Building Multi-layer Social Knowledge Maps with Google Maps API

MinEr Liang[1], Julio Guerra[2,3] and Peter Brusilovsky[2]

[1]School of Computer Science, Fudan University, Shanghai, P.R. China
michelle_liang@fudan.edu.cn

[2]School of Information Sciences, University of Pittsburgh, Pittsburgh, PA, USA
jdg60@pitt.edu, peterb@pitt.edu

[3]Instituto de Inform ática, Universidad Austral de Chile, Valdivia, Chile
jguerra@inf.uach.cl

Abstract. Google Maps is an intuitive online-map service which changes people's way of navigation on Geo-maps. People can explore the maps in a multi-layer fashion in order to avoid information overloading. This paper reports an innovative approach to extend the "power" of Google Maps to adaptive learning. We have designed and implemented a navigator for multi-layer social knowledge maps, namely ProgressiveZoom, with Google Maps API. In our demonstration, the knowledge maps are built from the Interactive System Design (ISD) course at the School of Information Science, University of Pittsburgh. Students can read the textbooks and reflect their individual and social learning progress in a context of pedagogical hierarchical structure.

Keywords: Adaptive Learning, Multi-layer Social Knowledge Maps, Google Maps API, Treemap

1 Introduction

Google Maps is an intuitive online-map service [10]. It changes people's life by providing navigation of GIS systems in a multi-layer fashion so people do not need to worry about information overloading. For example, when travelling from Minneapolis to Pittsburgh in the U.S., Google Maps shows you need to travel six states on the interstate highway system. As you travel, you can zoom in or out to different levels of detail depending on your local, city, state, or interstate traffic. A multi-layer zoomable interface of Google Maps provides remarkable navigation support flexibility: on each layer it shows most critical details that are required to make decisions on this layer of navigation. Is it possible to build such an intuitive navigation tool for learning? There is a certain need for a multi-layer navigation support in E-learning. When making decisions about our learning we do operate on several layers. For example, on the college education layer, students think about degrees and courses, paying little attention to the structures below the course level. On the course level, students think about course topics, book chapters, projects and other top-level course elements. Within a specific topic or chapter they consider sub-topics and activities and so forth. Can we build a so called "Google Learning Maps" that can provide a multi-layer support of student learning?

However, multi-layer knowledge maps are not as easy to build as multi-layer geographic maps. The problem is that there are two conflicting approaches for knowledge organization, both being popular in knowledge modeling and personalized learning: cognitively-oriented concept based modeling [5] that treats a body of knowledge as a network of relatively fine-grain concepts, and pedagogically oriented topic based modeling [16] that represents knowledge as a taxonomy of relatively coarse-grain topics. What is the right approach to build multi-layer knowledge maps to bridge this two approaches?

The good news is: the integrated hierarchical structure of a pedagogical model, a knowledge model and a student model is usually captured in the adaptive courseware environment as a web-based tutoring framework [17]. We hypothesize that it is possible to build multi-layer knowledge maps based on this integrated hierarchical structure. Furthermore, with the advancement of Web 3.0, it is promising to embrace the open social student model [11] so that students can benefit from both cognitive and social learning. In response to the demand above, we propose an innovative approach to build Multi-layer Social Knowledge Maps with Google Maps API. The Multi-layer Social Knowledge Maps is based on the integration of a pedagogical model, a knowledge model and an open social student model. It is important to provide navigation for students to interact with the learning resources in the Multi-layer Knowledge Maps, similar to how you can explore with the navigation on Google Maps. Furthermore, students need a sense of direction when they progress; otherwise, it is easy to get lost in the learning process. It is important for learners to reflect both individual and social learning progress, “similar” to what you can do with the GPS positioning system.

In the next section, we provide a short literature review on the related work. The overall design of building multi-layer social maps is provided in section 3. In section 4 we present the architecture. We demonstrate the interface component Progressive-Zoom in section 5. Finally, we summarize this work and discuss the future research plan.

2 Related Work

Open student model is used to support meta-cognitive activities such as reflection, planning, and self-assessment by providing feedback with respect to students’ learning and knowledge [2]. Open social student model is provided for students’ benefit from both cognitive and social learning [11]. Social Visualization on student performance helps students get more engaged in learning activities and have better performance in self-assessment quizzes [8,11]. However, this visualization work is based upon a one layer knowledge map.

Varied efforts were devoted to build knowledge maps which are more than one layer. In Knowledge See II [4], two-layer knowledge maps were developed based on the Self Organized Maps (SOM) to provide adaptive navigation support. In QuizMap [6], multi-layer knowledge maps were developed based on Treemaps to facilitate both open social student modeling and adaptive navigation support. But no Zoomable interface is provided to address the information overloading issues, which is the main contribution of our approach.

Our approach is mainly inspired by the intuitive visualization of an online mapping service, such as Google Maps [10], which provides incrementally more detail in the domain of GIS systems. In the domain of e-learning systems, a concept-map navigator is provided by KHAN Academy [7] based on Google Maps API. But it fails to provide a balance of local/global view, and students may easily get lost in their learning process. In the domain of bioinformatics, the Google Maps API is used to visualize multi-scale structures, e.g. Genome Projector, Protein Interactive Network, and Gene Co-expression [1,12-13].

3 Building Multi-layer Social Knowledge Maps

As explained in the introduction, the corner-stone of multi-layer social knowledge maps is the integration of a pedagogical model, a knowledge model and an open student model for adaptive learning systems.

The pedagogical model provides the instructional practice, which is usually organized in a hierarchical layering structure, for example, lectures, topics and sub-topics. And it follows an instructional sequence, for example lecture 1, lecture 2, etc. The knowledge model structures concepts by establishing the relations between them. In this work, we structure concepts following the organization of the material from which they are extracted (pedagogical model) filtering the important or relevant concepts in each layer. As a result, key concepts are usually introduced from general to specific in the pedagogical model. The open user model keeps track of the students' progress when they navigate through the multi-layer knowledge maps.

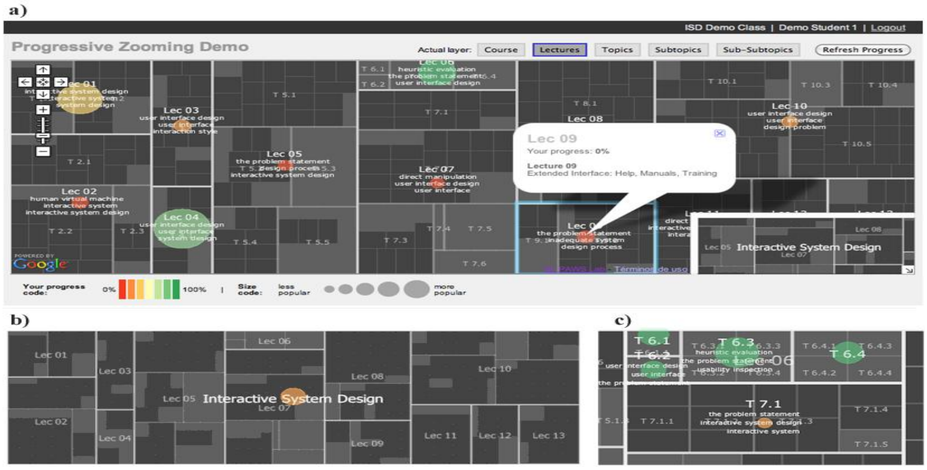

Fig. 1. ProgressiveZoom **interface.** The navigation through the course hierarchy of topics (lectures, topics, sub-topics) is performed using the semantic zooming paradigm: for example, by zooming in or out from the level of the lectures (shown in this figure), the next level of topics is shown and the information is displayed according to that. At each level, key concepts are presented to help students monitor progress. The feedback is presented in the form of multi-color and variable size bubbles. The size of the bubble represents the group popularity from less popular to more popular. Each instruction unit can be clicked, which leads to further leaning material and tools. See demo at http://adapt2.sis.pitt.edu/pz/test.jsp

As shown in Fig.1, this integrated hierarchical structure is visualized with the Treemap layout [15]. The related learning materials, tools, and student's learning progress can be captured as overlay along the hierarchical structures, filtered by these key concepts.

Through the ProgressiveZoom interface (Fig. 1), students can zoom in or out in different levels of detail. We hypothesize that this approach will help students increase the awareness and engagement in their learning process.

Currently we focus on building the knowledge maps from one course, the Interactive System Design (ISD) course at the School of Information Science, University of Pittsburgh. Nevertheless the multi-layer social knowledge maps can be extended to include other courses and disciplines/degree programs, similar to how Google Maps can include other countries and continents. This addresses the scalability and global/local view challenges in visualization.

3.1 Pedagogical Model

The structure of the course is organized in layers. From the top layer, the sequence of courses is defined in course layer (only one course is shown in our current implementation). The sequence of lectures is defined in the lecture layer. For the topic layer, more than one topic can be defined for each lecture. Each topic can be a book chapter or a section of a book chapter. The first three layers are manually defined by the instructor, which conforms to the pedagogical practices.

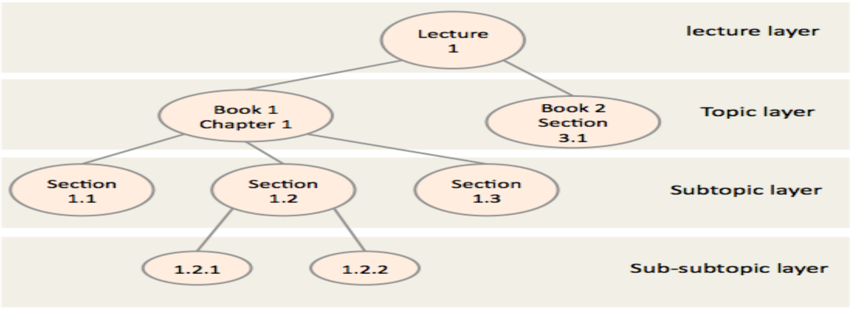

Fig. 2. The Pedagogical Model

The subsequent lower layers are automatically structured by analyzing the sections and subsections of the chapter or sections in the topic layer. Our current approach is to exploit the numbering of the chapter and section headings. For example, if the topic layer (third layer) has the heading "1 Introduction", then the next layer will be all headings from the same book starting with "1.1 ...", "1.2 ..." etc. Analogously, the documents in the fourth layer corresponding to the third layer item "1.1 ..." will be "1.1.1 …", "1.1.2 ...", "1.1.3 …", etc. As shown in Fig. 2, the structure for each lecture forms a tree in which each node and its child nodes represent a document and the sub-heading documents respectively.

3.2 Knowledge Model

The knowledge unit is a concept. Key concepts are concepts that suit the "level of generality" of the corresponding layer in the pedagogical model. The relevant concepts are extracted automatically using Yahoo! Content Analysis API [20]. Then concepts are structured in layers of generality overlaying the pedagogical structure. The method performs recursively from the lowest layer to the highest in every node as follows:

First, it aggregates all concepts from the child nodes and adds all concepts extracted from the document itself. In higher levels, there will be a considerable list of concepts!

Secondly, it computes the "generality" measure of each concept by summing the term frequency of the concept in a node and all the generality of its child nodes (see Equation 1.) We consider generality as a combination of relevance of a concept in the node and its child nodes. This means that a concept will be general if it is relevant in the node and in all (or the majority) of its child nodes. Since we extracted concepts instead of terms, we consider the term frequency as a simple and reasonable measure of relevance.

$$g_{k,e} = tf_{k,e} + \sum_{c \in C} g_{k,c} \; (1)$$

Here $g_{k,e}$ is the generality of the concept k in the node e, $tf_{k,e}$ is the term frequency of k in the node of e, C is the set of all child nodes of e and $g_{k,c}$ is the generality of the concept k in the child node c.

Currently, we select 3 concepts with the highest level of generality to display as key concepts for each node.

Equation 1 seems to work fine for our first prototype. However, further evaluation and revision of the concept extraction and generality computation is needed and planned in the future work.

3.3 Open Social Student Model

The Open Social Student Model keeps track of the students' individual and group progress by propagating the progress from higher to lower layers and by aggregating from the lower layer to higher layer along the pedagogical hierarchical structure. Currently, the student model is simplified to reflect the progress of the student in reading the documents of the course. In this approach, the individual progress of the student in one specific node is recursively calculated by aggregating the progress in its child nodes as specified in Equation 2.

$$p_{u,e} = \frac{v_{u,e} + \sum_{c \in C} p_{u,c}}{|C| + 1} \quad (2)$$

Here $p_{u,e}$ is the progress of the student *u* in the node *e*. The notation $v_{u,e}$ takes value 1 if the student *u* has read at least once the documents in the node *e*; the value of 0 otherwise. *C* represents the set of all direct child nodes of *e*. The value $p_{u,c}$ is the progress of the student u in the child node *c*.

In order to promote social learning, we also present the group progress along the pedagogical structure. Currently we use the aggregated popularity as a simple group progress indicator (see Equation 3.)

$$pop_{g,e} = \frac{w_{g,e} + \sum_{c \in C} pop_{g,c}}{|C| + 1} \quad (3)$$

Here $pop_{g,e}$ is the popularity level of the node *e* for the group (class) *g*. The value $w_{g,e}$ is the number of times the document in the node *e* has been visited. *C* is the set of all the direct child nodes of *e*. The value $pop_{g,c}$ is the popularity of the child node *c*.

4 The Architecture

As shown in Fig. 3, the architecture consists of three blocks. At the first block (left), we build the pedagogical model from the course syllabus and textbooks. Five textbooks for the course Interactive System Design are used. All the text information of these books have been extracted and stored in XML files.

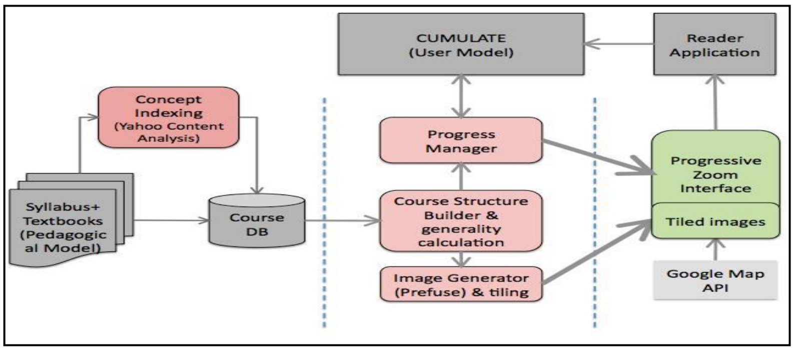
CUMULATE (User Model)
Reader Application
Concept Indexing (Yahoo Content Analysis)
Progress Manager
Progressive Zoom Interface
Syllabus+ Textbooks (Pedagogical Model)
Course DB
Course Structure Builder & generality calculation
Tiled images
Image Generator (Prefuse) & tiling
Google Map API

Fig. 3. The Architecture

Using the syllabus, we define the first higher 3 layers: course, lectures and topics. All topics are chapters or sections previously selected by the instructor from the textbooks. Lower layers (sub-topics, sub-subtopics) are built automatically and correspond to the sections and subsections of the book chapters or sections defined as topics in the topic layer.

We use Yahoo! Content Analysis API to extract concepts from textbooks by inputting the content from the text books files (in XML format). Then we calculate the term frequency for every concept in every node (document) stored it in the database.

At the second block (middle) we combine the knowledge model and the pedagogical model into an integrated layered structure that every node will show its top three concepts. The Course Structure Builder module is a Java class that reads the course definition from database, build the hierarchical structure by analyzing the documents, get the concepts and aggregate them through the structure and compute the concept generality.

The Image Generator module uses the Prefuse Visualization Toolkit [19] for generating both images for the different layers and a coordinate XML file for further interaction. Images are generated using a Squarified Tree Map algorithm [3]. Images are tiled (cut and layered) by using G-Language API [9]. The Progress Manager computes the aggregated individual and group progress, and updates the student model with the actions captured in the interface. The student model is stored in the CUMULATE User Model Infrastructure [18].

At the last block (right), we implement the Progressive Zoom interface (Fig. 1). We use Google Maps API and tiles images generated from the Treemap layout to build custom maps. Progress and popularity level indicators are shown as overlays to the base map. We also link the interface to a textbook reader, which has been implemented in KnowledgeSeaII [4].

5 ProgressiveZoom: the navigator

ProgressiveZoom (Fig. 1) provides an interactive visualization interface for students to interact with learning resources and to reflect individual and social learning status based on the multi-layer social knowledge maps.

5.1 The Visual Coding

We use the visual coding to attract students to the right attention. Bubbles of different colors and sizes (Fig. 4) represent the progress of the user and the popularity of the node respectively. Color represents progress from red (0%) to green (100%) in a seven step scale (each part represents 1/7 of 100% progress).

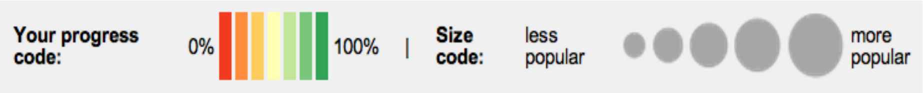

Fig. 4. Individual progress with diverging color scheme and group popularity with different bubble sizes. It is intuitive to guide students to pay attention to bigger red bubbles.

The size of the bubbles represents group popularity. Students can easily spot where the hot topics are according to the bubble size. As the Google Map API uses scales from the earth, the radius of a bubble is measured in kilometers (see Equation 4.)

$$r = \frac{50000gp + 300000}{zoomlevel} (4)$$

Where *gp* is the group popularity calculated as described in Equation 3 above. In the level of zoom 1, Null visits (null popularity) in a node is represented as a 300 km radius bubble. Each point in the level of popularity increases the bubble radius in 50 km. The value is divided by the level of zoom for keeping consistency of what sizes represents independently of how many pixels are assigned per length unit.

5.2 Interactive Interface

We follow the visual-information-seeking-mantra: "Overview, zoom, filter, then details on demand." [14]. Specifically, Semantic Zooming is used to show different layers at different zoom levels. Overlay is also shown differently according to the layer of the zoom level. Details on demand are also implemented. The click event triggers an overlay balloon window containing data of the node. By clicking on the link that appears on the balloon, a textbook reader is loaded in a new browser window and the activity is recorded in the database. By pressing the refresh progress button, the new progress can be observed by a change of the color on the corresponding bubble.

As a result, students can navigate through the course material without information overloading. For example, as shown in Fig. 1, when student-one first accesses the ISD course, his individual progress is in light red as an overview of the course. Then he zooms in to lecture level, and checks out the status of each lecture. Now he clicks on Lecture9 to diagnose the problem. He can see no progress is made in this lecture. He can continue to zoom into topics level and read the textbook through the reader. After he has worked on the material, the progress is aggregated and reflected on the higher lever, for example the lecture and course level. He can also compare with his peers by the bubble sizes, which gives him a hint how popular that learning unit is.

6 Summary and Future Work

In this paper we have presented an innovative approach to build multi-layer maps with Google Map API. In our demonstration, the knowledge maps are built from the Interactive System Design (ISD) course at the School of Information Science, University of Pittsburgh. It allows students to read the textbooks and reflect their individual and social learning progress in a context of pedagogical hierarchical structure.

For the future research we plan to improve the domain modeling and the interface design. From the modeling perspective, we want to evaluate the concepts extracted and to explore further mechanisms of information extraction. Also, a better track of the user progress and user knowledge model should be done by aggregating and propagating the level of progress among concepts. Concept generality calculation should be evaluated and revised further. From the interface perspective, we plan to provide explicitly support for self-guided learning through dynamic query, overviews and details on demand, and linked view. These tasks include Diagnose learning needs, Formulate learning goals, Identify learning resources, Select and implement learning strategies and Evaluate learning outcomes. Further user study is needed to evaluate how well this interface can support learning.

Acknowledgements

This research was supported in part by the National Science Foundation under Grant No. 0447083

MinEr Liang is currently a Visiting Scholar at the School of Information Science, University of Pittsburgh. She would like to thank the Faculty, Staff, and Colleagues of the school for their support during the work reported in this paper. This work was supported by a grant from the National Natural Science Foundation of China (No. 60973026), the Shanghai Leading Academic Discipline Project (project number B114), and the Shanghai Committee of Science and Technology (nos. 08DZ2271800 and 09DZ2272800).

Julio Guerra is supported by Chilean Scholarship (Becas Chile) from the National Commission for Science Research and Technology (CONICYT, Chile) and the Universidad Austral de Chile.

References:

1. Arakawa, K., Tamaki , S., Kono, N., Kido, N., Ikegami, K., Ogawa, R., Tomita, M.: Genome Projector: zoomable genome map with multiple views. BMC bioinformatics, 10(1):31, 2009.
2. Bull, S.: Supporting learning with open learner models. In: Proceedings of 4th Hellenic Conference on Information and Communication Technologies in Education, Athens, Greece, September 29-October 3, pp. 47–61. (2004).
3. Bruls, Mark; Huizing, Kees; van Wijk, Jarke J.: "Squarified treemaps", in de Leeuw, W.; van Liere, R., Data Visualization 2000: Proc. Joint Eurographics and IEEE TCVG Symp. on Visualization, Springer-Verlag, pp. 33–42.
4. Brusilovsky, P. and Rizzo, R.: Map-Based Horizontal Navigation in Educational Hypertext. In: K. M. Anderson, S. Moulthrop and J. Blustein (eds.) Proceedings of 13th ACM Conference on Hypertext and Hypermedia (Hypertext 2002), College Park, MD, June 11-15, 2002, ACM, pp. 1-10.

5. Brusilovsky, P., Eklund, J., Schwarz, E.: Web-based education for all: A tool for developing adaptive courseware Computer Networks and ISDN Systems 30 1-7, 291-300. (1998)
6. Brusilovsky, P., Hsiao, I.-H., and Folajimi, Y.: QuizMap: Open Social Student Modeling and Adaptive Navigation Support with TreeMaps. In: C. Delgado Kloos, D. Gillet, R. M. Crespo García, F. Wild and M. Wolpers (eds.) Proceedings of 6th European Conference on Technology Enhanced Learning (ECTEL 2011), Palermo, Italy, September 20-23, 2011, Springer-Verlag, pp. 71-82.
7. Concept-map Navigator: http://www.khanacademy.org/exercisedashboard. KHAN Academy.
8. Falakmasir, M.H., Hsiao, I.-H., Mazzola, L., Brusilovsky, P.: The Impact of Social Visualization on Student Performance. In: Proceedings of ICALT2012, 2012.
9. G-language API: http://www.g-language.org/wiki/
10. Google Maps: http://en.wikipedia.org/wiki/Google Maps.
11. Hsiao, I.-H., Bakalov, F., Brusilovsky, P., and König-Ries, B.: Open Social Student Modeling: Visualizing Student Models with Parallel Introspective Views. In: Proceedings of 19th International Conference on User Modeling, Adaptation, and Personalization, UMAP 2011, Girona, Spain, July 11-15, 2011, Springer-Verlag, pp. 171-182.
12. Jianu,R, Demiralp, C., Laidlaw, D.H.: Visualizing and exploring tractograms via two-dimensional connectivity maps. In: Proceedings of ISMRM'10, 2010.
13. Jianu, R., Laidlaw, D.H.: Visualizing Protein Interaction Networks as Google.Maps. In: IEEE Visualization 2010 Poster Compendium, 2010.
14. Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. In: Proceeding Visual Languages, 1996. IEEE, pp. 336 - 343.
15. Shneiderman, B.: Treemaps for Space Constrained Visualization of Hierarchies: an historical summary of Treemap research and applications (2004), http://www.cs.umd.edu/hcil/treemaps/
16. Sosnovsky, S., Brusilovsky, P.: Layered Evaluation of Topic-Based Adaptation to Student Knowledge. In: Proceedings of Fourth Workshop on the Evaluation of Adaptive Systems at 10th International User Modeling Conference, UM 2005, July 26, 2005, pp. 47-56.
17. Specht, M., Oppermann, R.: ACE – adaptive courseware environment. The New Review of Hypermedia and Multimedia 1998.
18. The CUMULATE user model infrastructure: http://adapt2.sis.pitt.edu/wiki/CUMULATE
19. The Prefuse Visualization Toolkit: http://prefuse.org/
20. Yahoo! Content Analysis API: http://developer.yahoo.com/contentanalysis/