

Knowledge Servers for the Classroom

Michael Wollowski, Kevin Ridsen

Rose-Hulman Institute of Technology, wollowski@rose-hulman.edu, risdenkj@rose-hulman.edu

Abstract - We describe a knowledge-server we developed for our data structures course. This server holds information traditionally contained in textbooks. Additionally, it houses information that is not easily contained in textbooks, such as animations and presentation slides of important concepts. It furthermore references materials from diverse sources available on the World Wide Web. In addition to course materials, the server makes available a variety of ways in which to navigate the information. In addition to searching concepts and a directory listing of concepts, it makes available a navigable graph, representing the ontology of the course materials. This enables students to more quickly find information and to better see relationships between materials introduced in the course. We see knowledge servers as taking the role of textbooks in the not so distant future.

Index Terms – Education, Knowledge Server, Semantic Web, Textbook.

INTRODUCTION

It is no secret that the physical textbook is on its way out. Most textbooks have companion web-sites containing slides, code, and exercises. Some even contain electronic copies of the textbook enabling students to quickly search for contents. While this is a natural next step, it does not take full advantage of the current status of computing technology.

In this paper, we describe an approach we have taken to advance the state of affairs. It is based on our own practice of maintaining a course web-site in which we maintain links to additional learning materials such as animations, our own slides and materials from Wikipedia®. We kept those materials but replaced a static schedule page with a server that enables students to navigate and search the materials in a variety of ways. Among others, they can navigate a graph, representing ontological relationships between concepts.

There has been a good amount of work arguing for the use of information technology in education. The popular press has been highlighting the benefits of semantic web technologies and suggested the use of digital media to replace printed textbooks [2, 4, 6, 9]. The Journal of Interactive Media in Education published a special issue entitled the “Educational Semantic Web” [3]. In it, authors argued the benefits of the semantic web in education. A similar argument was made in a workshop entitled “Reconsidering the Textbook” [1]. In [5], the authors foresee that “Higher learning will undergo a major restructuring

during the next couple of decades.” They see information technology as playing a major role in facilitating this process.

While knowledge servers seem an intuitive next step in the evolution of textbooks, concrete examples are hard to come by. Person® which owns Prentice Hall® and Addison Wesley® has companion sites for its textbooks that make slides and code from the books available. Through the web-site www.coursesmart.com, Pearson makes available electronic copies of their textbooks. In this paper we describe an approach that we feel takes textbooks to the next level and fits the mold suggested in the papers quoted above.

We would like to point out that this work is interested in advancing the state of the art of learning materials in the context of a bricks and mortar classroom. We are at this time not concerned about intelligent tutoring systems or similar learning environments.

APPROACH

Over the past several terms, we have continually revised the course materials of our Data Structures course, so that they contain resources to animations and materials from the web that some students find more appealing than that of the book. Additionally, we divided the information of some of our slide presentations into separately addressable chunks so that they can be explicitly referenced.

For example, for the subject matter of Red-Black Trees, our course web-site gives students access to the following kinds of materials:

- A definition
- Positive example
- Negative example
- Slides for bottom-up insertion and removal algorithms
- Slides for top-down insertion and removal algorithms
- An animation for bottom-up insertion and removal algorithms.
- An animation for top-down insertion and removal algorithms.
- A programming assignment in which students are asked to implement top-down Red Black trees.
- Unit test cases for the programming assignment.

Some of the materials come from the slides of the book we currently use. Additional materials come from Wikipedia or pages found on the Web. Some animations are housed on external servers; some are written by us and made available through our own servers. We believe that the diversity of the materials is a key benefit of a course materials server. It

helps to address differences in prior knowledge and learning styles. As we ask our students to bring lap-tops to class and have internet access in all of our classrooms, students do search the web for materials which they feel best explains the concepts covered in class. We support this behavior by asking students to email us links to materials they find useful and by making some of those links available to all students in the class.

All of those materials are available through static web-pages such as schedule pages, daily outlines and links to resources. As such, we have a key component of our newly developed system: the knowledge. While we are happy with the variety of course materials we offer, we feel that students do not always grasp the relationships that hold between the data structures covered in class. To address this issue, we developed a knowledge server which houses the information already contained on our course web-site but enables students to more conveniently access and find it. In addition to convenient search features, we developed an ontology of key concepts and display it as a navigable graph.

A first prototype of our system was developed as part of a class project two terms ago. My students and I developed an initial ontology and each group of students was asked to implement a portion of it as well as a rudimentary knowledge server. We since developed our system which we will describe next.

KNOWLEDGE REPRESENTATION

Our ontology is constructed with the semantic web in mind. While we do not use XML, our data can easily be converted to semantic web triples. We chose not to use semantic web syntax as it is cumbersome to edit without a graphical editor such as Protégé [7] and because there does not seem to be any semantic web annotated course materials with which we could interact. Figure 1 contains the portion of the ontology that relates to Red-black trees.

```
class, Data Structure
class, Tree
class, Binary Tree
class, Binary Search Tree
class, Balanced Binary Search Tree
class, Red Black Tree

subClassOf, Tree, Data Structure
subClassOf, Binary Tree, Tree
subClassOf, Binary Search Tree,
    Binary Tree
subClassOf, Balanced Binary Search Tree,
    Binary Search Tree
subClassOf, Red Black Tree,
    Balanced Binary Search Tree
```

FIGURE 1
REPRESENTATIVE SAMPLE OF ONTOLOGY.

The `class` triples define the concepts of our ontology and the `subClassOf` triples define sub-class relationships

among classes. For now, we use the concepts real names as identifiers, removing a layer of indirection in which real names have to be associated with more formal identifiers. While the class data are represented as pairs, we will refer to them as triples as the pairs notation for classes is an accepted abbreviation in RDFS/OWL. A triple is considered a statement, consisting of a resource, property, and property value. In our notation, the resource appears first, followed by the property and finally the value of the property.

In addition to the ontology, we created associations between concepts in the ontology and course materials to be displayed. Those associations too are represented through triples. For lack of a better term, we call them *instances*. A representative sample of instances can be found in figure 2 in which we display the data for Red Black trees. Notice that the triples contain addresses of course materials.

```
Red Black Tree,
    Definition,
    Red-Black-Tree-def.html
Red Black Tree,
    Example,
    Red-black-tree-exa.jpg
Red Black Tree,
    Bottom-up Insertion Algorithm,
    Red-Black-Trees-Insertion.pdf
Red Black Tree,
    Top-down Insertion Algorithm,
    Red-Black-Trees-Insertion.pdf
Red Black Tree,
    Bottom-up Insertion Applet,
    http://people.ksp.sk/~kuko/bak/big
Red Black Tree,
    Top-down Insertion Applet,
    http://www.institution.edu/class/
    csse/csse230/schedule/day13/
    RedBlackTreeApplet.html
Red Black Tree,
    Alternate name,
    RB Tree
```

FIGURE 2
REPRESENTATIVE SAMPLE OF INSTANCES.

SYSTEM OVERVIEW

Our knowledge server is implemented through a Java servlet and is served through an Apache Web Server.

We separated the ontology from the instance data and placed them into different files. This is current practice in semantic web research and we follow it. Should the need arise; both files can be combined into one. The course materials are located in several directories of our file system. The server produces the GUI on the fly, among others, including CSS and JavaScript code. The architecture of our system is shown in figure 3.

Upon start-up, the servlet reads the ontology file and processes it in several ways. For each `class` triple, an

instance of the `Entry` class is generated. The fields of that class are shown in figure 4. The entire string of the `class` triple is placed in the `keyword` field. Each instance is inserted into a hash table, using the entire contents of the `keyword` field as a key.

For each `subClassOf` triple, we locate the appropriate entry in the hash table. Should there be no entry in the hash table, we generate one. As such, the class information is unnecessary. However, we like our system to be follow good practices so as to quickly share the information with appropriate semantic web services, should the opportunity arise. We add the name of the super concept to an `ArrayList` named `subConceptOf` and generate a symmetric entry in the instance that is the super concept. The latter are stored in an `ArrayList` called `subConcepts`. Should the super concept not exist in the hash table, we create an entry for it too.

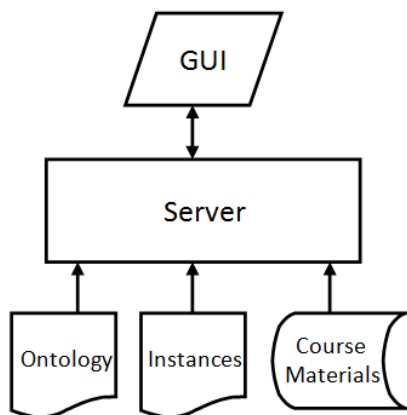


FIGURE 3
ARCHITECTURE OF OUR SYSTEM.

Once the sub-concepts have been determined, sibling information is computed and stored in a separate `ArrayList` called `siblings`.

Notice that the entries in the `ArrayLists` are of type `String`. We store the names of concepts rather than pointers to their instances, because the server needs to know the names when generating the HTML mark-up for the GUI. As such, we remove a level of indirection as indicated above. The instances can be quickly located in the hash-table.

```

private String keyword;
private ArrayList<String> alternateName;
private ArrayList<String> subConceptOf;
private ArrayList<String> subConcepts;
private ArrayList<String> siblings;
private ArrayList<ArrayList> properties;
  
```

FIGURE 4
FIELDS IN THE ENTRY CLASS.

The triples annotating the course materials are read next. We use the first part of a properties triples to locate a corresponding entry in the hash-table. The property and property value are stored in an association list and added to the `ArrayList` called `properties`. For example, the triple: “Red Black Tree, Definition, Red-Black-Tree-def.html” is stored in the entry for “Red Black Tree” as: [Definition, Red-Black-Tree-def.pdf] This enables us to add arbitrary properties to our instances knowledge base. It should be noted that the instances are added in the order in which they appear in the file. In particular, they are not sorted alphabetically. This gives the editor of the file housing the instances some control over the order of appearance in the GUI.

One kind of property is treated differently, namely the “Alternate Name” property. We maintain a dedicated `ArrayList` for them, called `alternateNames`. Whenever this kind of triple appears, we will extend this particular `ArrayList`.

GUI

When visiting the server, the user is presented with three ways of locating information: (1) a search box, (2) a directory listing of concepts and (3) a graph representing an ontology over those concepts. This initial page is shown in figure 5.

CSSE 230 Knowledge Server

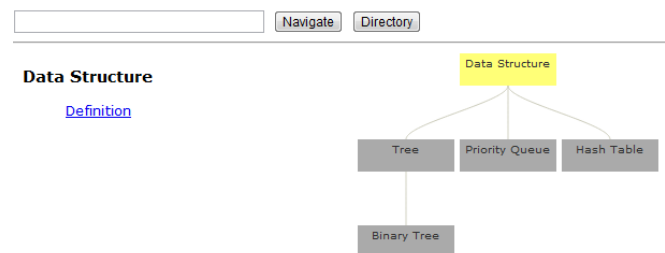


FIGURE 5
INITIAL SEARCH PAGE.

The search box contains an auto complete feature which is pre-loaded with the names of the concepts in our system. Figure 6 shows the result of typing the letter ‘b’. This makes typing and selecting concepts easier and compensates for having to search names which include capitalization and spaces. Upon selection of a keyword, we search the data stored in our data structure and display the instance data in an area below the search box.

If a user is unsure about the name of a concept or simply wants to explore the contents of the server, they may request a directory listing of concepts. Figure 7 shows the results of such a request.

In the directory listing, concepts are listed in alphabetic order. Links on this page simply invoke the search functionality of the server, initialized with the keyword as it appears in the listing.

CSSE 230 Knowledge Server

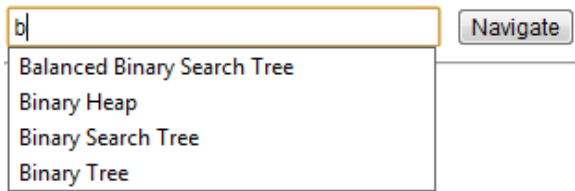


FIGURE 6
AUTOCOMPLETE FEATURE.

On start-up, we display the upper portion of the ontology graph. The user may click on any of the concepts resulting in several events: we search our internal data structure just like we do when the user employs the search box and we display the instance information just like we do when searching. As such, the search box, the directory listing and the graph are all equivalent ways of locating information in our system. The result of searching for Red-black trees is displayed in figure 8.

An additional feature enables users to reposition the graph at will in the right portion of the GUI.

CSSE 230 Knowledge Server

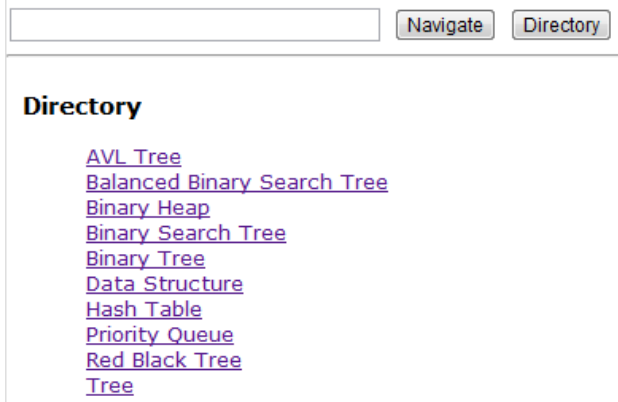


FIGURE 7
DIRECTORY OF CONCEPTS.

Figure 8 also displays another feature of our system. Certain file types are displayed in a frame of our GUI. They include .html and several image file types. The purpose is to afford a quick preview of key information such as definitions and examples. Users may click on the displayed information or the separate “Popout” button to display the information in a separate tab. Files which typically do not display well, such as .pdf and applets are displayed outright in a separate tab.

All in all, we have a variety of ways in which to navigate the information in our system.

CSSE 230 Knowledge Server

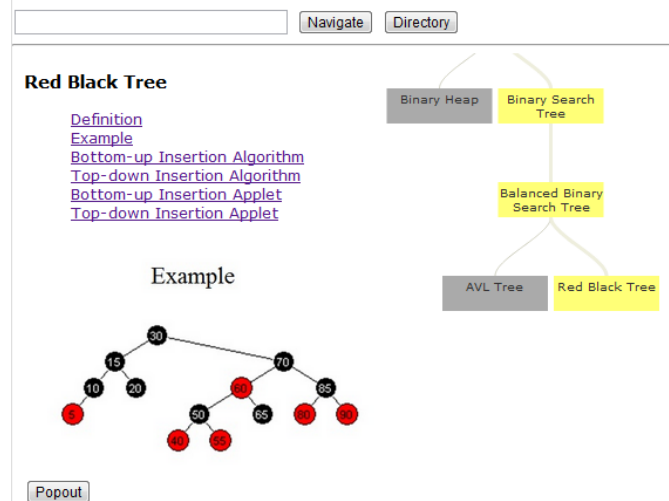


FIGURE 8
RESULT OF SEARCHING FOR RED-BLACK TREES

EVALUATION

We deployed the knowledge server about half way through the term. We would have liked to deploy it at the beginning of the term, but decided to improve the GUI to its current form and we were still editing the ontology and knowledge base.

To obtain feedback about the usefulness of the knowledge server, we assigned a homework in which students are asked to perform a comparative evaluation of several data structures. In particular, they were asked to develop a “Field Guide” for a wide set of data structures that were covered in class. As part of this guide, students had to provide an example application that serves to highlight the strength of each data structure. Additionally, they had to justify their choice by appealing to the complexity of major operations, i.e., inserting, removing and lookup.

They were asked to use the knowledge server first, before venturing on to other sources of their choice. We then conducted an anonymous survey of our students to assess the usefulness of the contents and the interface of the system. Table 1 provides the mean responses for questions 1 and 2 using a scale of 1-5; strongly agree to strongly disagree. As we can see, students liked the user interface a lot and were happy with the contents.

TABLE 1
RESPONSES TO QUESTIONS 1 AND 2

Survey Item	Mean
To develop the Data Structures Field Guide, the contents of the Knowledge Server was useful.	2.5
The user interface of the Knowledge server is useful.	1.9

We followed up these questions with six open ended questions in which students were invited to provide specific feedback. We will summarize key feedback and respond to it.

- *What are some of the contents items of the Knowledge Server that you found useful?* Students liked the definitions which are part of each data structure. Since they have only seen some of the data structures in prior courses, it helped them to familiarize them with those. We suspect that the definitions were also helpful in distinguishing among the many data structures that we did study in our course.
- *What are some of the user interface items of the Knowledge Server that you found useful?* Students very much liked the diagrammatic representation of the ontology. Students commented that it help them understand the relationship between the data structures.
- *What are some items that we should consider adding to the contents of the knowledge server?* Several students would have liked to see information on sorting and searching. Some students would have liked to see direct information on the runtimes of operations. Several students would have liked to see less external information such as provided by the links to Wikipedia pages and more information produced by us, among others providing sample code. We are in the process of adding knowledge about sorting and searching and plan to continuously add useful data.
- *What are some improvements we should consider making to the user interface?* Students made several suggestions. A key suggestion was to have the pop-out button display information not in a separate tab but in a separate pane. We since implemented this and most of the other suggestion made by our students.
- *Do you have any additional feedback about the knowledge server?* Students would have liked to see more of our own material.
- *To develop the Data Structures Field guide, which resource(s) did you find most useful?* Students found Wikipedia most useful.

The response to the last question suggests that our knowledge server is only useful in the sense that it contains links to appropriate Wikipedia pages. We would like to argue that the assignment based on which our students were asked to evaluate the server does not do justice to the power of our server. The majority of the materials that we developed are designed to learn the data structures covered in the course. As such, an additional survey needs to be conducted that assesses the usefulness of the knowledge server when it comes to learning the data structures. We suppose that it is some consolation that students did not cite the textbook even though it is well written.

CONCLUSIONS

We plan to add the following extensions to our system.

- 1) The most important extension would see this system integrated with other courses at our own institution and beyond. Our Data structures course is the third course in our introductory course sequence. We are interested in extending the server to cover materials from our CS-1 and CS-2 courses. We use a very popular textbook for the Data structures course and as such would be interested in sharing our server with colleagues who use the same book and possibly a similar approach to teaching the materials. It would be exciting to collaborate with colleagues from other institutions to further develop the course materials so as to customize them to different kinds of learning and teaching styles. The Sakai Project [8] seems to be a good vehicle for facilitating such efforts. Perhaps contents producers with mass media experience choose to join the effort.
- 2) We intend to add a social networking component that enables students to submit additional learning materials and have them reviewed by fellow students. We believe that such a component has several benefits. It would result in a better system and it would draw in students by giving them ownership in the learning materials.
- 3) We intend to annotate the properties file with a classification. While we are unsure about the syntax, we plan to associate chapter and section numbers with the items appearing in the ontology and instance files. This way, we can provide a third way of navigating the materials, facilitating self-study.
- 4) We are currently experimenting with a schedule maker, based on the classifications listed in (3).

I. Future of Textbooks

What does this mean for the future of textbooks? It is doubtful that they will be printed. Certainly, distribution in electronic format saves money and enables at the very least free form search. Publishers could use the table of contents to build an ontology and use it to annotate at least major sections of a book. While not ideal, a good step in the right direction. We believe that the future of textbooks lies in an approach as used by Wikipedia®, i.e. a collaborative approach where authors provide different kinds of learning materials appealing to a variety of learning styles and backgrounds. While it is harder to get credit for work done, it is not hard to imagine alternatives. If push comes to shove, contents modules could be made available through iTunes®.

REFERENCES

- [1] Bierman, P., "Reconsidering the Textbook", <http://serc.carleton.net/files/textbook/summary.pdf>, 2006
- [2] Butler, D., "The Textbook of the Future", *Nature*, Vol. 458, 2009, pp 568-570.
- [3] Anderson, T., Whitlock, D., M., "Special Issue: The Educational Semantic Web", *Journal of Interactive Media in Education*, Vol.1, 2004.
- [4] Lewin, T., "In a Digital Future, Textbooks are History", *The New York Times*. August 9, 2009.
- [5] National Research Council. "Preparing for the Revolution", *National Academies Press*, 2002.

- [6] Ohler, J., "The Semantic Web in Education", *EDUCAUSE Quarterly*, Vol 31, No 4. October-December 2008.
- [7] Protégé. <http://protege.stanford.edu/>
- [8] Sakai. <http://www.sakaiproject.org/>
- [9] Warlick, D., "Textbooks of the Future", In *Tech & Learning*. <http://www.techlearning.com>, 2004