to select any concept to receive exercises in it. The knowledge inference algorithm determined their level of mastery and highlighted concepts that it estimated participants had mastered.

Upon completion of the study, participants were asked to visit the OCTAL URL where they were then forwarded to the post-survey. The study was terminated once a participant had completed the post-survey. More details on the participant experience is found in Section 4.3.

7 Results

7.1 Metacognition Study

Of 238 enrolled students in CS10: The Beauty and Joy of Computing, 99 agreed to participate in the study and signed the consent form (42%). Of those, 33 visited the site and began the pre-survey. 31 completed the pre-survey and were subsequently returned to OCTAL. Participants were then presented with usage instructions and the set of concepts in either list form or graph form, depending on which group their participant ID was assigned (Section 4.3).

Twenty-six participants went beyond the instructions to click on a concept and view at least one exercise. We consider this group of participants (10.9% of the full body of CS10 students, 26.2% of all participants) to be our user group and show an analysis of their use of the tool below. Only 8 participants completed the post-survey (3.4% of all of CS10, 8.1% of all participants). Precisely half (4) of these participants were in the linear group with the remaining in the graph group. We discuss potential reasons for the drastic reduction in participation below.

For every participant, we compare the responses for each question on the pre- and post-surveys (Appendix A) for any differences in metacognition after using OCTAL. To identify any changes in metacognition between the linear and graphical groups we average the differences among participants in each group and compare those averages (Table 1).

There were no statistically significant changes in metacognition by any comparison ($p \geq 0.09$ for all differences).

Although disappointing, this is perhaps not unexpected with $n = 8$. There are a number of reasons that may have contributed to a decline in participation. The following are several, with some suggestions for improvements to the protocol:

- We were unable to keep track of participant email addresses due to Institutional Review Board (IRB) restrictions on the protocol. Supporting this in the future would allow us to send periodic reminders to use the tool and participate in the study.

Question	$\overline{\Delta}$	p-value	Group Δ	p-value
1	0.125	0.69	-0.5	0.21
2	0	1.00	-0.25	0.36
3	0	1.00	0.25	0.36
4	-0.25	0.65	-0.75	0.52
5	0	1.00	0	1.00
6	0.25	0.62	0	1.00
7	-0.25	0.51	0	1.00
8	-0.125	0.66	0	1.00
9	-0.125	0.75	0.25	0.62
10	0.5	0.23	0	1.00
11	0.625	0.09	-0.25	0.67
12	0.125	0.77	0	1.00
13	0.125	0.69	0	1.00
14	0.25	0.44	0	1.00
15	-0.125	0.55	0	1.00
16	0.125	0.62	0	1.00
17	0.125	0.71	-0.25	0.36
18	0.375	0.12	-0.25	0.54
19	-0.125	0.59	-0.25	0.36
20	0.125	0.74	0	1.00
21	0.25	0.41	-0.25	0.36
22	0.375	0.45	-0.5	0.13
23	0.625	0.20	-0.25	0.54

Table 1: Metacognition pre- and post-survey response data. Average change in responses is shown in $\overline{\Delta}$. Differences in average responses between linear and graphical groups are shown in Group Δ.

- Require participants to complete the post-survey before the midterm exam. Students used the tool as a study aid and, once the exam was over, never returned.

- Embed OCTAL within a curriculum directly. This would provide IRB exemption and is therefore more permissive to reminder emails.

In addition, the use of a think-aloud protocol instead of pre- and post-surveys may yield more conclusive results [36].

7.2 Exercise completion

As part of the data collection process, the OCTAL system stores exercise views and attempts. Participants were required to complete the pre-survey before they could access the tool. Although 31 participants completed this prerequisite, 5 did not use OCTAL beyond viewing the welcome page with instructions. Twenty-six participants clicked on a concept and viewed at least one exercise, and this group serves as our user base for the following analysis.

At the onset of the study, participants were handed a randomly-selected card with a pre-generated ID. Each set of IDs had been evenly assigned to either the linear or graphical group prior to random distribution. Since group was assigned prior to distribution, and not at the system's first interaction with the user, the set of users is not split evenly between the two groups. Fifteen (57.7%) of the users were in the linear group while 11 (42.3%) were in the graph group.

Of the user base, 23 interacted with the tool to complete one exercise (88.4%). All 15 participants in the linear group completed at least one exercise while only 8 of the graphical group did the same. Three participants (all from the linear group) completed all exercises. Thirteen participants (11 linear, 2 graphical) completed at least half of the exercises. Table 4 in Appendix B provides more detailed usage data for all users.

The linear group's increased participation is marginally statistically significant over the graph group ($p = 0.01$, Mann-Whitney U). Our protocol prevented us from interviewing participants, so understanding the factors behind this result are based on speculation. For instance, participants may have found the graph view confusing or unfamiliar, but this would require additional usability studies to resolve.

7.3 Sessions

We also wanted to determine if these users returned to the tool by looking at participant's sessions. The data only contains timestamps for user actions, so it is not possible to know if a user has left the page and returned. Average differences between timestamps do not provide any useful data to determine reasonable session lengths since timestamp differences follow a power law distribution: median time between activity for a user is 18 seconds but an

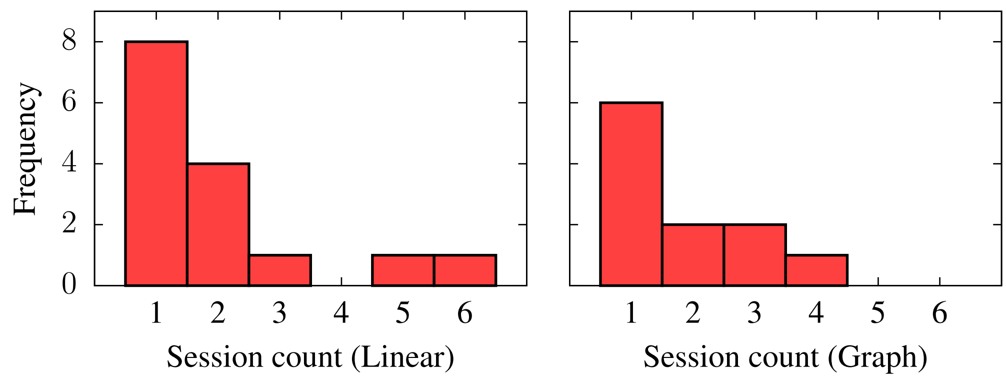

Figure 13: Histogram of sessions per user by group.

average of 10,019 seconds (2 hours, 46 minutes, 59 seconds). We therefore apply an arbitrary, but conservative, definition of 60 minutes to define a session. By choosing this length, we estimate that a reasonable user will not spend longer than an hour to solve and attempt an exercise without first leaving and returning.

With this definition of a session, we find the 26 users had a total of 50 sessions; approximately 1.9 per user. Figure 13 shows a histogram of the number of sessions participants initiated in each group. Participants in the linear group visited in 30 total sessions; 2 sessions per user on average. Meanwhile, the graph group participants visited 20 times for a ratio of 1.8 sessions per user. This is not a statistically significant difference ($p = 0.99$, Mann-Whitney U) to determine which group was more likely to return.

To investigate any potential differences in user activity, we divide each session into one of two types: "view" sessions, in which a participant viewed (but did not attempt) one or more exercises, or "attempt" sessions, in which a participant viewed and attempted one or more exercises. By this metric,

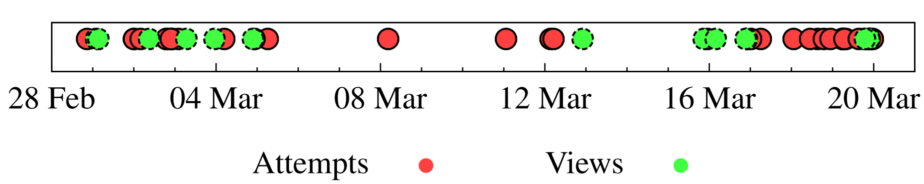

Figure 14: Time line of session initiations differentiated by session type.

Group	User	Path Score	Residual
Linear	**1**	**−0.234**	**1.195**
	2	**0.533**	**0.571**
	3	**−1**	**0**
	5	1	0
	6	−0.136	1.080
	7	1	0
	8	0.68	0.535
	9	0.944	0.334
	10	0.857	0.023
	11	0.933	0.272
	12	0.4	0
	13	0.4	0
Graph	**16**	**−0.755**	**0.310**
	17	**−0.5**	**0**
	20	−0.369	1.099
	21	0.057	0.131

Table 2: Path scores for users with attempts in more than one concept. Bolded rows represent participants that completed the post-survey. User numbers match with that in Appendix B, Table 4, but not all had sufficient data to be represented here.

37 sessions (74% of all) contained at least one attempt with the linear group representing 24 of those (80%) and the graphical group 13 (65%).

This indicates that users did not attempt any questions in 26% of their sessions. These users, however, sometimes did return to attempt exercises or view others: of the 9 participants with view sessions, 5 returned for another session of any type.

A time line of the sessions for the duration of the study is shown in Figure 14. Interestingly, but perhaps unsurprisingly, most sessions are centered around the release of the tool and the time just before CS10's midterm exam.

7.4 Path Scoring

We created a score to describe every user's path through concepts in order to investigate the differences between the group's navigation. Our hypothesis

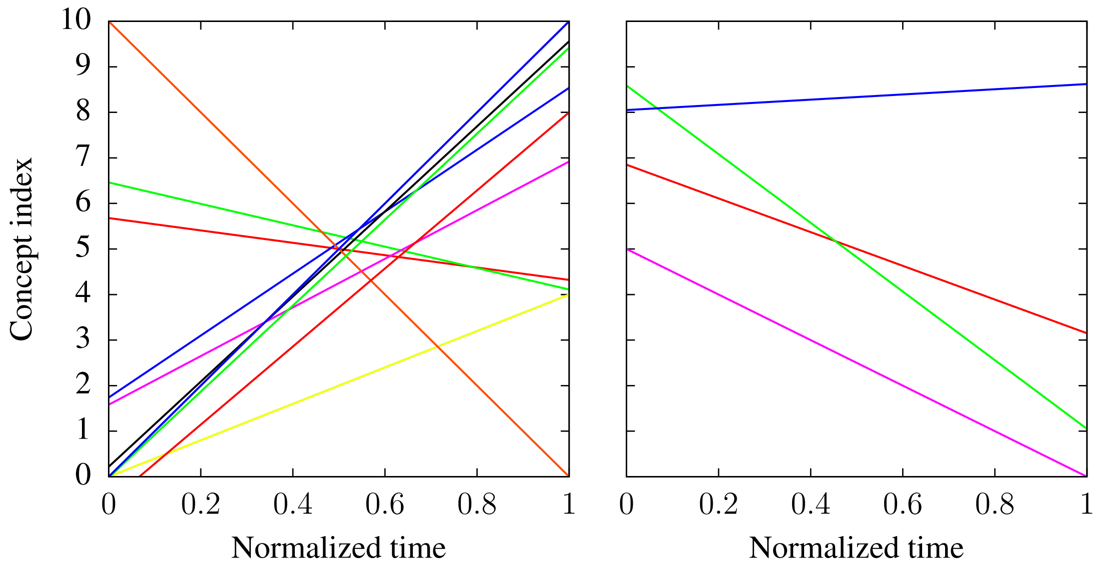

Figure 15: Best linear fit for normalized paths for all participants in the linear (left) and graphical (right) groups. Concept index matches that shown in Table 3.

is that users presented with a linear view are more likely to navigate it in a top-down or bottom-up fashion, while those in the graph group are more likely to choose concepts based on learning need.

Index	Concept name
0	Variables
1	Variable Mutation
2	Conditionals
3	Loops
4	Lists
5	Functions
6	Tree Recursion
7	Tail Recursion
8	Algorithmic Complexity
9	Fractals
10	Concurrency

Table 3: Concept names by index.

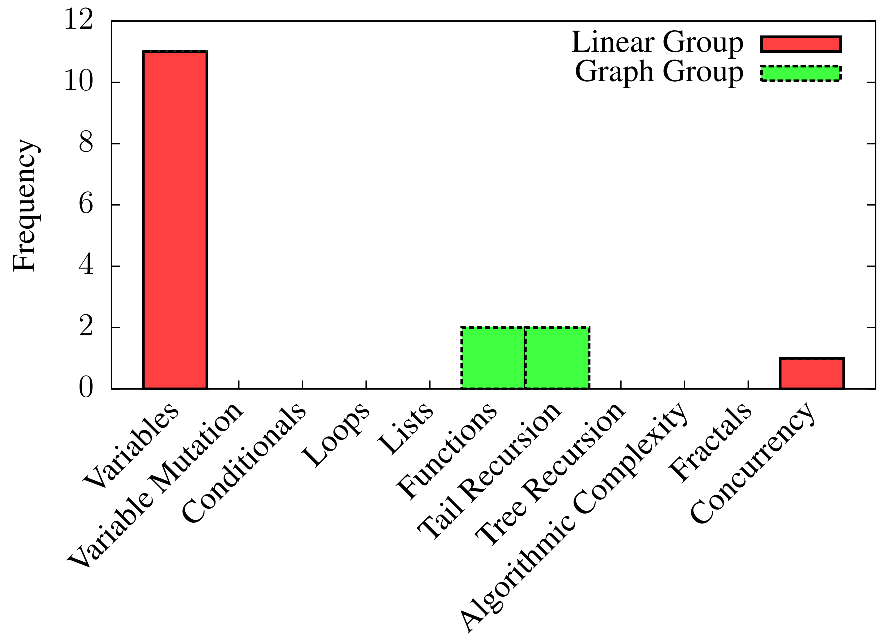

Figure 16: Histogram of starting concepts by group.

Path scoring requires that users attempted at least one exercise in at least two different concepts. This restricts the user base further to 16 total users; 12 from the linear group and 4 from the graphical. The statistically significant drop in users in the graph group may impact the following results; these participants may have been more motivated to use the tool for its intended learning purpose. However, it is still interesting to note the differences in navigation among the groups.

First, we create a reference list of concepts sorted in increasing order of distance from the root (Table 3). This is the same order as the list presented to linear users. Each concept is assigned an evenly-spaced value in the range $[0, 1]$ based on its index in this list. This creates, in essence, a reference path that we use to compare participants' paths.

To generate a score for each participant, we first remove "view" events from a participant's activity stream. We therefore only consider exercise attempts since this is a more reliable indicator that a participant is giving thought to the concept and not haphazardly clicking. For a user u, the stream of events E_u is sorted by time and reduced such that a concept visited at

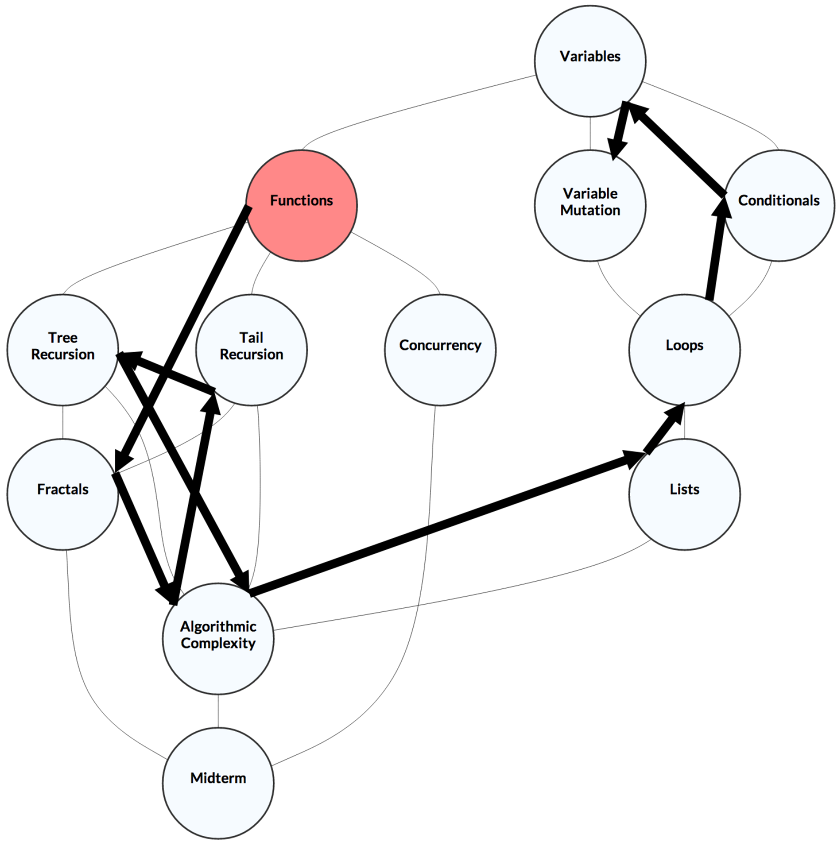

Figure 17: The path of user 16 from the graph group through the concept graph. The path begins from the red-highlighted Functions node.

attempt E_{ua} is unique against adjacent attempts: $E_{u(a-1)} \neq E_{ua} \neq E_{u(a+1)}$. Finally, the stream of events E_u is normalized over time such that all values of a are evenly spaced within the range $[0, 1]$.

This results in a normalized path for participants across time and concepts. Each path ignores duration of activity and quantity of attempts and represents movement through concepts. Graphically, the paths are normalized to fit within the unit square bounded by $(0, 0)$ to $(1, 1)$. Figures 18 and 19 in Appendix C show normalized paths before scoring for each group.

We apply a least-squares linear fit for every user u in the event stream E and obtain a slope that represents the path score for that participant (Figure 15). The least-squares residual value indicates the difference in actual values compared to predicted, and is used to determine the score's error (Table 2).

The error for participants that moved in extreme hops to distant concepts can be significant. User 1, for instance (Table 2), began at a middle concept, continued to all post-requisites, and finally "wrapped around" the list of concepts to finish the complete set of exercises from the beginning.

A score of 1 indicates that user's path through the concepts was identical to the reference. In other words, users 5 and 7 attempted exercises precisely in the order presented to them with no deviation. User 3 with score -1 attempted exercises in precisely the opposite order: working up from the bottom of the list. Interestingly, this same user attempted exercises in all concepts, but simply did them in reverse order.

Predictably, users in the graph group did not follow an order analogous to the majority of the linear group. In fact, whereas all but one of the linear group began at the first concept (Variables), all participants in the graph group began at a middle concept and trended towards prerequisite nodes (Figure 16). Through a qualitative look at their paths, those users appear to navigate by clusters of nodes. By way of example, the path for user 16, a participant in the graph group who completed the post-survey, is shown in Figure 17.

The paths of two other graph participants were similarly clustered through concepts. The fourth graph participant attempted a question in Functions followed by an exercise in Variables and never returned. Although a statistically significant number of graph participants withdrew from the tool after completing the pre-survey, it is interesting and encouraging to see that those that did remain followed unique paths around the graph (Figure 19 in Appendix C).

8 Future Work

The implementation may benefit from further integration into LMS and MOOC platforms. To reduce effort for content authors, for instance, OCTAL could evolve into a tool that relies mostly on graph display and knowledge estimation while relying on third party platforms to supply exercises.

Additional work must be done in order to identify the reasons for the marginally statistically significant drop in participation by users presented with the graph view. This future work could also identify which UI principles, if any exist, might be best for tools based on concept maps. Participants in the graph group that persisted to use the tool also showed that they might apply more thought to where to begin their interaction with the exercises, a result that requires more study to reduce confounding issues of self-selection.

Modifying future experiments to include usability studies and think-aloud metacognition studies would provide additional insight into student's behavior with the tool.

The knowledge estimation algorithm contains ample opportunity for future work. Most prominently, it allows for edge weights to define the relative importance of prerequisites for a concept (Section 5.6). The graph's implementation in OCTAL, however, does not currently support edge weights and so all prerequisites are given equal weighting. Additionally, the initial parameters have been defined based on best-effort estimate and are not properly trained. A study to collect this data and train the parameters may offer significant improvements in knowledge estimation performance.

There are also several assumptions which underlie the current algorithm, any one of which offers an opportunity for future research. First, we assume that knowledge structure we provided in the study (Section 6), is a sane and useful representation of basic computer science concepts. Second, we make the strong assumption, shared with BKT, that a student's knowledge of a given concept is binary, as well as generalizations about partial knowledge. Third, the algorithm ignores any impact that time might have on the modeling of student knowledge, either in the sense of providing an additional source of information about knowledge acquisition, or even in imposing a meaningful ordering on observed responses to assessment items.

Traditionally, Bayesian Knowledge Tracing and the algorithms which derive from it model student knowledge as a binary indicator of mastery. Though difficult to capture in the computation of a posterior probability of total mastery, it may be more realistic to model partial knowledge of a subject.

9 Conclusions

OCTAL was built as a tool to allow learners the opportunity to explore a course's underlying prerequisite structure of topics. Learners are presented with a concept graph and may attempt exercises in any topic they select. A knowledge estimation algorithm infers their level of mastery of concepts based on their performance on the exercises. Although the knowledge estimation and graph structure provide metacognitive hints to the user about how they might choose to engage with the material, they are allowed free navigation throughout the tool.

OCTAL resides at the intersection of a number of fields; education technology and research, concept maps, and latent knowledge estimation. Each field contains a large body of existing work that has informed OCTAL's design. It is our hope that by building a tool that combines these fields we can provide learners with new means of studying and interacting with course material.

A study to gauge improvements in metacognition revealed no statistically significant results. One reason might be due to very low participation with $n = 8$. We suggest future protocols consider think-aloud studies and more frequent reminders to participants to strive for meaningful results and increased participation.

OCTAL may remain a viable technology to provide users with additional learning opportunities, although additional work must be done to understand the drop in participation by users presented with a concept graph versus those presented with a list. We observed fascinating and encouraging trends in the navigation of the concept graph from those participants that did continue to use the tool. Predictably, learners presented with a list of concepts mostly proceeded through the list in order. The participants that viewed the concept graph, however, tended to start at a concept near the middle and navigated by clusters of concepts.

We have worked to create a tool that is also usable outside of the context of research. With authoring tools and the capability to embed into popular online learning platforms, we hope that OCTAL will continue to be refined and used by researchers and instructors alike.

10 Acknowledgments

This research was carried out in part by support from the Chamberlins towards research in online education tools.

Zachary MacHardy has provided considerable assistance on nearly all aspects of the project. His help has been invaluable.

Colorado Reed contributed his support on the Metacademy and `kmap` codebases and guidance in discussions on knowledge estimation and modeling.

Kristin Stephens-Martinez and the students of the Berkeley Institute of Design (BiD) have provided fantastic feedback throughout the process.

Professors Armando Fox and John Canny and the students of CS294-94 (Autograding for Online Education) provided feedback and suggestions on early versions of this project.

Advising from both Senior Lecturer SOE Dan Garcia and Professor Armando Fox have provided irreplaceable direction and discussion that have shaped the project but also our approach to thinking about online education.

Thank you to all.

References

[1] Almond, R. G., and Mislevy, R. J. Graphical models and computerized adaptive testing. *Applied Psychological Measurement 23*, 3 (Sept. 1999), 223–237.

[2] Anderson, J. *Cognitive Psychology and Its Implications.* Worth Publishers, 2005.

[3] Anderson, T. *The Theory and Practice of Online Learning.* AU Press, 2008.

[4] Baker, R. S. J. d., Corbett, A. T., and Aleven, V. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Intelligent Tutoring Systems*, B. P. Woolf, E. Ameur, R. Nkambou, and S. Lajoie, Eds., no. 5091 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, Jan. 2008, 406–415.

[5] Bloom, B. S. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher 13*, 6 (June 1984), 4–16.

[6] Chang, K., Sung, Y., and Chen, S. Learning through computer-based concept mapping with scaffolding aid. *Journal of Computer Assisted Learning 17*, 1 (2001), 21–33.

[7] Coburn, C. Rethinking scale: Moving beyond numbers to deep and lasting change. *Educational Researcher 32*, 6 (Aug. 2003), 3–12.

[8] Collins, J. A., Greer, J. E., and Huang, S. X. Adaptive assessment using granularity hierarchies and bayesian nets. In *Intelligent Tutoring Systems* (1996), 569577.

[9] Consortium, I. G. L. Learning Tools Interoperability. `http://www.imsglobal.org/toolsinteroperability2.cfm`, 2014.

[10] Corbett, A. T., and Anderson, J. R. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction 4*, 4 (Dec. 1994), 253–278.

[11] Duffy, T. M., Lowyck, J., and Jonassen, D. H. *Designing Environments for Constructive Learning*, 1st ed. Springer Publishing Company, Incorporated, 2012.

[12] Eisner, J., Kornbluh, M., Woodhull, G., Buse, R., Huang, S., Michael, C., and Shafer, G. Visual navigation through large directed graphs and hypergraphs. *Proc. of IEEE InfoVis, Poster/Demo Session 116* (2006).

[13] Falmagne, J.-C., Cosyn, E., Doignon, J.-P., and Thiry, N. The assessment of knowledge, in theory and in practice. In *Formal Concept Analysis*, R. Missaoui and J. Schmidt, Eds., no. 3874 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, Jan. 2006, 61–79.

[14] Gong, Y., Beck, J. E., and Heffernan, N. T. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *Intelligent Tutoring Systems*, V. Aleven, J. Kay, and J. Mostow, Eds., no. 6094 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, Jan. 2010, 35–44.

[15] Google. Instructions to host images on Google Drive. `https://support.google.com/richmedia/answer/4351301?hl=en`, 2014.

[16] Hadwin, A. F., and Winne, P. H. Conotes2: A software tool for promoting self-regulation. *Educational Research and Evaluation 7*, 2-3 (2001), 313–334.

[17] Hambleton, R. K., Swaminathan, H., and Rogers, H. J. *Fundamentals of Item Response Theory*. SAGE, July 1991.

[18] Hoffman, B., and Spatariu, A. The influence of self-efficacy and metacognitive prompting on math problem-solving efficiency. *Contemporary Educational Psychology 33*, 4 (Oct. 2008), 875–893.

[19] Hübscher, R., and Puntambekar, S. Integrating knowledge gained from data mining with pedagogical knowledge. In *EDM* (2008), 97–106.

[20] Novak, J. D., and Caas, A. J. The theory underlying concept maps and how to construct them. *Florida Institute for Human and Machine Cognition 1* (2006).

[21] Palincsar, A. S. Keeping the metaphor of scaffolding fresh-a response to c. addison stone's the metaphor of scaffolding: Its utility for the field of learning disabilities. *Journal of Learning Disabilities 31*, 4 (1998), 370–373.

[22] Piazza. Adding Piazza to edX. `https://s3.amazonaws.com/piazza-materials/LTI-Setup/AddingPiazzatoedX.pdf`, 2014.

[23] Puntambekar, S. Concept Mapped Project-based Activity Scaffolding System (CoMPASS) Project. `http://www.compassproject.net/info/software.html`, 2014.

[24] Puntambekar, S., and Goldstein, J. Effect of visual representation of the conceptual structure of the domain on science learning and navigation in a hypertext environment. *Journal of Educational Multimedia and Hypermedia 16*, 4 (October 2007), 429–459.

[25] Puntambekar, S., Stylianou, A., and Goldstein, J. Comparing classroom enactments of an inquiry curriculum: Lessons learned from two teachers. *Journal of the Learning Sciences 16*, 1 (2007), 81–130.

[26] Reckase, M. *Multidimensional Item Response Theory*. Springer, July 2009.

[27] Reed, C. kmap Library. `https://github.com/cjrd/kmap`, 2014.

[28] Reed, C., Grosse, R., , and the Metacademy development team. Metacademy (GitHub). `https://github.com/metacademy/`, 2014.

[29] Reed, C., and Grosse, R. Metacademy. `http://metacademy.org/`, 2014.

[30] Romero, C., and Ventura, S. Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 40*, 6 (2010), 601–618.

[31] Sarkar, M., and Brown, M. H. Graphical fisheye views. *Commun. ACM 37*, 12 (Dec. 1994), 73–83.

[32] Schraw, G., and Dennison, R. S. Assessing metacognitive awareness. *Contemporary Educational Psychology 19*, 4 (1994), 460 – 475.

[33] Segall, D. O. Computerized adaptive testing. *Encyclopedia of Social Measurement. Amsterdam: Elsevier* (2005).

[34] Sharma, P., and Hannafin, M. J. Scaffolding in technology-enhanced learning environments. *Interactive Learning Environments 15*, 1 (2007), 27–46.

[35] Stone, C. A. The metaphor of scaffolding: Its utility for the field of learning disabilities. *Journal of Learning Disabilities 31*, 4 (1998), 344–364.

[36] Veenman, M. V., Bavelaar, L., Wolf, L. D., and Haaren, M. G. V. The on-line assessment of metacognitive skills in a computerized learning environment. *Learning and Individual Differences 29*, 0 (2014), 123 – 130.

[37] Vygotsky, L. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, 1978.

[38] Walsh, B. Markov chain monte carlo and gibbs sampling.

[39] Wood, D., Bruner, J. S., and Ross, G. The role of tutoring in problem solving*. *Journal of Child Psychology and Psychiatry 17*, 2 (1976), 89–100.

Appendices

A Metacognition Experiment Surveys

The following survey was given to participants in the metacognition experiment (Section 6). Each question asked participants to respond on a Likert scale: Strongly Disagree, Disagree, Neutral, Agree, or Strongly Agree.

Please select your level of agreement for each of the following statements.

1. I ask myself periodically if I understand the material.
2. I consider alternative solutions before I answer a problem.
3. I employ strategies that have worked for me in the past.
4. With respect to the material, I understand my strengths and weaknesses.
5. I think about what I need to know before beginning a problem.
6. I know what is important for the understanding of each problem.
7. I learn best when I already know something about the topic.
8. I am good at remembering information.
9. I ask myself if there was an easier way to do things after finishing a task.
10. I periodically review to help me understand important relationships.
11. I ask myself questions about the material before beginning an assessment.
12. I think of several ways to solve a problem and choose the best one.
13. I am aware of what strategies I use when I study.
14. I am a good judge of how well I understand something.
15. I find myself using helpful learning strategies automatically.
16. I pause frequently to check my understanding.
17. I know when each strategy I could use would be most effective.
18. When I'm having trouble, I change which strategy I employ.
19. I stop and return to information which is unclear.
20. I reevaluate my assumptions when I am confused.
21. I ask myself how well I am doing when I learn something new.
22. I ask myself how what I am learning is related to what I already know.
23. I ask myself if I learned as much as I could have once I finished the task.

B OCTAL Usage Data

Group	User	Exercise Views All	Exercise Views Unique	Exercise Views Ratio	Exercise Attempts All	Exercise Attempts Unique	Exercise Attempts Ratio	Sessions
Linear	**1**	**41**	**24**	**1.71**	**34**	**24**	**1.42**	**2**
	2	**41**	**23**	**1.78**	**31**	**19**	**1.63**	**2**
	3	**15**	**13**	**1.15**	**13**	**11**	**1.18**	**1**
	4	**7**	**7**	**1.00**	**1**	**1**	**1.00**	**1**
	5	46	24	1.92	41	24	1.71	1
	6	50	24	2.08	42	24	1.75	1
	7	44	23	1.91	31	22	1.41	2
	8	64	22	2.91	54	22	2.45	6
	9	46	21	2.19	31	18	1.72	2
	10	43	21	2.05	27	14	1.93	5
	11	45	20	2.25	36	18	2.00	1
	12	25	14	1.79	19	12	1.58	1
	13	34	14	2.43	24	14	1.71	3
	14	7	5	1.40	4	4	1.00	1
	15	2	2	1.00	1	1	1.00	1
Graph	**16**	**51**	**24**	**2.13**	**38**	**22**	**1.73**	**4**
	17	**19**	**7**	**2.71**	**15**	**4**	**3.75**	**3**
	18	**8**	**4**	**2.00**	**6**	**4**	**1.50**	**1**
	19	**1**	**1**	**1.00**	**0**	**0**	**0.00**	**1**
	20	46	24	1.92	33	21	1.57	1
	21	21	8	2.63	16	7	2.29	2
	22	6	4	1.50	5	4	1.25	1
	23	4	3	1.33	1	1	1.00	3
	24	4	3	1.33	1	1	1.00	2
	25	2	2	1.00	0	0	0.00	1
	26	1	1	1.00	0	0	0.00	1

Table 4: OCTAL participant usage data, split into groups. Bolded rows are those participants that completed the post-survey. A session contains activity that takes place within an hour of other activity.

C OCTAL Participant Paths

C.1 Linear Group

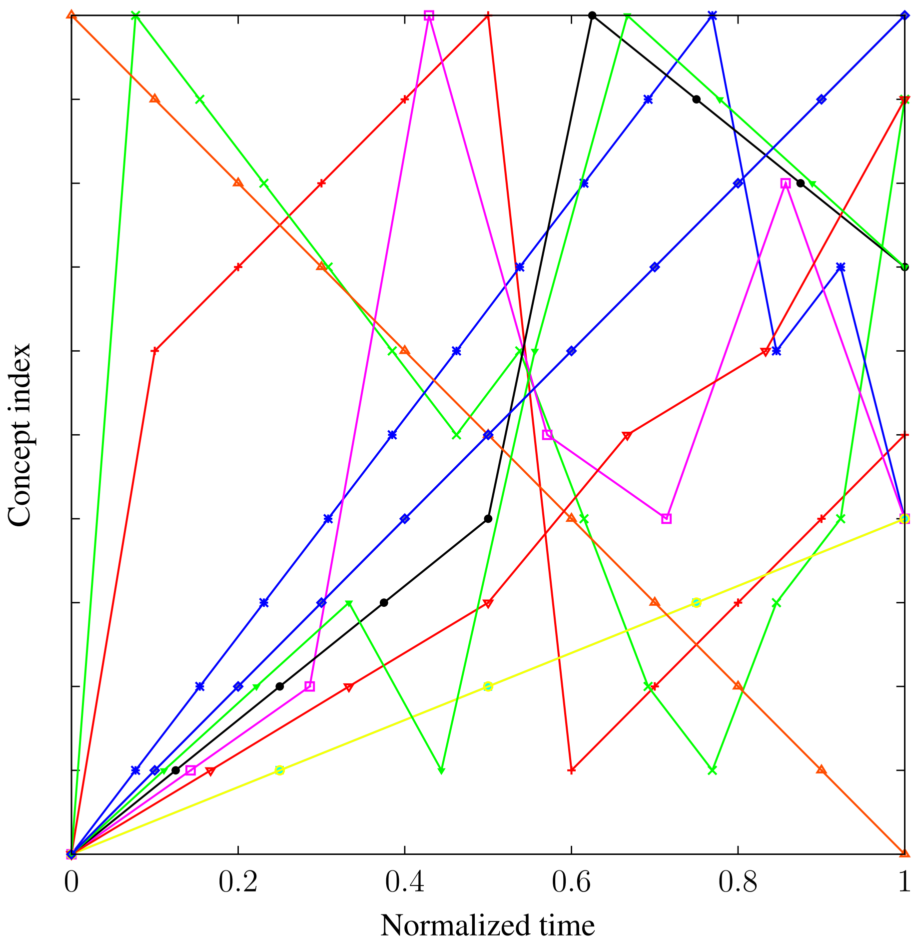

Figure 18: Normalized paths for all participants in the linear group. All but one participant in the linear group began at the first concept, and most progressed linearly through the list. Concept index matches that shown in Table 3.

C.2 Graphical Group

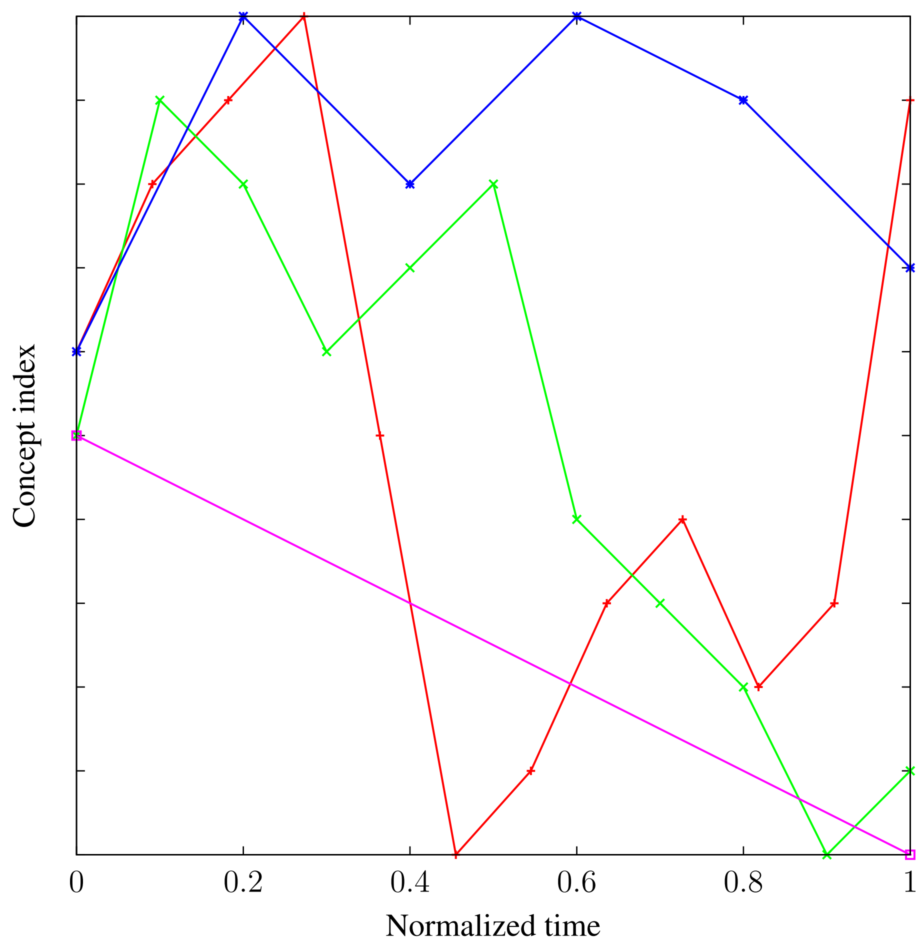

Figure 19: Normalized paths for all participants in the graphical group. All participants in the began at a middle concept and navigated by cluster. Concept index matches that shown in Table 3.

D Recommendations for Content Authors

D.1 Concept Graph

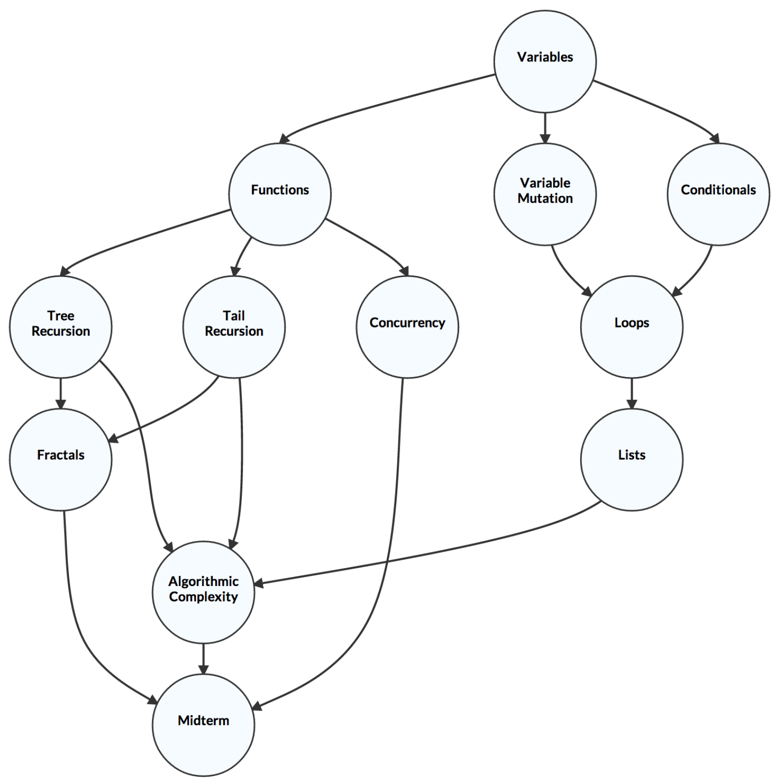

Figure 20: Concept map for concepts from the first exam (Quest) to second (Midterm) from CS10: The Beauty and Joy of Computing.

An OCTAL unit concept graph is a directed acyclic graph of concepts and could be designed by an instructor, teacher, member of course staff, or some other expert in the field, or perhaps designed by a learner as part of an exercise. There are no imposed technical limits on the graph design other than the structural limitation that the graph must contain no orphaned nodes and no cycles. Despite this, we intend graphs to contain a relatively coarse granularity of concepts for the following reasons:

- Students may be overwhelmed by a complex graph with extremely fine granularity.
- It is non-trivial to design a navigable concept graph with variable granularity or zoom levels.
- Coarse granularity simplifies exercise assignment to concepts.
- Reduction in the complexity of hand-building the hierarchy of nodes.

As part of the research study (Section 6) we implemented a concept graph from the first exam (the Quest) to the second (the Midterm) from CS10: The Beauty and Joy of Computing. This content represents approximately one third of the material in the course. We found that creating a graph for this quantity of topics in the course allowed the graph to have a pleasing and manageable number of nodes. However, we did not perform a study to determine the ideal quantity of nodes in the graph and we leave this investigation to future work or to the larger field of concept map research.

The graph in Figure 20 contains a final leaf (the node labeled "Midterm") that is present only because of a technical limitation in the version of OCTAL that was used for the study. In that version, the Midterm concept was a special case that was selected by default that provided a participant with text that introduced the graph and explained the usage of the tool. The current version of OCTAL does not have this limitation and will display the introductory text without the need for such a node.

We iterated the graph after presenting it to subject matter experts and recommend a similar course of action for content authors. However, we suggest that designers also consider iterating the graph with feedback from learners.

D.2 Exercises

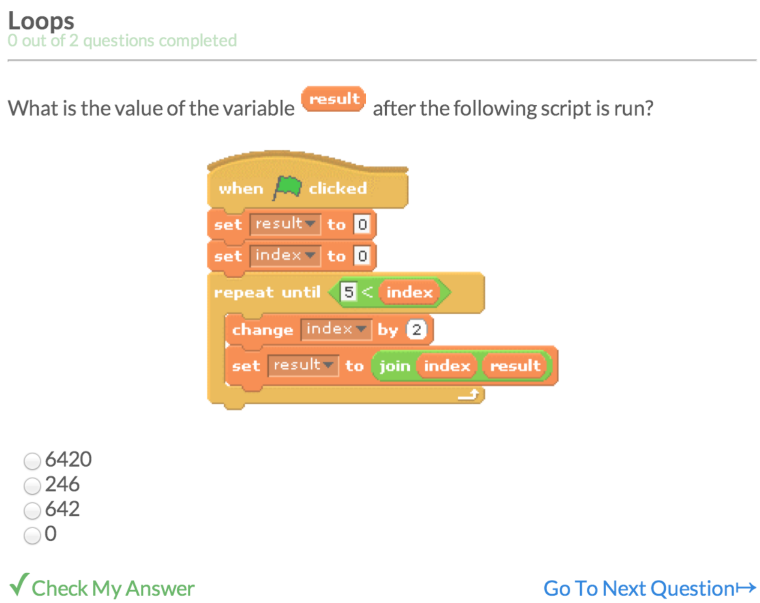

Figure 21: Sample exercise in the "Loops" concept category.

OCTAL leverages an expert-derived bank of exercises tagged with relationships to nodes in the concept graph. This association enables knowledge estimation based on a value of correctness for each exercise in a concept (Section 5.6).

Since the knowledge estimation algorithm depends on the correctness of the exercise and not the value itself, the tool theoretically can support questions in any arbitrary format including multiple choice, short answer, and so on. However, the current implementation restricts exercises to multiple choice.

We suggest at least five questions per concept in order to maximize the tool's utility for a learner.

Exercise text supports HTML to enable image embedding. OCTAL does not, however, support uploading images to the system, so the images must be inline linked (hotlinked) from another source. One freely available option is Google Drive [15]. In this option, creating a Google Drive folder, making it public, and placing images inside allows those images to be hosted on Google and referenced in OCTAL. This is how we embedded images for our study (Figure 21).

D.3 LTI Integration

To use LTI integration, a content author must simply perform the following steps:

1. When creating an OCTAL unit, record the LTI consumer key and shared secret presented at the bottom of the page.
2. In edX Studio for a course, select "Advanced Settings" under the Settings tab, and modify the following settings:
    - advanced_settings should be ["lti"]
    - lti_passports should be ["<id>:<key>:<secret>"], where <id> is an LTI passport ID (octal-lti would be fine), <key> is the consumer key, and <secret> is the shared secret (Step 1).
3. Go to the content page, and create a Section and Subsection where you would like the integration to take place, if none exists yet.
4. Edit the subsection and add a "New Unit".
5. Under "Add New Component", select "Advanced", and then click "LTI".
6. Add the OCTAL LTI URL under launch_url. This is in the form http://octal.danallan.net/maps/N/lti where N is the OCTAL unit number.
7. Add the LTI passport ID (from Step 2; in this case, "octal-lti") to lti_id.
8. Ensure open_in_a_new_page is True.
9. Click Save.

E Obtaining the Source Code

OCTAL is open-source and freely available to download and edit. The repository is available on GitHub, with the following link representing the final version as described in this text:
`https://github.com/danallan/octal-application/tree/v2.0-ms`

The version used in the metacognition study (Section 6) is available here:
`https://github.com/danallan/octal-application/tree/v1.1-pilot+postsurvey`