SkillBuilder: Interactive Learning Resource Graph.

Alex Jose, Christophe Kunesh
Department of Computer Science
CMS, California Institute of Technology
1200 E. California Blvd
Pasadena, California 91126
{ajose, ckunesh}@caltech.edu

ABSTRACT

There is no shortage of learning materials, both on and off of the internet. However, an important piece of information that is typically lacking from these resources is, given a set of consumed learning resources and a particular learning goal, what the best "next step" to take would be in order to most efficiently reach your learning goals. In this paper we discuss a method for organizing the diverse learning materials that exist on the internet, and implement this method as a web-app.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Intelligent Tutoring Systems, Curriculum Graph

1. INTRODUCTION

The internet is full of useful learning resources - PDFs of textbook chapters, Khan Academy videos, one-off educational blog posts, and so on. Most of these resources, unfortunately, are not organized at a larger scale - it is hard to know what a good 'next step' is from the resource you just used. Even resources that are organized in a local sense, like a multi-part blog post or video series, lack the global contextualization that could, for example, be provided by a knowledgeable mentor ("So you got a lot out of this video? Then you should look into these other topics and books to reinforce your understanding and progress to more advanced topics").

To find a solution for this problem, we can take inspiration from what has emerged to be the typical 'context' for learning materials - curricula. There are a great number of different 'scales' of curricula: at the smallest scale might be a TODO list for completing a homework problem; a slightly larger scale could be a sequence of material to be consumed or produced for completion of a specific class; a degree program could be seen as a larger-scale curriculum composed of these smaller ones; and we might currently define an 'education' as a sequence of these meta-curricula.

These various scales of curricula can be seen as a single, muti-scale digraph. Most basically, we could have nodes represent singular "learning resources", and edges represent a directed 'link' between two resources, meant to imply that there exists a pedagogical relation and ordering between those two nodes.

A path over nodes in the digraph would represent a self-contained 'curriculum', taking a user from one point in 'skill-space' to another. Various other graph statistics and algorithms could be applied to extract useful information from such a 'learning resource graph' - pathfinding, search, degree, and so on.

2. RELATED WORK

This is both an academic and practical problem, so we reviewed both academic papers and existing websites to get an idea of prior work in the field.

2.1 Websites Reviewed

A currently popular form of organizing online learning material is the MOOC (Massively Open Online Courseware). MOOCs are very similar to meat-space classes in the sense that there is typically a linear, non-branching sequence of learning resources created specifically for the course (or adapted from a similar course).

A resource very similar to what I propose is the Khan Academy exercise dashboard , which organizes content from Khan Academy into an interactive graph. While the current version of the page does not seem to show it, I think previously the page also had hierarchical categorization of the learning resources, such that users viewing the graph at a high level saw general categories like 'algebra' or 'calculus', and upon zooming in saw more and more specific topics until they reached the level of individual Khan Academy pages.

Metacademy, a site we learned about soon after starting the project. Of all the sites reviewed, it is most similar to what we imagine a mature version of SkillBuilder would look like, save for one thing: Metacademy's focus on expert-curated, rather than community-curated, learning paths. Furthermore, the site is currently focused on machine learning only, but its creators plan to other topics soon. Whether expert-

curated or community-curated paths are the right choice remains to be seen; Metacademy's model will no doubt provide better content initially, but I fear the site will not scale well as it transitions to covering other topics.

Haskell.org has an interesting meta-tutorial, which demonstrates the usefulness of 'branching curricula' - rather than presenting a linear sequence of learning material to consume, the meta-tutorial attempts to provide more specific recommendations based on more thorough categorization (like having a set of tutorials for users that are new to programming, or areexperienced with some language but have never done functional programming, and so on).

The Math Atlas, "a gateway to modern mathematics", is a "a collection of short articles designed to provide an introduction to the areas of modern mathematics and pointers to further information, as well as answers to some common questions. The material is arranged in a hierarchy of disciplines, each with its own index page." For example, you can click on the index for "Abstract harmonic analysis" and be taken to a page linking to relevant sub-topics, like "Measure on groups and semigroups", "Hypgergroups", etc. While the site is dated, it's very similar to what this project (or, at least, the math sub-portion of this project) would ideally be like.

The University of Illinois at Urbana-Champaign site has an interesting interactive curriculum graph that automatically plots course paths from the a root node to any desired course in the Engineering curriculum. For example, I can mouse over the 'Biophotonics' course node, and see the highlighted path of all the prerequisite courses for that class.

The Stacks Project is an interesting website about "algebraic stacks and the algebraic geometry needed to define them" which makes use of a digraph of proof results to construct a collaboratively-made textbook.

2.2 Literature Reviewed

Constantinescu [1] represents school curricula as DAGs and uses the Highest Level First with Estimated Time (HLFET) scheduling algorithm to generate valid (in the sense that all prerequisites are met before taking a given course) course schedule. Some of the algorithms used in the paper necessitate the curriculum graph be acyclic, but it seems that allowing cycles would make for more flexible representation. The scheduling program also makes use of a 'node cost' representing the amount of time necessary to 'execute' a course, which can be used for scheduling purposes and to give an idea of how heavy a course load is.

Lightfood [2] introduces a number of interesting questions about course scheduling in the context of curricula in higher education. Specifically, the questions are: 1) Where in the curriculum should introductory topic coverage be placed? 2) Where should reinforcement of assessment topics take place? 3) Where should primary and secondary objective assessment be located? These could all be relevant to determing a satisfactory structure for a more general curriculum graph. The paper recommendsusing a variety of graph-theoretic measure in determining specific answers to these questions. The paper specifically recommends that: "the out-degree can be used to locate those courses best suited to introduce topics and perform baseline assessment. The in-degree is useful in locating courses where exit assessment and higher-level learning activities should take place. The centrality measures of betweenness and eigenvector are valuable in identifying courses appropriate for assessment and topic reinforcement. Finally, the clustering coefficient can be used to find courses best suited to implement changes into a tightly connected clusters of courses."

Gestwicki [3] This paper describes the usefulness of curriculum graphs to various members of an academic community, in both a fine-grained (seeing specific paths to graduation or the shortest path to take a specific course, visualize future effects of taking a specific course) and coarse-grained (for administrators - see shape of curriculum, find 'hidden' prerequisites, etc.).The paper also goes into specific UX recommendations for implementing such a graph.

Auvinen [4] proposes creating a curriculum graph not over courses, but over the concepts taught in courses ('learning outcomes') using the STOPS model. This approach offers a variety of advantages, namely ensuring that all prerequisite topics are sufficiently covered without overlap before starting a course (information that could be lost if nodes in the graph represent courses only), to motivate the material taught in a course by showing how it will be useful in the future, and to allow students to choose course schedules based on their desired future competences.The paper states that the design can be applied to any field where "knowledge is hierarchical and courses have prerequisite connections"; what kind of material would be a poor fit? This paper also makes it clear that boolean set membership might be less descriptive than desirable - in addition to having fuzzy resource relations, it would be useful to have fuzzy set membership (for example, a blog post could be somewhat related to one topic, but very related to another).The paper also covers the basics of Intelligent Tutoring Systems (ITS), which can help guide learning materials to, say, help a student understand a type of problem he or she is item having trouble with.

Rollande [5] is notable for taking a more formal approach to graph-based curricula, and for AND/OR graph to represent multiple scales of a curriculum.

Fung [6] presents a knowledge representation format specifically for representing information about electronic learning materials, using a hybrid approach of adirected graph combined with finite state machines. One interesting thing is the use of different types of set-membership; specifically, type-of and part-of relations, which each have different practical meanings in the context of a hierarchical knowledge graph. For example, mean, median, and mode are types-of Central Tendency, meaning they should inherit some of the attributes of Central Tendency. Part-of relations indicate complementary membership - i.e. A and B could be part-of C, such that representing C as only A or B separately would be incomplete - A and B must be used in conjunction to fully represent C. This difference is practically similar to the AND/OR graph from the previous reference. There is also a distinction in prerequisite links - contained-in (i.e. topic X is used in describing topic Y) vs. applied-to (i.e. topic X provides an 'initial condition' for topic Y to be learned)

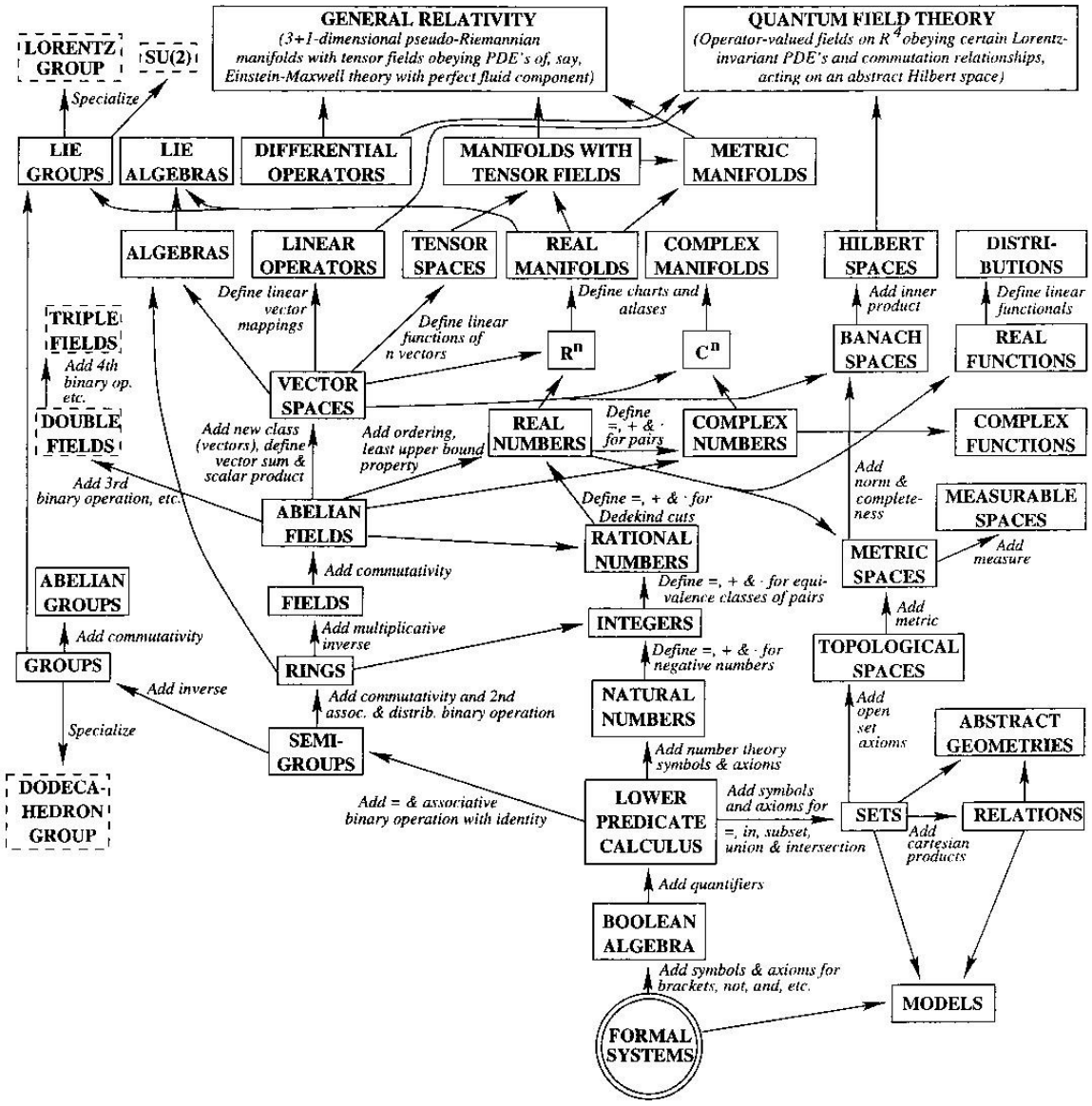

FIG. 1. Relationships between various basic mathematical structures. The arrows generally indicate addition of new symbols and/or axioms. Arrows that meet indicate the combination of structures — for instance, an algebra is a vector space that is also a ring, and a Lie group is a group that is also a manifold.

Figure 1: A digraph of math topics.

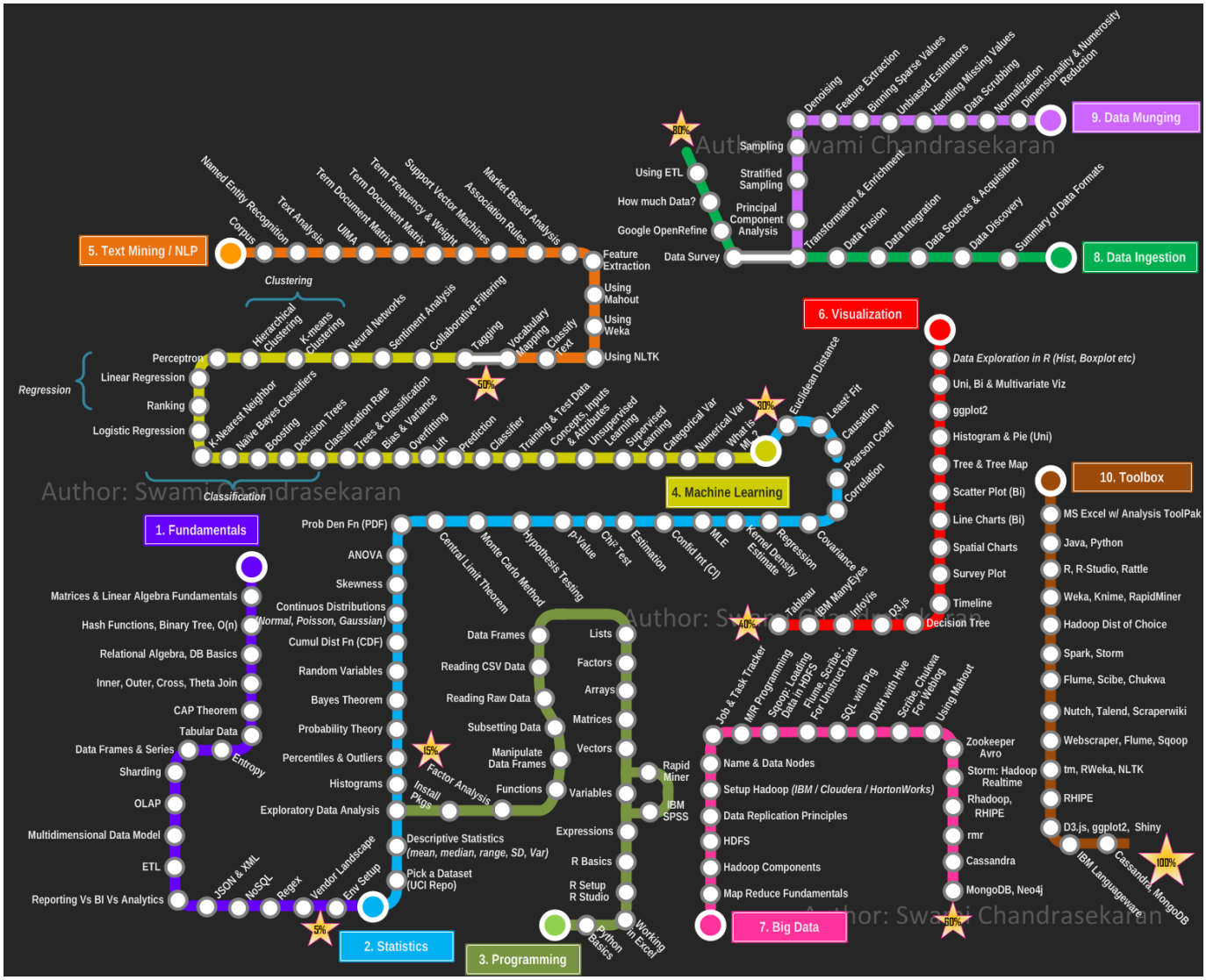

Figure 2: A 'learning path' on Data Science.

links.

Gunel's paper [7] is partially interesting due to its coverage of the student-modelling aspect of an 'Intelligent Tutoring System'. Modelling the abilities of the learner will be an important part of effectively teach material. The paper also looks at how the difficulty of learning material could be automatically calculated.

Hwang [8] introduces is the "item test relationship table", an alternative to typical graded tests wherein the result is a list of 'mastered topics' rather than a grade, allowing students to see where they need more work, rather than the grade-based case, where "multiple learning outcomes are assigned to the same grade".

3. DISCUSSION AND GOALS

There are a variety of technical challenges that need to be solved in order to create and maintain a learning resource graph. A few examples:

- What is the most useful and efficient datastructure for maintaining the graph? What set of information is necessary and sufficient for constructing the graph?
- Assume you have a collection of learning resources and an oracle that can tell you some information about each resource (perhaps some some combination of: categories that resource belongs to, prerequisites for that resource, quality of the resource, difficulty of the resource, etc.). How can you construct a learning resource graph?
  - Now assume that the oracle gives only partial information and is frequently wrong - how can you ultimately arrive at the same 'true' graph?
  - Alternately, given a community of agents that are incorrect (faulty internal model), deceptive (fabricates responses), and lazy (contribute infrequently), how can we arrive consistently and quickly at ground truth learning resource graph?
- Assume a learning resource is added randomly to a pre-existing graph - how can you move it to its proper position? What series of questions could you ask a 'stupid oracle' to move the resource to its proper position? How can we display the graph such that its data is easily accessible to users? How much of this process can be automated? What precisely must be done by the community, and what can be done by a computer?

3.1 Reputation System

Upon looking at several related resources such as Metacademy and Expii, we noticed that while they certainly allow for user interactivity in some sense, the content is static. Any changes to the learning resource graph must be made directly by those that maintain the site. While this is certainly a valid approach, we believe it to be a very limiting one. For example, Metacadamy was developed by two experts in Machine Learning, and as such most of the content is limited to the topics of Artificial Intelligence and Logic. This may be fine for users who want to learn only about these topics, but it is not scalable to all the other types of learning. Unfortunately, just bringing in more and more experts to add content will not adequately solve the problem. There are too many learning resources and the web of knowledge is too vast to be optimally curated by a few experts. We believe one potential solution is to crowdsource the learning graph, which will make it fully scalable. It also has the potential to generate a more robust and optimized graph. This is similar to how crowdsourced medicine can diagnose a rare illness even the most experienced doctor may not be aware of.

Of course, this has its own challenges. One of the biggest issues with allowing any user to modify the learning resource graph is ensuring that the changes make sense. Users who are malicious or simply not qualified for certain topics will introduce suboptimal changes to the graph. To tackle this problem we drew inspiration from websites such as Quora and StackOverflow that have to deal with communities of users adding content. A common element we noticed is that users develop a quantifiable "reputation" over time. Those users who consistently add content or make changes that are well-received by the other users get higher scores, and in turn often get more influence on the site's direction and content, whereas those who make malicious or ill-informed decisions get lower scores, and in turn their effects on the site are minimized. In a sense, instead of curating the content directly, we can let the users curate the content amongst themselves. This in turn curates the users, which will indirectly curate future content. Much like the process of evolution, the learning graph and the community of users will become more and more optimized as time goes on.

To explore how a reputation system could be implemented to help optimize the growth of the learning graph, we developed a PyGame demo over the first few weeks, both as a proof of concept and a fallback should the web-based app not reach completion. We also made an initial "usage simulation" of the site, which attempts to construct an approximation of a ground truth graph given a bunch of "users" with slightly mutated versions of that graph. Then, using a simple consensus-based reputation system, we can filter the user-suggested graph to arrive at something more similar to the ground-truth.

The other important aspect of implementing a user-modifiable learning graph is to ensure that the graph is both visualizable and easy to modify, with changes saved to our main graph database. After finishing a prototype for the reputation system, we turned our focus to developing a web application.

3.2 Web App

Since our reputation system prototype was implemented using the Python language, we initially considered using a Python web framework in the hope that it would speed up the development process. We believed that it would both minimize the time spent learning a new language and also simplify the integration of the PyGame demo into the final web app. Django seemed to have the most support of any of the Python frameworks, so we spent the first couple of weeks familiarizing ourselves with it. We read and watched various Django tutorials about the framework and how to

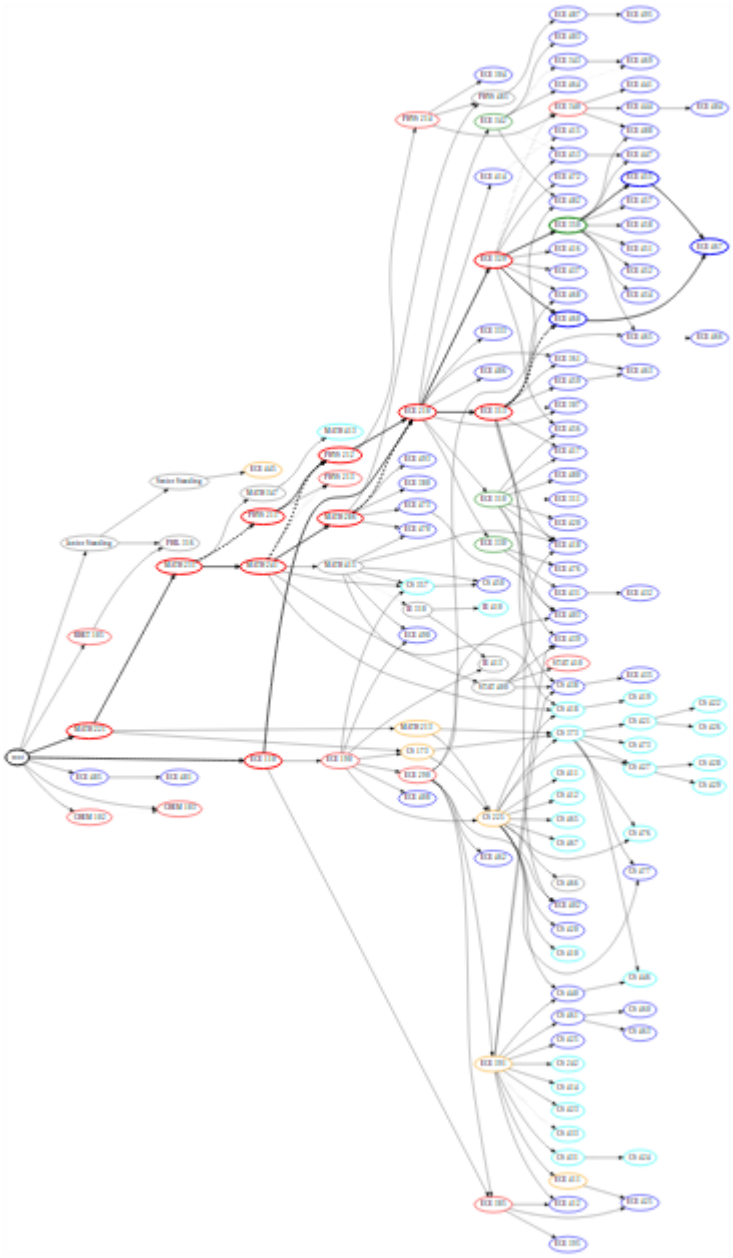

Figure 3: A digraph of college courses, with the prerequisites of one course highlighted.

set it up, and before long we managed to get a rudimentary app working locally. We also researched various packages and tools we thought might be helpful. This including looking at different different data visualization tools, including several JavaScript libraries. After a bit of deliberation, we decided to use vis.js for our graph visualization, but D3.js was also a fallback option if we needed more functionality later on. One of the benefits we noticed of using vis.js over D3.js is that it is graph-specific, and it has a much simpler learning curve as well. It also seemed to have enough configuration options and flexibility for at least a simple app prototype, and soon were able to get some of the vis.js example graphs working with our framework.

Another important issue we had to tackle was to figure out our database management system. Although we were both used to using relational databases, since the crux of our app is to maintain and display a graph, we decided to switch to using a graph database for our project. The Neo4j graph database in particular seemed well-suited to our task at hand. Since the data would almost certainly be very connected, we believed a graph database would be advantageous for our traversal-like queries. It would alleviate the problem of potentially costly recursive joins and make the graph highly scalable, which was a crucial aspect of our project.

This decision led us to reconsider using Django for our web framework. Although Django does support using NoSQL databases, it is optimized for relational databases. We thus decided to change course and use Node.js with Express for our backend framework. One of the nice benefits of using Node.js is that both the backend and the frontend would be mostly JavaScript. Also, there seemed to be good support for using Node.js with a graph database like Neo4j, so we believed that the switch from Django to Node.js would help simplify the process of developing the site. After finalizing which tools we were going to use, the next step was to start to fleshing out the website.

The last few weeks we worked on coding our demo site, and we were able get a demo of a learning graph working on the front-end, where visitors are able to add nodes (learning resources) and edges (relationships) to an initial graph. Also, we added a template such that selecting a particular learning resource displays relevant data including a description, links to useful tutorials, as well as the resource's prerequisites and what it is a prerequisite for based on its relationships in in the graph. We also added clustering to the nodes. This way, the visualization would scale well for a lot of learning resources. As the user zooms out, the nodes become more and more clustered, and as the user zooms back in the clusters open back up. Additionally, using Ajax and Neo4j's REST API we were able to get user changes to nodes on the visualization to persist to the database.

4. CONCLUSION AND FUTURE WORK

We have not been able to get the relationships to persist correctly to the database but in future work we would get all aspects of the frontend and backend working together seamlessly. We would also want to generate the initial graph using the data in our Neo4j database. Integration of the PyGame reputation system will optimize the scalability of the app. The UI can also be improved quite a bit to help improve user experience and make it more appealing for people to use the web app and make modifications to the learning graph.

Given the growing number of very similar sites, it's hard to want to continue work on this project - there are very likely other, larger teams that are better equipped to solve the problems this kind of project presents than we are. Even so, we have yet to see a site representing the precise combination of utility and scalability that we think is necessary to make a site like SkillBuilder successful.

5. ACKNOWLEDGMENTS

The authors would like to thank Qiuyu Peng and Professor Steven Low for his guidance on this project. All funding was provided by the Computer Science department of the California Institute of Technology.

6. REFERENCES

[1] Constantinescu, Irina, and Flavius Manea. "Scheduling Courses of the Academic Curriculum." Computer Science Master Research 1.1 (2011): 42-48.

[2] Lightfood, Jay M. "A Graph-Theoretic Approach to Improved Curriculum Structure and Assessment Placement." Communications of the IIMA 10.2 (2010): 59-74.

[3] Gestwicki, P., "Work in progress - curriculum visualization," Frontiers in Education Conference, 2008. FIE 2008. 38th Annual , vol., no., pp.T3E-13,T3E-14, 22-25 Oct. 2008

[4] Auvinen, T. "Curriculum Development Using Graphs of Learning Outcomes." First EUCEET Association Conference New Trends and Challenges in Civil Engineering Education, Patras, Greece. 2011.

[5] Rollande, R.; Grundspenkis, J., "Graph based framework and its implemented prototype for personalized study planning," e-Learning and e-Technologies in Education (ICEEE), 2013 Second International Conference on , vol., no., pp.137,142, 23-25 Sept. 2013

[6] Fung, I.P.-W., "A hybrid approach to represent and deliver curriculum contents," Advanced Learning Technologies, 2000. IWALT 2000. Proceedings. International Workshop on , vol., no., pp.209,212, 2000

[7] Gunel, Korhan, and Rifat Asliyan. "Determining Difficulty of Questions in Intelligent Tutoring Systems." Turkish Online Journal of Educational Technology 8.3 (2009).

[8] Hwang, Gwo-Jen. "A conceptual map model for developing intelligent tutoring systems." Computers Education 40.3 (2003): 217-235.

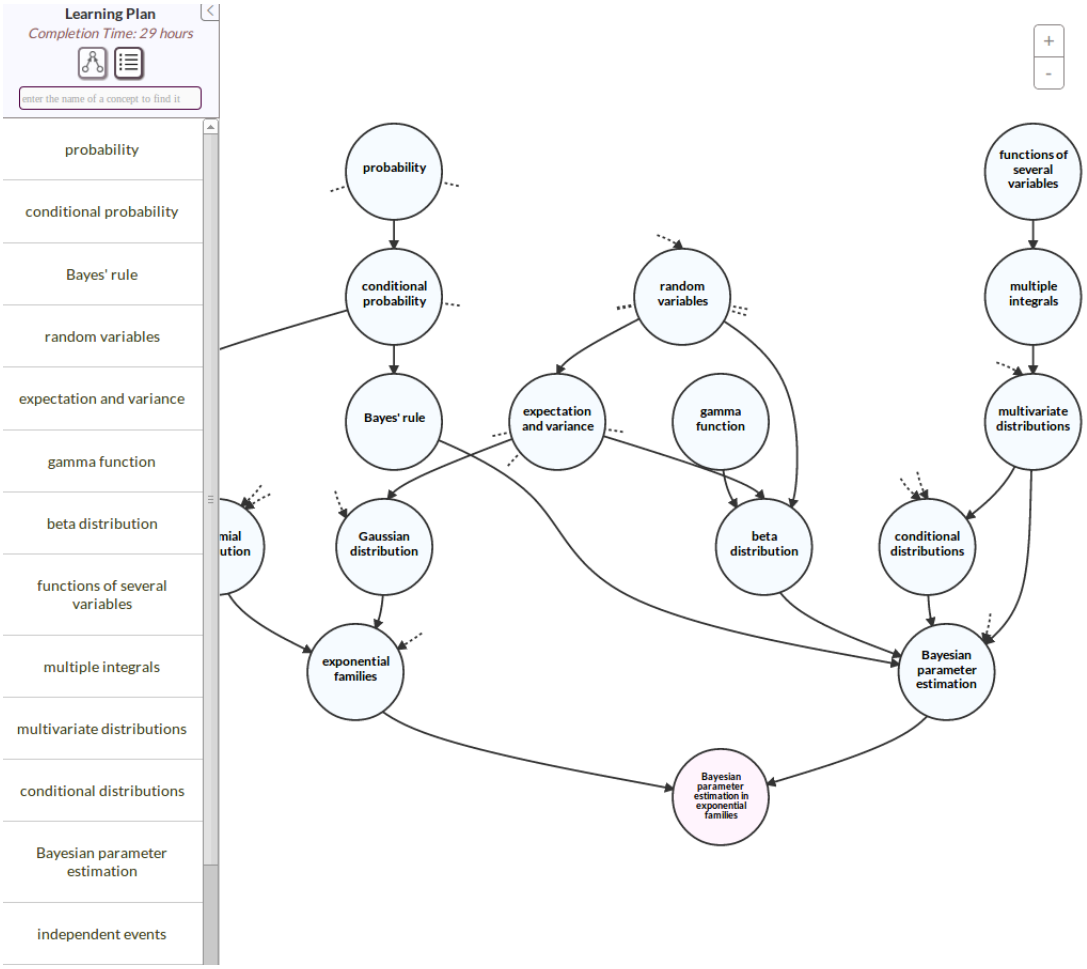

Figure 4: Metacademy's interface.