
Course Builder Skill Maps

Boris Roussev

Google, Inc.
1600 Amphitheater Parkway
Mountain View, CA 94043 USA
borislavr@google.com

Pavel Simakov

Google, Inc.
1600 Amphitheater Parkway
Mountain View, CA 94043 USA
psimakov@google.com

John Orr

Google, Inc.
1600 Amphitheater Parkway
Mountain View, CA 94043 USA
jorr@google.com

Amit Deutsch

Google, Inc.
1600 Amphitheater Parkway
Mountain View, CA 94043 USA
amitdeutsch@google.com

John Cox

Google, Inc.
1600 Amphitheater Parkway
Mountain View, CA 94043 USA
johncox@google.com

Michael Lenaghan

Google, Inc.
1600 Amphitheater Parkway
Mountain View, CA 94043 USA
mlenaghan@google.com

Mike Gainer

Google, Inc.
1600 Amphitheater Parkway
Mountain View, CA 94043 USA
mgainer@google.com

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the owner/author(s).
L@S 2016, April 25-26, 2016, Edinburgh, Scotland UK
ACM 978-1-4503-3726-7/16/04.
<http://dx.doi.org/10.1145/2876034.2893374>

Abstract

In this paper, we present a new set of features introduced in Course Builder that allow instructors to add skill maps to their courses. We show how skill maps can be used to provide up-to-date and actionable information on students' learning behavior and performance.

Author Keywords

Skill Maps; Adaptive Learning; Learning Analytics; MOOCs

Introduction

On-line learners need up-to-date information on their progress, areas of strength and weakness, how to adapt their learning behavior, and personalized paths through the material they are learning [7]. Instead, they often get static information presented to them, with interactivity and personalization limited to scores on graded, summative assessments. Instructor visibility into student performance is often limited to averages and distributions of class scores. Worse, gathering this data usually happens after, rather than during, student engagement with a unit of content [5].

To address these shortcomings, online learning platforms employ learning analytics [5]. To implement these analytics for specific pieces of content, course designers need to define the knowledge domain of their course, and understand student learning states.

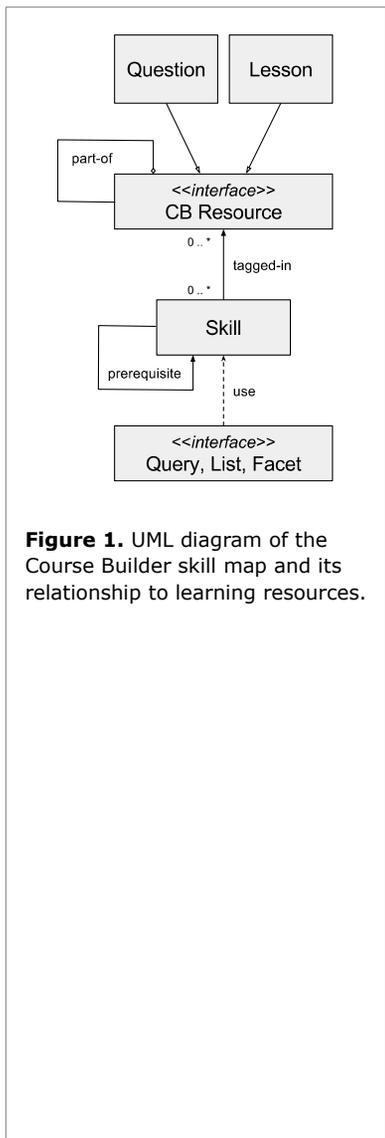


Figure 1. UML diagram of the Course Builder skill map and its relationship to learning resources.

Course Builder (CB) is an open-source platform developed by Google for delivering online courses at scale. Over the past four years, Google, nonprofit organizations and universities have used CB to create hundreds of online courses in multiple languages reaching over 2 million students worldwide. CB version 1.10 includes new features that allow instructors to add skill maps to their courses. This paper describes the approach taken in CB and compares related approaches.

Course Builder Skill Map Model

In the CB model, a skill is any unit of knowledge in the course which can be taught or assessed. In CB lessons contain course material such as text, videos, and activities. Lessons within a course are usually presented linearly. Now each lesson can also be tagged with the *skills* taught. A skill may depend on several previous skills taught elsewhere in the course, and it may have zero or more follow-on skills that depend on it. Navigating the course by skill, rather than lesson, allows students to traverse the course nonlinearly. For example, students interested in just one skill may view only the subset of lessons that teach that skill.

In addition to lessons, instructors are able to tag questions and other learning resources with skills as illustrated by the Unified Modeling Language diagram in Figure 1. The *Query, List, Facet* interface and the *use* association model a set of classes implementing functionality for querying, grouping, aggregating, and faceting the skill map. The recursive *part-of* association on Resource defines the course resource graph through aggregation and transclusion.

Course Builder Skill Map Features

We have extended the instructor’s view of the course outline to show the skills taught in all lessons. We have also added a skill map dashboard with a tabular and graphical view. On the tabular view the instructor can see all the skills, their prerequisites, follow-ons, and what lessons they are taught in. In the graphical view the instructor can see the web of dependencies between skills. Here, the instructor can focus in on a particular skill or rearrange the layout of the graph to understand better the intricate relationships of the skills.

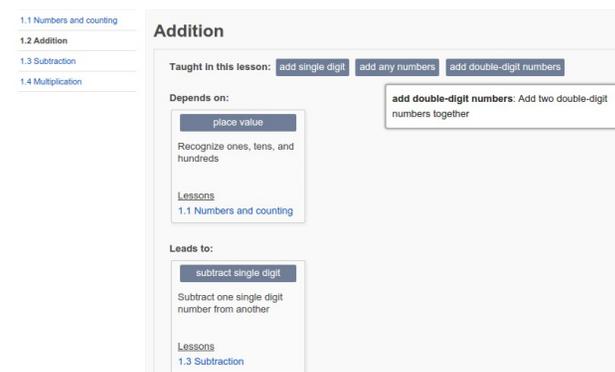


Figure 2. Student's perspective of the lesson view

The student’s perspective of the lesson view is shown on Figure 2. It has a collapsible knowledge panel listing the skills taught in a particular lesson. The knowledge panel shows details about the skills in a lesson, including name, description, and level of mastery; it also shows the lesson’s prerequisite skills and the skills that follow-on from it. Each skill on the panel links to the content unit where it is taught, so students who want to find their own pathways through the course

For the instructor, the skill map opens up interesting new analytics information. We have added two new views to the analytics dashboard. The first displays the student proficiency distributions for all skills in a unit, see Figure 3. The second draws a chart that compares student progress on selected skills over the lifetime of a course.

From a pedagogical point of view, skill maps enable a goal-oriented course design approach: course creators can start with the skills they want to teach and design the course content around them.

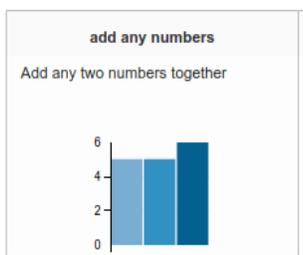


Figure 3. Analytics dashboard for skill competencies

and explore all the different links can use the panel to personalize their learning experience [6].

In addition to displaying information about the skills taught in a lesson, the knowledge panel provides a meta-cognitive signal, which may help students get deeper understanding of the content and explore the course in a way that meets better their personal needs.

Comparison with Two Related Systems

The Open Learning Initiative (OLI), which began at Carnegie Mellon University in 2002, creates web-based courses designed so students can learn effectively without an instructor. OLI uses a two-tiered model: high-level objectives which specify what students will be able to do at the end of each instructional module, backed by one or more skills representing a discrete concept or knowledge component [2]. Interactive activities and quizzes are used to assess students' learning of the various skills. OLI uses Bayesian Knowledge Tracing (BKT) to estimate the probability a student has learned each skill [1] and mastery of an objective is calculated by averaging the scores of the skills that make up the objective.

Unlike the OLI model, the CB skill map is single-tiered, with the addition of the prerequisite relation. The OLI model uses objectives in order to roll up aggregate data for reporting, since skill maps tend to be large and the fine-grained data of competency on a skill-by-skill basis is often not useful. However the OLI model encodes one particular set of roll-ups in the many-to-many mapping of skills to objectives. In our model, we recognize the need for rollups but leave open the possibility of aggregating along multiple different axes, e.g. by course section (that is, unit or lesson); by terminal skill

(that is, skills which sit at the head of chains of dependencies); or by other categories. This flexibility is reflected in the Query/List/Facet interfaces in Figure 1 above.

The OLI model also does not include any prerequisite relation between skills. In CB, prerequisite relations make it possible to make inferences about the student learning state. If we conclude that a student knows a particular skill, we can reason probabilistically that the student also knows all its prerequisite skills. We can apply the same process of reasoning going from prerequisites to follow-on skills. This can be used in efficient adaptive testing.

Learning Space Theory (LST) is used as the theoretical basis for ALEKS. In LST, the basic unit of knowledge is a parameterized question [3] and the student knowledge state is the set of all questions in the domain that the student is capable of solving. Not all knowledge states are feasible; a student who cannot add single-digit numbers likely cannot add double-digit numbers either. The collection of all feasible states satisfying certain regularity conditions is called a *learning space*. At least initially, a learning space is designed using human experts. The time of these experts is expensive, so a major issue is state explosion. A learning space with n questions theoretically has 2^n states although in practice there are much fewer. Pruning the space is done with a series of entailment queries of the form "Do you believe a student who has failed questions $\{q_1 \dots q_n\}$ would also fail question q_{n+1} ?" [3] and the number of queries required can easily grow to the thousands.

Unlike LST, CB explicitly defines the structure of the knowledge domain using the prerequisite relation. Our deterministic skill map model cannot express alternate sets of prerequisites, whereas in LST there may be more than one way to achieve mastery on a question. One could address this limitation by extending the base skill map model with a belief network derived from the skills graph with prior and conditional probabilities for the nodes.

Moreover, the trade-off for LST's generality is the complexity of validating the learning space. This commonly involves systematically questioning experts to capture their personal knowledge structure. The limitations include a lack of consistency between the experts' structures and the large number of questions the experts have to answer.

Because the CB prerequisite graph is designed by domain experts we can assume that it is *correct* but possibly *incomplete*. When course developers underestimate relationships between skills, they have failed to recognize their own *expert blind spots*: areas where the very understanding necessary to teach a domain leads experts to miss the learning needs of novices [4]. Thus we can use a validation algorithm which focuses only on adding the missing prerequisites.

References

- [1] Ryan Baker, Albert Corbett, and Vincent Alevan. 2008. More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. Intelligent Tutoring Systems Lecture Notes in Computer Science, 406-415.
- [2] Norman Bier, Sean Lip, Ross Strader, Candace Thille, and Dawn Zimmaro. 2014. An Approach to Knowledge Component/Skill Modeling in Online Courses. White paper. Retrieved October 5, 2015 from <http://oli.stanford.edu/>.
- [3] Jean-Clause Falmagne and Jean-Paul Doignon. 2011. Learning Spaces. Interdisciplinary Applied Mathematics. Springer.
- [4] Pamela Hinds. 1999. The curse of expertise: The effects of expertise and debiasing methods on prediction of novice performance. *Journal of Experimental Psychology: Applied*. 5, 2: 205-221.
- [5] Marsha Lovett. 2012. Cognitively informed analytics to improve teaching and learning. Presentation at EDUCAUSE Sprint. Retrieved October 5, 2015 from <http://www.educause.edu/ir/library/powerpoint/ESPNT122Lovett.ppt>.
- [6] Barbara Means, Yukie Toyama, Robert Murphy, Marianne Bakia, and Karla Jones. 2010. Evaluation of Evidence-Based Practices in Online Learning: A Meta-Analysis and Review of Online Learning Studies. US Department of Education. Washington, DC/
- [7] Gilbert Paquette, Olga Mariño, Delia Rogozan, and Michel Léonard. 2015. Competency-based personalization for massive online learning. *Smart Learning Environments*. 2, 4.