

# Educational Social Linking in Example Authoring

I-Han Hsiao  
School of Information Sciences,  
University of Pittsburgh  
135 N. Bellefield Ave.,  
Pittsburgh, PA 15260 USA  
+ 1 412 624 9437  
ihh4@pitt.edu

Qi Li  
School of Information Sciences,  
University of Pittsburgh  
135 N. Bellefield Ave.,  
Pittsburgh, PA 15260 USA  
+ 1 412 624 7666  
qil14@pitt.edu

Yi-Ling Lin  
School of Information Sciences,  
University of Pittsburgh  
135 N. Bellefield Ave.,  
Pittsburgh, PA 15260 USA  
+ 1 412 641 9990  
yil54@pitt.edu

## ABSTRACT

Each educational resource management website relies on an authoring tool to provide example content. It takes time and experience for authors to create valuable content. Providing support during authoring can affect the quality and quantity of the examples. In this paper we propose a mashup solution to automatically link community wisdom to authors and ease various difficulties in authoring.

**Categories and Subject Descriptors:** H.0 General

## General Terms

Design, Reliability, Experimentation, Human Factors

## Keywords

Social Linking, Example Authoring, Mashup, Wikibooks.

## 1. INTRODUCTION

Each educational resource management website relies on an authoring tool at the backend to provide content. The authoring tools are usually tailored to specific systems and require experienced authors or well-trained editors to provide valuable content. Most of the time they require authors to use APIs or other references at hand to do the authoring. Inexperienced authors have to spend a lot of time to find resources for authoring quality examples. In general, authors may search the internet for finding similar examples and reusing the partial codes. This can be a very time consuming process. It is also important to be able to validate examples found on the internet which requires a validation functionality for the authoring tools. The availability of the similar examples, the easiness of retrieval and the relevance affect the developing process and the quality and quantity of the examples. We hypothesize that providing relevant examples during authoring phase will facilitate the process of example authoring and help to author higher quality examples. To address this issue, we propose a mashup solution to automatically link community wisdom to authors and ease various difficulties in authoring. Each mashup module dynamically pulls relevant examples for the users. We aim to utilize the flexibility of mashups to increase the value of authoring tools. The following sections present the description of the proposed solution. We focus on a web-based integrated example authoring environment in java programming language.

Copyright is held by the author/owner(s).  
HT'08, June 19–21, 2008, Pittsburgh, Pennsylvania, USA.  
ACM 978-1-59593-985-2/08/06.

## 2. TECHNOLOGIES OF EDULINK

EduLINK is an educational social linking system for example authoring. It's a web-based authoring system with dynamic authoring support. Figure 1 presents an example screen of the system. The left side of the screen is the authoring panel. Each mashup module is presented into a tab page on the right hand side. The authoring panel is designed in the notepad format. It allows users to author the code as keeping a note. It removes the burdens from authors to spend time on getting used to different tailored authoring interfaces, for instance, line by line editing or unchangeable class name or variables etc. Each mashup module presents related example codes and source links.

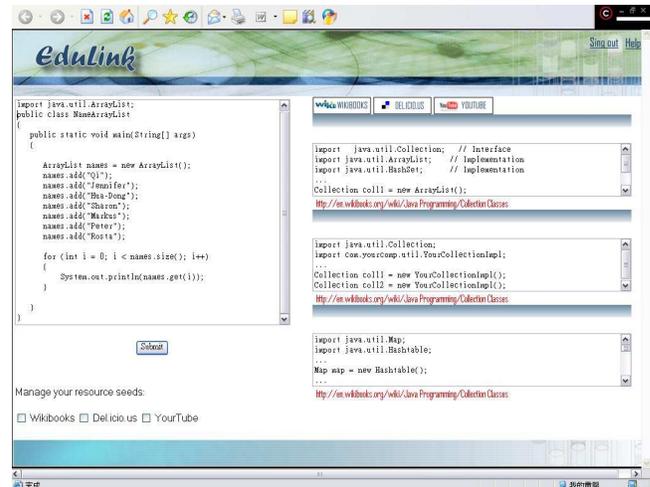


Figure 1 EduLINK Interface

### 2.1 Process Flow

The input of EduLINK is directly from users' example source codes in the authoring panel. The system includes a parser engine which performs two tasks. First, it examines the syntax of the example and catches the syntax error in the provided examples, if there's any. We define the caught concepts as KC (Key Concepts). The key concepts will be passed to each mashup module. Second, the parser extracts a list of concepts from the entire example, regardless the completeness of the example. We define the list of concepts as AC (Analyzed Concepts). AC will also be passed to each mashup module. We use the java ontology to parse the concepts, which was developed in [1]. It's initially created to describe the content of the Sun Java Tutorial and can be accessed at [http://personal-reader.de/rd/f/java\\_ontology.rdf](http://personal-reader.de/rd/f/java_ontology.rdf). Each mashup module performs three retrieval tasks. First, they get the KC from the parser engine and proceeds mining on each web

application of the mashup module. Second, each entry of results will be indexed by each on-site mashup parser. Third, a ranked example codes will be returned to users by calculating the cosine similarity with the list of concepts generated and the AC derived

## 2.2 Mashup Modules

We use three well-known websites, Wikibooks, Delicious and YouTube as the resources. Data on these resources are socially tagged resources by community of users. We elicit data from these websites instead of generic search from search engines. Each module will be discussed in the following sections of the paper. The system architecture (Figure 2) illustrates the data flow and corresponding activities.

### 2.2.1 Wikibooks

Wikibooks [2] is a free content textbook and manual based on Wikimedia foundation wiki sets. It began on July 10, 2003, since then Wikibooks has grown to include over 28,678 pages, in more than 10 languages. Wikibooks has a complete Java book including plenty of Java source codes with detailed commands and usage explanations. It is a good source to provide the Java programming examples. Wikibooks Sample Codes Extracting Module (Wikibooks Module) is focused on extracting the examples codes according to the author editing Java source codes.

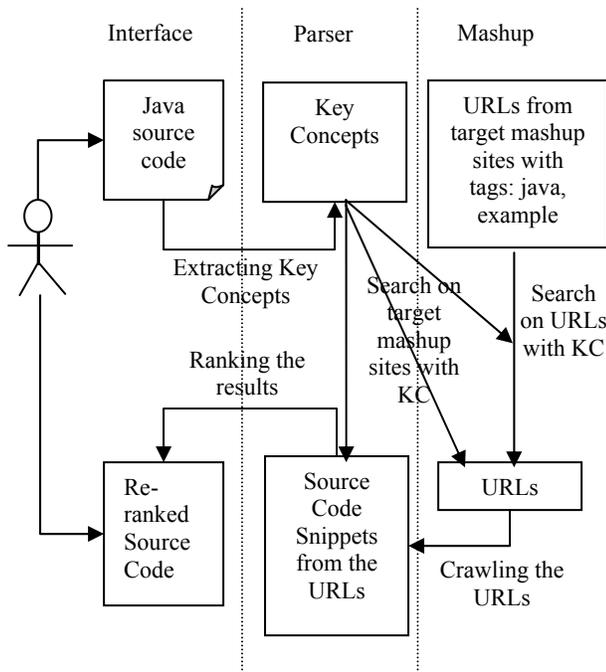


Figure 2 System Architecture

### 2.2.2 Del.icio.us

Del.icio.us is a collaborative tagging system for web bookmarks [3][4]. It started in late 2003 and currently has more than 3 million users and 100 million bookmarked URLs [5]. We discover the URLs contained “java” and “example” tags are mostly the links to java example code collections. Fewer users tag their URLs specifically anchored to an example. Therefore, in this mashup module, we grab the top 10 java example collection websites from

from users. EduLINK will provide results for users according to every authoring validation or submission. Iterations of results are expected to assist users in example authoring.

del.icio.us and combine the key concepts to locate the example code snippets.

### 2.2.3 YouTube

YouTube is a video sharing website. It contains 77.3 million videos and 2.89 million users since 2005 [6]. This module follows similar procedure as the way we mash del.icio.us site. We retrieve a stream of video clips tagged with terms such as “java” or “tutorial”. Using Key Concepts and mining the video description and tags, we provide a narrowed-down list of videos.

## 3. DISCUSSION AND FUTURE WORK

We conducted a small pilot experiment with our project team members. The preliminary results show that examples of Wikibooks and del.icio.us are useful to support us in example authoring. YouTube also provides more concrete sense about the overall implementation. However, extracting video context is a challenge in our current design. Therefore, we do not do the video context mining in this study. From our experiences, we found that comparing current three mashup modules, YouTube is relatively less useful for users in direct authoring reference.

The project includes a controlled experiment which is ongoing. The experiment includes three predefined tasks and two arbitrary tasks. Predefined tasks are designed based on concepts and randomly assigned to subjects. Subjects will be given specific instructions to follow for authoring the examples. All of their activities will be recorded in our session logs. In arbitrary tasks, subjects can choose any concepts and design their examples. The authoring activities will be recorded too. The subjects are randomly assigned to control or experimental group. We expect to observe differences in the performance of the experimental group, the time spent, and their satisfaction. We also plan to collect more diverse and comprehensive examples from other online resources, such as Google Code Search etc. The comparison across each mashup module will also be reported.

## 4. REFERENCES

- [1] Henze, N., P. Dolog, and W. Nejdl, Reasoning and Ontologies for Personalized e-Learning in the Semantic Web. Educational Technology & Society, 2004. 7(4): p. 82-97.
- [2] Wikibooks: [http://en.wikibooks.org/wiki/Main\\_Page](http://en.wikibooks.org/wiki/Main_Page)
- [3] Golder, S. & Huberman, B.A. Usage patterns of collaborative tagging systems. Journal of Information Science, Vol. 32, No. 2, 198-208 (2006)
- [4] Del.icio.us: <http://del.icio.us/>
- [5] <http://www.techcrunch.com/2007/09/06/exclusive-screenshots-and-feature-overview-of-delicious-20-preview/> (Last Access: March26<sup>th</sup>2008)
- [6] YouTube: <http://www.youtube.com/>