Using blogs as design primitives in implementing collaborative e-learning systems

Alexei Tretiakov and Roland Kaschek
Information Systems Group
College of Business, Massey University, New Zealand

Eyas El-Qawasmeh
Dept. Computer Science
Applied Science University, Jordan

We propose to use blogs as design primitives to enable interaction in e-learning systems. As a proof of concept, we implement a prototype of a concept mapping tool employing blogs attached to concepts to enable students to discuss these concepts in-place.

Keywords: blogs, design primitive, concept mapping, collaborative learning

Introduction

In the age of Web 1.0 the prevailing Web design primitives were web pages, links and web forms. Web users came to understand their behavior very well, and to form mental models of Web sites in their terms. With the emergence of Web 2.0 (O'Reilly, 2005), a new set of widely understood design primitives is getting established, in particular, blogs and RSS feeds. The structure of a blog - a series of postings by blog "owner", each followed by freely entered reader comments - is so well understood that a blog is perceived in its entirety, rather than as a collection of web pages with web forms connected by links.

Some of the well established educational paradigms, in particular, constructivist and situated learning, put a strong emphasis on interaction (both student-student and student-teacher) (Brookfield & Preskill, 2005; Soller, 2001; Van Merrienboer & Paas, 2003). Since blogs proved to be highly effective in fostering interaction (Axel & Jacobs, 2006), one could promote interaction by incorporating blogs at appropriate locations in an e-learning resource. Students would perceive each blog as a single primitive rather than as a complex structure, resulting in "smaller" mental models and hence, better usability, easier interactions, and, ultimately, better learning outcomes.

To trial the idea of using blog as a design primitive in e-learning systems, we implemented a prototype of a concept mapping tool employing blogs attached to concepts to enable sense making via collaborative discussions. The rest of the article is devoted to this tool.

Concept mapping tools

Concept maps are visual models structured as directed or undirected graphs. Concept maps and their variations, such as mind maps, proved to be useful as graphical knowledge representation tools for human users who are able to derive semantics from context. In particular, concept maps are extensively used in education, both to present knowledge to students, and to enable students to "construct" their own knowledge in a structured way (Concept Mapping, 1997).

A number of concept mapping tools are available (see, e.g., the comprehensive list maintained at http://www.mind-mapping.org). Some of these tools are open source, such as Conzilla (Knowledge Management Research Group, 2006). While some of the existing concept map tools support user collaboration (for example, the CmapTools software from Institute of Human and Machine Cognition (IHMC, 2006)), collaboration functionality they offer does not follow a familiar interaction paradigm (such as the interaction paradigm offered by blogs), and it is not used very widely.

Technological solution

We implemented the prototype by using the ActiveGrid environment (ActiveGrid, 2006), which allowed us to leverage model-driven development and to complete the implementation very quickly. We made use of DB2 Express-C DBMS from IBM to manage the application's data, because DB2 Express-C is the only currently freely available DBMS that implements the recursive query feature of the SQL99 standard allowing to formulate queries involving transitive closure. As it will be seen in subsequent sections, such queries are often useful to infer the consequences of beliefs represented in the form of conceptual maps.

To represent graph structure visually, via automatic layout, we employed the dot utility from Graphviz graph visualization open source project (Graphviz, 2006).

Prototype features

The prototype we implemented allows each student to maintain a directed acyclic graph of vertices and edges, with vertices representing concepts, and edges - relationships between concepts. Each edge is labeled by one of the available (provided by the system) relationship types, while vertices are untyped.

Vertices and their attributes / blogs

In the prototype, a student can add a new vertex. Vertices are global, in the sense that the identity of a vertex is determined by its name only, and not by the identity of its creator. A student can include a vertex in his own model (reflecting his perspective of the target domain) by either creating a new vertex or by importing a vertex from another student's model.

A blog is associated with each vertex. A student can attach posts to a vertex in his own model, or to a vertex in a model owned by any other student in the study group. Since vertices are global, such posts automatically become visible in all models that incorporate the vertex. Any number of posts can be attached to a given vertex. Post are ordered sequentially according to their creation time. No strict semantics is prescribed to a post, but we would expect that a post will either be an attribute of a vertex (some data which is part of the model), or reflect the student's opinion about the model. Such an opinion could be voiced as a message to another student. The model itself reflects the student's perspective of the world: it is not necessarily correct or complete, and correctness or completeness is not necessarily definable. Hence the difference between "an attribute" and "a student comment" is somewhat blurred. Moreover, comments posted by the same student at different points in time may contradict each other, reflecting a shifting position, or the construction of knowledge.

A post remains associated with its creator. Hence, a student who does not wish to take contributions by some or all other students into account can disregard their posts by hiding them.

Any student can attach any number of comments to any post (his own, or owned by any other student in the study group). Analogously to the semantics of posts, the semantics of comments is not strictly defined and is a matter of use. We expect that comments would either add to the data in the post, or contain metadata relevant to the data in the post, or describe the user's attitude with respect to the post and its context. Again, it is not possible to make sharp distinctions between these possibilities.

A vertex with its posts and their comments can be viewed as a multi-authored blog, and can in principle be published via RSS or Atom to make it available for syndication, as any other blog. Although current implementation does not provide RSS or Atom feeds for blogs, it would be very easy to add such a feature.

A user can remove a vertex from his model, even if it has posts attached to it, which does not cascade to remove the posts (which, hence, remain visible in any other models that include this vertex).

In a similar manner, a post or a comment can be removed by its creator. It is not possible to remove a post if it has comments attached to it. Hence, once a post was commented on by another user, consensus between these users is needed to remove the post.

This design ensures that user's contribution in any form can not be removed without the agreement of its creator, while, at the same time a user can ignore contributions by others.

Edges and their labels

A user can link vertices included in his model by creating directed edges between them. Each edge has to be labeled by using one of the available labels. Semantically, a label reflects the nature of the relationship between the two connected concepts (represented by vertices).

In the current implementation, labels available for labeling edges (link types / roles) are predefined by the teacher and can not be introduced by students themselves. We note that in principle, four different combinations are possible, as displayed in the following table:

Table 1: Student created / imposed vertices and edge labels - four possibilities

Vertex names	Edge labels
Student defined	Student defined
Imposed	Student defined
Student defined	Imposed
Imposed	Imposed

When both vertex names and edge labels are student defined, one would expect that although student creativity is unrestricted, it would be difficult to achieve cooperation between students, because the vocabulary of different students may not coincide or even intersect. On the other hand, by imposing both edges and vertex labels one would restrict the ability of the students to capture the semantics in terms of models. While depending on the nature of the problem to be explored the optimal approach may differ, in the current implementation we chose to allow students to introduce their own vertices (and thus to expand the domain), while imposing the edge labels available (and thus restricting the nature of the relationships to be considered). This constitutes a compromise between maximal creativity and maximal interoperability. We note also that certain edge labels may lead to problems with query semantics, which makes it difficult to allow students to define their own labels - this aspect is elaborated below in the "Queries" section.

To add an edge to his model, a student can either explicitly specify source and target vertices and the label, or alternatively, the student can import an edge from another student's model. A student can remove an edge from his model with no restrictions.

Queries

Current prototype implements only two types of queries:

1. List vertices directly connected by edges to the current vertex, and
2. List vertices connected to the current vertex over any number of edges with the same label.

While the semantics of query 1) is clear for as long as the semantics of edge labels is clear, for query 2) one needs to assume some sort of transitivity. Say, if A "contains" B, and B "contains" C, and it is very plausible to assert that A "contains" C, so in this case query 2) is clearly appropriate. If A "likes" B, and B "likes" C, than concluding that A "likes" C is not as certain as in the case of "contains", but still somewhat plausible (e.g., in the sense of potentiality). Query 2) then may still be appropriate, depending on the nature of the problem under investigation (it should be noted that connecting to a "friend of a friend" is a principle frequently employed by social networking applications). On the other hand, A "hates" B, and B "hates" C, in any reasonable interpretation, would not imply that A "hates" C (indeed, one could plausibly argue the opposite). In this case, query 2) clearly does not have any useful semantics.

In the prototype, the edge labels available for use are predefined; hence, we just make sure that the labels that are available are the ones for which query 2) results in an outcome with a reasonable semantics.

Semantic problems also emerge if loops are allowed in the graph. Say, if A is "more expensive than" B, B is "more expensive than " C, and C is "more expensive than" A, calculating transitive closure starting from A may lead a student to believe that A is the least expensive choice, while clearly the structure is such that, with no further information available, all choices are equally good. To avoid such situations, in the current prototype we do not allow loops in the graph. Hence, in the current prototype, all graphs are directed acyclic graphs (directed graphs with no loops), and edges labeled the same can be viewed as defining a partial order on vertices.

Identity, security and collaboration

The security model used by the prototype ensures that each student is able both to create his own models independently from other members of the study group, and to collaborate with other members of the study group by commenting on their models, by receiving comments, and by importing parts of their models into his model. This allows us to accommodate a wide range of collaboration strategies, from individual work (no collaboration), to clear-cut division of roles, with each student creating a well-defined part of a model, to be later integrated into a group's model. Collaboration strategies in between these two extremes, involving several students trying to represent largely the same concepts and relationships while discussing them to negotiate meaning in the process of construction are particularly effective at enabling learning (Gunawardena, Lowe, & Anderson, 1997). (Here, we ordered collaboration strategies according to the degree to which each of the participants is aware of / responsible for the final outcome.)

The prototype uses the relevant features of ActiveGrid to provide logon security. However, each student, after logging on, can access the model of any other student in the study group.
When accessing his own model, the student can add vertices and edges, and can add posts to vertices and comments to posts. The student can also remove vertices and edges from his model (all vertices and edges in the model, by definition, have the same owner as the model itself), and remove his own comments and remove his own posts that have no comments (see the discussion on removing commented posts in the "Vertices and their attributes / blogs" section above).

When accessing another student's model, the student can add posts to vertices, and comments to posts, but he can not add vertices or edges. Also, as when accessing his own model, the student can remove his own comments, and remove his own posts for as long as they have no comments.

In addition, when accessing another student's model, for each vertex or edge that is not present in the student's own model, the student is offered an opportunity to import it (an Import button shows in the user interface).

Example of use

Figure 1 shows a visual representation of a model created by student Brown, as shown by the prototype. It corresponds to a scenario in which Brown uses the prototype to record his beliefs about relative merits of various holiday destinations, and to explore options for choosing a holiday destination based on these beliefs.

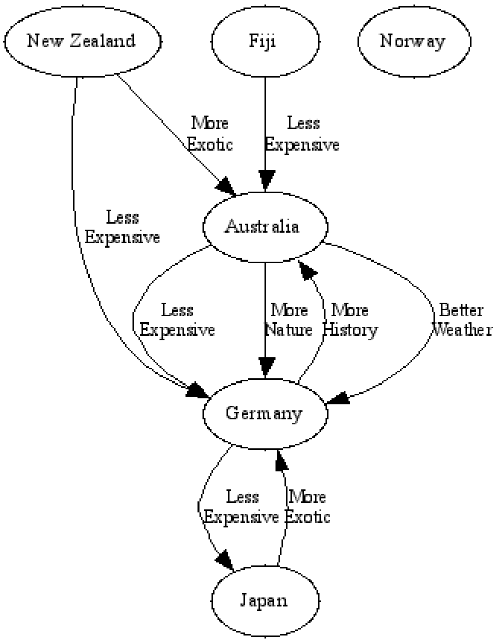
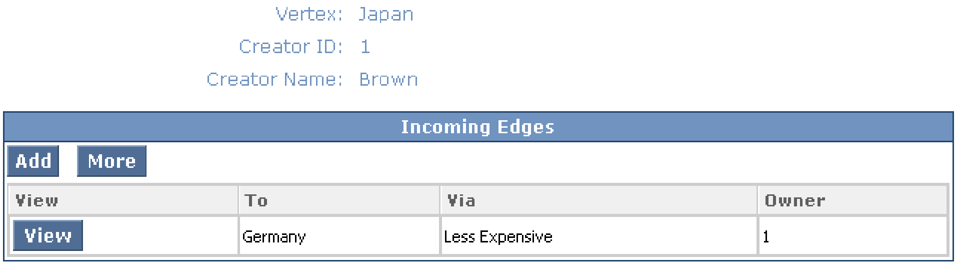


Figure 1: Graphical representation of a model - created in the prototype using dot

In Figure 1, the direction of edges (depicted as arrows) corresponds to the word order in English sentences depicting the corresponding relationships, for example "Fiji is less expensive than Australia" or "Japan is more exotic than Germany". The types of relationships included in the example are such that transitive closure queries are meaningful. For example "Fiji is less expensive than Australia" and "Australia is less expensive than Germany" lead one to conclude that "Fiji is less expensive than Germany".

Figure 2 shows the outcome of query 1) for Japan. This view is obtained simply by clicking on the name of a vertex in a list of vertices added to the model.

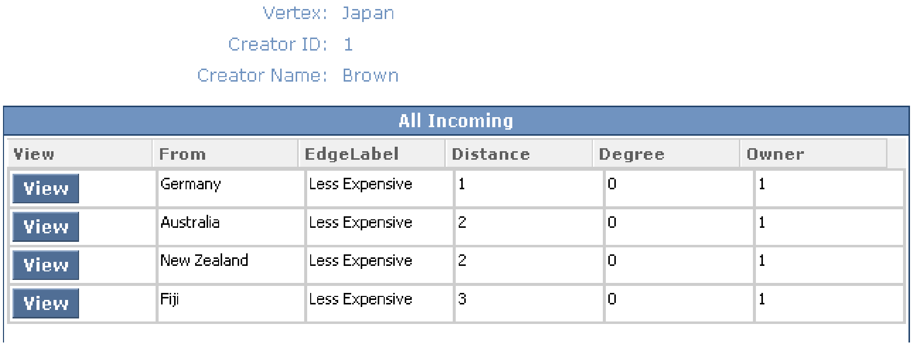


Vertex: Japan
Creator ID: 1
Creator Name: Brown

View	To	Via	Owner
View	Germany	Less Expensive	1

Figure 2: The outcome of query 1) for Japan

To view the outcome of query 2), Brown would click on the More button, resulting in the screen shown in Figure 3.

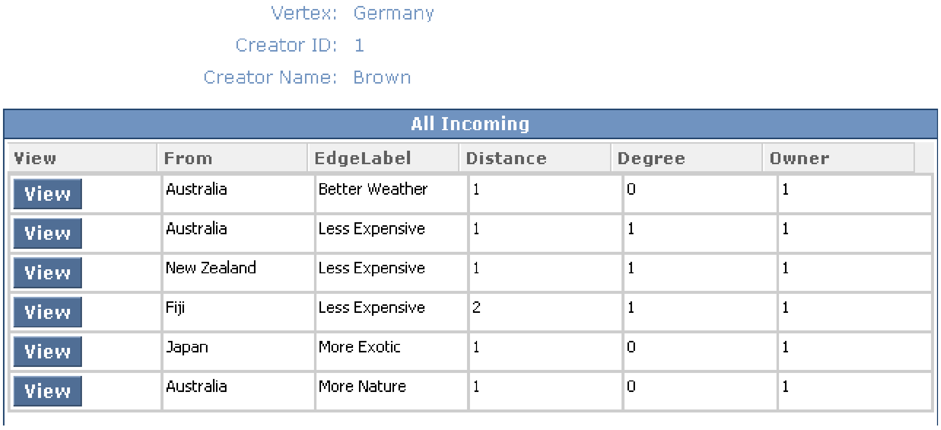


Vertex: Japan
Creator ID: 1
Creator Name: Brown

View	From	EdgeLabel	Distance	Degree	Owner
View	Germany	Less Expensive	1	0	1
View	Australia	Less Expensive	2	0	1
View	New Zealand	Less Expensive	2	0	1
View	Fiji	Less Expensive	3	0	1

Figure 3: The outcome of query 2) for Japan

In both query 1) screen and in query 2) screen one can activate View for any of the listed vertices to navigate to the query 1) screen for the corresponding vertex (which can then be refined to a query 2) screen by pressing More). Hence, the structure of the model is experienced by the student as the structure of a web site - something that most users nowadays find easy to conceptualize.



Vertex: Germany
Creator ID: 1
Creator Name: Brown

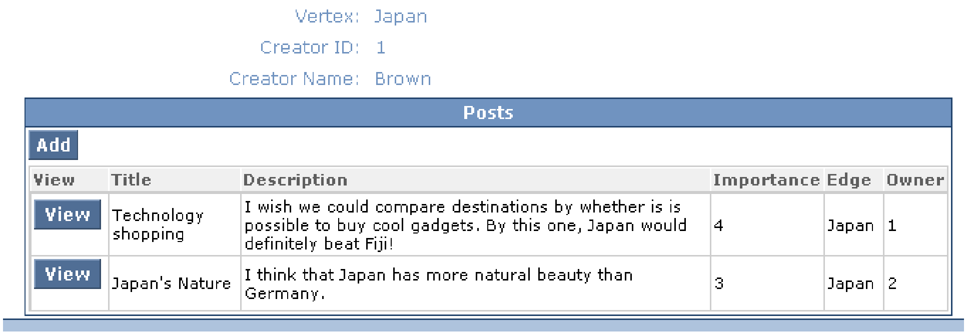
View	From	EdgeLabel	Distance	Degree	Owner
View	Australia	Better Weather	1	0	1
View	Australia	Less Expensive	1	1	1
View	New Zealand	Less Expensive	1	1	1
View	Fiji	Less Expensive	2	1	1
View	Japan	More Exotic	1	0	1
View	Australia	More Nature	1	0	1

Figure 4: The outcome of query 2) for Germany

Both Figure 2 and Figure 3 are clearly relevant to the problem of choosing a suitable holiday destination. Both figures display alternatives to the currently considered destination that are "better" in some respect (in this case - less expensive). The peculiarity of Figure 3 is that it displays not only the direct beliefs, but also their consequences, thus providing a holistic picture. Brown may not have realized that separately acquired beliefs on how different destinations compare to each other actually imply that Fiji is less expensive than Japan. In complex cases, such a holistic representation may lead one to re-examine the original beliefs and to reflect it in the structure of the model by adding and/or removing suitably labeled edges.

The prototype enables exploration along several dimensions at the same time, as it is illustrated in Figure 4. For Germany, less expensive, more exotic etc. alternatives can be explored.

Figure 5 shows the blog associated with Japan. Note that one of the postings is by a student other than Brown (it is by Smith with user ID 2). Smith added this post by logging on as himself, and then accessing Brown's model.



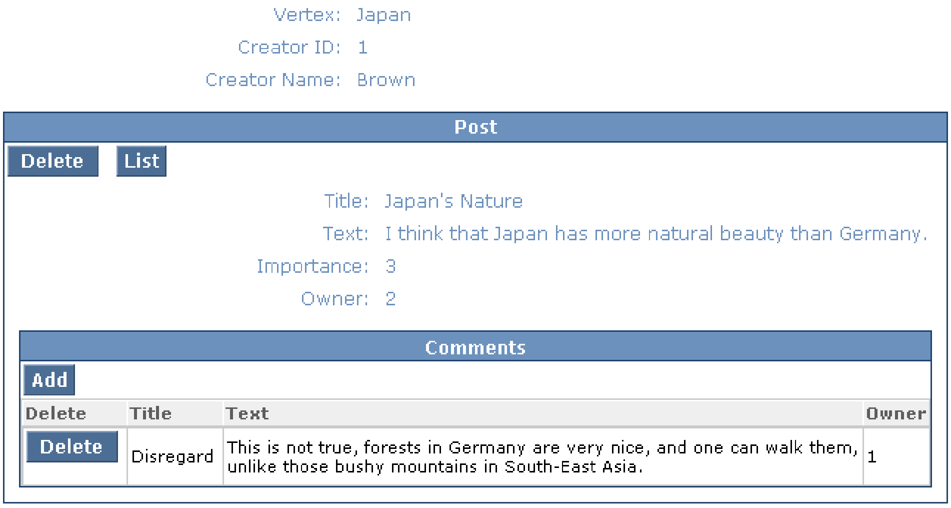
Vertex: Japan
Creator ID: 1
Creator Name: Brown

Posts

View	Title	Description	Importance	Edge	Owner
View	Technology shopping	I wish we could compare destinations by whether is is possible to buy cool gadgets. By this one, Japan would definitely beat Fiji!	4	Japan	1
View	Japan's Nature	I think that Japan has more natural beauty than Germany.	3	Japan	2

Figure 5: Blog associated with Japan

Figure 6 shows comments to the "Japan's Nature" post. Apparently, Brown rebuffed the suggestion made by Smith.



Vertex: Japan
Creator ID: 1
Creator Name: Brown

Post

Title: Japan's Nature
Text: I think that Japan has more natural beauty than Germany.
Importance: 3
Owner: 2

Comments

Delete	Title	Text	Owner
Delete	Disregard	This is not true, forests in Germany are very nice, and one can walk them, unlike those bushy mountains in South-East Asia.	1

Figure 6: Details of the "Japan's nature" post for Japan

Smith, however, can now create his own model in which Japan has "More Nature" then Germany, and then invite Brown to view it by making a relevant posting in the Japan blog.

Conclusions and future work

We demonstrated the feasibility of implementing a complex cooperative e-learning environment using blogs as design primitives employed to enable interaction in-place.

Currently, work is underway to conduct a formal evaluation with students in a real course.

Apart from assessing the usability of blogs employed as design primitives, it is of interest to determine the prevailing patterns of use of blogs attached to model elements: are they going to be used to represent "data" - to extend models, or only as "meta data" - to collaboratively discuss and to evaluate models?

References

ActiveGrid (2006). *ActiveGrid - Enterprise Web 2.0.* http://www.activegrid.com/

Axel, B. & Jacobs, J. (2006). *Uses of Blogs.* Peter Lang, 2006.

Brookfield, S.D. & Preskill, S. (2005). *Discussion as a way of teaching: tools and techniques for democratic classrooms.* Jossey-Bass.

Graphviz (2006). *Graphviz - Graph Visualization Software.* http://www.graphviz.org/Documentation.php.

Gunawardena, C.N., Lowe, C.A., & Anderson, T. (1997). Analysis of a Global Online Debate and the Development of an Interaction Analysis Model for Examining Social Construction of Knowledge in Computer Conferencing. *Journal of Educational Computing Research*, 17, 397-431. https://doi.org/10.2190/7MQV-X9UJ-C7Q3-NRAG

IHMC (2006). *IHMC CmapTools.* http://cmap.ihmc.us.

Knowledge Management Research Group (2006). *Conzilla - The Concept Browser*. http://www.conzilla.org/wiki?n=Overview.Main.

O'Reilly, T. (2005). *What Is Web 2.0 - Design Patterns and Business Models for the Next Generation of Software.* http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html. Soller,

A. (2001). Supporting social interaction in an intelligent collaborative learning system. *International Journal of Artificial Intelligence in Education*, 12, 40–62.

Special Double Issue: Concept Mapping (1997). *Journal of Interactive Learning Research,* 8 (entire issue).

Van Merrienboer, J. J. G. & Paas, F. (2003). Powerful learning and the many faces of instructional design: towards a framework for the design of powerful learning environments. In *Powerful learning environments: Unraveling basic computers and dimensions*. Elsevier Science.

Please cite as: Tretiakov, A., Kaschek, R. & El-Qawasmeh, E. (2007). Using blogs as design primitives in implementing collaborative e-learning systems. In *ICT: Providing choices for learners and learning. Proceedings ascilite Singapore 2007*. https://doi.org/10.65106/apubs.2007.2587