A Framework for Cross-Platform Graph-based Recommendations for TEL

Mojisola Anjorin[1], Ivan Dackiewicz[2], Alejandro Fernández[2], and Christoph Rensing[1]

[1] Multimedia Communications Lab,
Technische Universität Darmstadt, Germany
{mojisola.anjorin,christoph.rensing}@kom.tu-darmstadt.de
[2] LIFIA-UNLP, La Plata, Argentina
{idackiewicz,alejandro.fernandez}@lifia.info.unlp.edu.ar

Abstract. A Technology Enhanced Learning (TEL) ecosystem is a kind of Digital Ecosystem formed by independent platforms combined and used by learners to support their learning. We believe that recommendations made across these different platforms by exploiting the synergies between them will benefit learners. However, building such cross-platform recommender systems poses new and unique challenges for developers. In this paper, we present a framework to support the development of cross-platform recommender systems for TEL ecosystems and discuss challenges faced. The framework decouples the development of the recommender system from the evolution of the specific platforms by combining graph-based algorithms, a unified data model, and a service oriented architecture. As proof of concept, the framework was effectively applied to develop a cross-platform recommender system in a TEL ecosystem having Moodle as the Learning Management System, Mahara as the Social Networking Service and Ariadne as Learning Object Repository.

Keywords: TEL, Recommender Systems, Cross-Platform, Framework

1 Introduction

A Technology Enhanced Learning (TEL) ecosystem, is a form of a Digital Ecosystem [2] inhabited by elements from various platforms used in parallel by learners and teachers. Such a simultaneous use of platforms is often found in communities of practice [9] also known as learning networks, where learning is mostly self-directed. In this paper, we focus on a TEL ecosystem with three platforms: a Learning Management Systems (LMS), a Social Networking Service (SNS), and a Learning Object Repository (LOR). An LMS offers activities as well as discussion forums and shared spaces such as wikis. Activities rely on learning objects (LOs) such as lesson notes and presentations. The visibility of a LO is normally limited to an activity. However when an LMS is used to support self-directed learning, it becomes particularly important that learners are aware of all activities, resources and peers they could potentially gain from. Nowadays, many learners participate in social networks connecting to other learners via Facebook[3], or posting learning tasks and following other learners on Twitter[4]. Contacts the students have on platforms such as an LMS are disconnected from the online social networks they belong to outside the classroom. It is therefore up to the students to replicate in each of these worlds the relationships they have built in the other. The potential to share knowledge and find valuable contacts across these platforms therefore remains unexploited. Initiatives such as the MIT OpenCourseWare[5] or the Ariadne Foundation[6] with its LOR demonstrate the increasing interest in collecting and sharing high quality learning material. LORs however are isolated from the LMS and SNS. There therefore exists an opportunity to provide learners with information across multiple platforms by considering the synergies between them.

In the following sections we propose a framework to empower a TEL ecosystem by generating cross-platform recommendations in each of them based on resources gained from all of them.

2 Related Work

Recommender systems based on approaches such as content based and collaborative filtering (CF) techniques have been shown to be very useful in TEL scenarios, especially in informal learning [8]. CF approaches use community data such as feedback or ratings from other users to make recommendations. Graph-based recommender techniques can be classified as neighborhood-based CF approaches [4]. A graph is used to represent the users or items as nodes and the edges as the transactions between them. PageRank [3] is an example of a graph-based approach based on a random walk similarity. Transitive associations are defined within a probabilistic framework where the similarity or affinity between nodes is calculated as a probability of reaching these nodes in a random walk on a weighted graph having a node for each state. The probability of jumping from one node to another is given by the weight of the edge connecting these nodes. In this paper, we implement PageRank using the information from the platforms that make up the ecosystem, to generate recommendations across them.

ReMashed [5] is a Mash-up Personal Learning Environment allowing learners to combine content from different Web 2.0 services to a personal view or mash-up. Learning resources are recommended using a CF approach that matches users with similar opinions and considers the learning goals of the learner. In contrast, we propose a framework to recommend activities, users and LOs across multiple platforms, thus pointing the learners to other valuable sources of information found on these different platforms without building a mash-up.

Recommender systems are often implemented as closed, internal components of larger applications having tightly coupled components. In contrast, APOS-

[3] http://facebook.com (last retrieved 30.06.2012)
[4] http://twitter.com (last retrieved 10.07.2012)
[5] http://ocw.mit.edu/index.htm (last retrieved 10.07.2012)
[6] http://www.ariadne-eu.org/ (last retrieved 10.07.2012)

DLE [1] for example, follows the SOA approach providing web services to publish knowledgeable person recommendations. Web services decouple the generation of recommendations from its presentation to the users. Our framework uses a similar approach. Furthermore, graph-based approaches are suitable for integrating data from various platforms, using the graph as the grounds for inter-operation. This is particularly interesting when combined with vocabularies and technologies that originate in the Semantic Web and Linked Open Data movements [6].

3 Cross-Platform Recommendation Framework

The framework is shown in Fig.1 where the TEL ecosystem comprises of an LMS, an SNS and a LOR. These platforms are independent of each other and have been implemented autonomously. The introduction of recommendations should neither increase coupling between these platforms, nor require intrusive changes that will hinder their maintenance. Moreover, the choice of platforms to be integrated must remain flexible, allowing for new alternatives to be introduced as a replacement for any of them or as a complement (i.e., there could be more than one LMS, SNS or LOR). To provide recommendations in such a TEL ecosystem, our framework adopts a service oriented architecture. The *Recommender* in Fig.1 is implemented as an independent component. It provides a parameterizable implementation of a graph-based recommender algorithm (1). The algorithm takes as input a graph with nodes representing items in each of the platforms and links representing relationships between them (2). The values given to the nodes and the weights for the edges influence how the algorithm ranks the elements. A service publishes a function that the platforms can call to retrieve recommendations (3). All changes in the platforms that are relevant to compute recommendations (i.e., to build the graph) are communicated (4) and stored by the recommender in its data model (5). The data model is also the basis for exchanging relevant data between the platforms and the recommender. Finally, there is a mapping (6) to generate the graph (i.e., the nodes and edges) from the data model. The mapping allows for the introduction of links that did not exist in the data model (e.g., links connecting semantically similar resources or links that connect users belonging to the same group).

The User Interface and Recommendation Lists: From the user's perspective, each platform introduces a recommendation list to the User Interface (UI) component. In Fig.2, the recommendation list is shown in Mahara (left side) and in Moodle (right side). The recommendations are personalized considering the user's current focus. For example, in Fig.2 recommendations are provided in Mahara for the user Albert Alonso taking into account that he is currently focused on viewing Bernard Berazategui's user profile. Depending on the recommendation strategy, the recommendation lists might include other users that Bernard has befriended, activities that he has completed, and resources that he frequently uses. Consequently, the recommendation lists contain items from any of the three integrated platforms: Activities, LOs and Users.

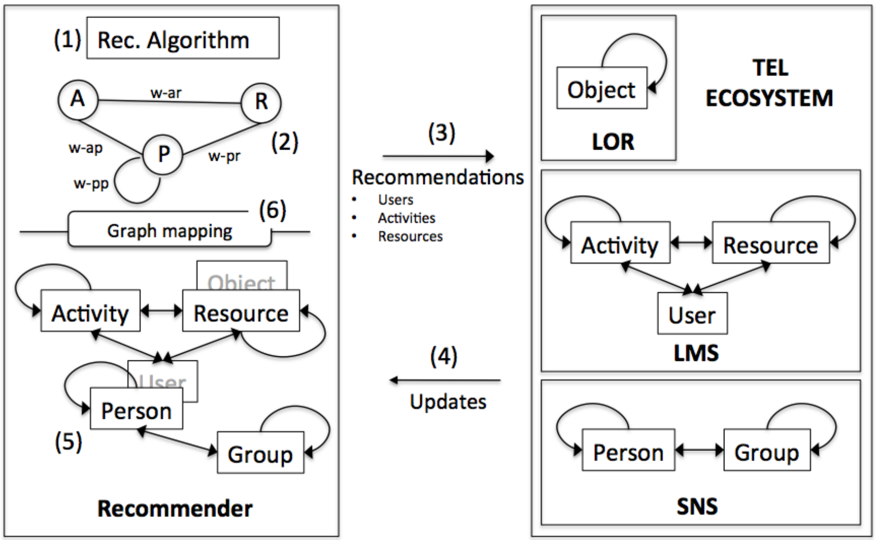

Fig. 1. Overview of the Cross-Platform Recommendations Framework

The Service Oriented Approach: The recommender component implements five core web-services. New resources are added through an addResource() service, taking as argument the unique identifier (URI) of the object. Attributes of the object and relationships are added/updated through calls to updateDataAttribute() and updateObjectAttribute() respectively. To retrieve recommendations, clients call the getRecommendations() service indicating the user and his current focus (a specific object). To encapsulate the development of the recommendations in the UI components (thus reducing coupling between these components and the rest of the functionality of the platforms) we follow a plug-in approach. Most open platforms support a plug-in extension mechanism. Our framework provides an interface that plug-ins can invoke to implement operations to display, register and handle events that correspond to changes to any of the relevant objects on the platform. Each platform is required to implement the recommender UI element as a plug-in component, ofcourse, depending on the platform, this can pose an implementation challenge.

The Data Model and Data Mapping: The data model serves two key purposes: First, it is used to create the graphs that feed the recommender algorithm. Second, it provides basic information about the objects that each of the platforms displays to the user. This approach has to remain generic enough to accommodate not only the platforms that we choose for the proof of concept (Moodle, Mahara and Ariadne) but other alternatives as well. The data model is stored in the form of triples. Each object has a unique id (a URI). Relationships between objects (objectURI, relationship, subjectURI) as well as object attributes (objectURI, attribute, value) are stored as triples. A certain object attribute relates

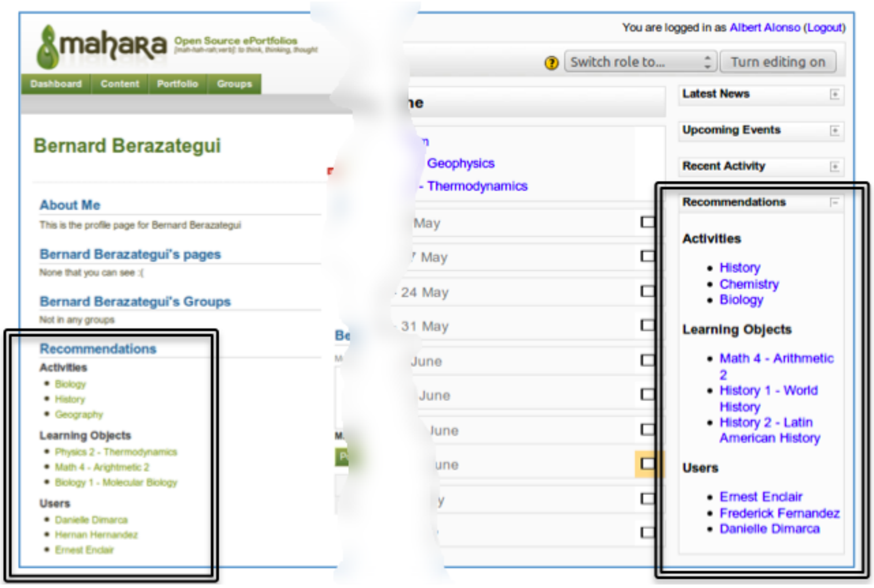

Fig. 2. Recommendation of activities, learning objects and users in Mahara and Moodle

the object to its type (e.g, a user object to the URI of the Person type). The data model aggregates information that would otherwise be disconnected, e.g, it connects LOs from the LOR to users and activities in the LMS. Therefore the definition of a common unique identifier (e.g, primary email for persons) across all platforms is needed to uniformly identify objects that are present on the different platforms, and become one in the data model. A challenge here is considering the access rights the user has in each system in order to only recommend objects the user is allowed to view. A common user authentication like single sign-on could be a solution.

The Recommender Algorithm: In this implementation, we choose the PageRank algorithm on the graph to produce a ranking of nodes. This is implemented using the JUNG (Java Universal Network/Graph) framework [7]. This ranking is the basis for the recommendation lists that are returned to clients. A graph mapping strategy generates the graph from the data model. First, it generates a node for each object (i.e., Persons, Activities and Resources). Nodes have values (e.g., the probability of reaching the node after a random jump) and the URI of the object they represent. A node's value is set in a way that increases the impact it has in the resulting ranking, e.g. the node representing the user or the object in focus starts with a higher weight. Then, the mapping strategy generates edges. The weight given to each type of edge can be configured to give certain connections higher relevance. In the current implementation, the relationships considered are user - user, user - resource, user - activity and activity - resource. The weights are calculated as the average number of relationships between the different types of nodes i.e. the number of resources accessed by the user/ the number of resources that have been accessed by any user. In the initial experiment, about 80 relationships are considered between 12 users, 15 LOs and 6 activities.

4 Conclusion

In this paper, we propose to take advantage of the synergies that arise across multiple platforms in order to generate cross-platform recommendations in a TEL ecosystem, aiming to further enhance the learning effort of the learners. Focusing on graph-based recommendations, we discussed design and implementation challenges. Providing effective recommendations requires experimenting with different platform combinations, and graph configurations. To ease the development efforts, we propose a framework to provide recommendations in a TEL ecosystem. As a proof of concept and to demonstrate the flexibility of such a framework, an implementation was made with Moodle as LMS, Mahara as SNS and Ariadne as LOR. Future work will be to integrate additional platforms in a different constellation of a TEL ecosystem and to conduct a usability study to evaluate the recommender algorithms used in the framework.

References

1. Beham, G., Kump, B., Ley, T., Lindstaedt, S.: Recommending knowledgeable people in a work-integrated learning system. Procedia Computer Science (2010)
2. Boley, H., Chang, E.: Digital Ecosystems: Principles and Semantics. In: Inaugural IEEE Int. Conf. on Digital Ecosystems and Technologies. Cairns, Australia (2007)
3. Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems 30, 107–117 (1998)
4. Desrosiers, C., G., K.: A Comprehensive Survey of Neighborhood-Based Recommendation Methods. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P. (eds.) Recommender Systems Handbook, pp. 107–144. Springer (2011)
5. Drachsler, H., Pecceu, D., Arts, T., Hutten, E., Rutledge, L., van Rosmalen, P., Hummel, H.G.K., Koper, R.: Remashed - recommendations for mash-up personal learning environments. In: Cress, U., Dimitrova, V., Specht, M. (eds.) EC-TEL. LNCS, vol. 5794, pp. 788–793. Springer (2009)
6. Fernández-Tobías, I., Cantador, I., Kaminskas, M., Ricci, F.: A generic semantic-based framework for cross-domain recommendation. In: Proc. of the 2nd Int. Workshop on Information Heterogeneity and Fusion in Recommender Systems. pp. 25–32. HetRec '11, ACM, New York, NY, USA (2011)
7. Madadhain, J., Fisher, D., Smyth, P., White, S., Boey, Y.: Analysis and visualization of network data using jung. Journal of Statistical Software 10, 1–35 (2005)
8. Manouselis, N., Drachsler, H., Vuorikari, R., Hummel, H., Koper, R.: Recommender Systems in Technology Enhanced Learning. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P. (eds.) Recommender Systems Handbook, pp. 387–415. Springer (2011)
9. Wenger, E.: Communities of Practice: Learning, Meaning, and Identity. CUP (1998)