CANOE: A Course Assembly and Normalization Tool for E-learning

Han Xu Honghan Wu Yuzhong Qu
Dept. Computer Science & Engineering, Southeast University，Nanjing，China 210096
{collin, xobject, yzqu}@seu.edu.cn

Abstract: Metadata plays a very important role in searching and discovering learning resources across the Web. To help users convert legacy Web-based courses into the equivalent ones that are conformant to popular metadata and content packaging standards, we designed and implemented a course normalization and assembly tool called CANOE. By offering the ability to perform course normalization and providing a high-level user view, CANOE enables the user to build standardized courses easily and quickly.

Keywords: course; metadata annotation; SCORM.

1 Introduction

Although the content format is based on common standards that are widely supported by Web clients, there remain difficulties in searching and reusing learning resources as their number is continually increasing. The use of metadata can efficaciously solve the problem. With metadata attached, learning resources can be easily managed and accurately searched across the Web. In addition, content packaging standards are necessary for exchanging learning resources across different tools and systems. Therefore, many organizations in this area have proposed their metadata and/or content packaging standards for e-learning like IEEE LOM [1], IMS [2] and SCORM [3], etc. With these standards approaching maturity, there is a growing demand for building courses that are conformant to them. However, there are some practical problems that may exist on the way to the widespread use of standardized courses.

Firstly, before these standards become widely popular, large volumes of Web-based courses have been constructed without conforming to any of them. As much effort has been made, it is neither economical nor reasonable to waste these *legacy courses* and build new ones from scratch. Instead, if we could take full advantage of these legacy courses, it will save much time and money in the construction of standardized courses, and the existing investment can be effectively protected. Secondly, though there have been many course tools (Microsoft LRN [4] for example) that can be utilized to create standardized courses, they tend to be kind of visualized editors for metadata and manifest files, which creates difficulty for authors who are not familiar with e-learning standards to build courses. Therefore, it will be of great significance to the wider deployment of standardized courses, if the above problem can be solved. To meet these challenges, we have designed and implemented a course normalization and assembly tool called CANOE (Course Assembly and NOrmalization tool for E-learning).

In this paper, the standard that CANOE supports is introduced firstly, and then the analysis and design of the tool is presented, and then a related algorithm is given, finally some conclusion and future work conclude this paper.

2 An Overview of SCORM

SCORM (Sharable Content Object Reference Model) is a set of Web-based e-learning standards developed by ADL (Advanced Distributed Learning). A requirement of the SCORM is that the learning content be interoperable across multiple LMSs regardless of the tools used to create the content [3]. SCORM does not prescribe the specific content and delivery format of a learning resource, which is supposed to be based on common standards that are supported by the Web client. What is actually defined by SCORM (refers to SCORM 1.2 if not

https://doi.org/10.23106/dcmi.952107912

otherwise stated) includes two parts: Content Aggregation Model (CAM) and Run-Time Environment (RTE).

In the Content Aggregation Model part, SCORM defines three kinds of components for the content model: Asset, Sharable Content Object (SCO for short) and Content Aggregation. For being able to be searched and discovered, a learning resource needs to be annotated with appropriate metadata that describes its characteristics. The SCORM CAM distinguishes three kinds of metadata according to what it describes: Asset Meta-data, SCO Meta-data and Content Aggregation Meta-data. The rest part of SCORM CAM is Content Packaging, which describes the content structure and sequence of navigation for a particular learning experience. The purpose of Content Packaging is to provide a standard means to exchange learning resources across different tools and systems.

3 Analysis and Design of CANOE

3.1 Requirement Analysis

As aforementioned, there are mainly two issues need to be addressed in creating standardized courses: how to deal with so many existing legacy courses in a reasonable manner and to make it as easy as possible for non-expert users to create standardized courses.

For the first issue, it can be solved by offering the ability to normalize legacy courses. We use the term *normalization* to denote the operation of exporting a standard (for example, IMS or SCORM, etc.) conformant course from an existing legacy course. To solve the second issue, a course tool is supposed to provide its users with a high-level user view i.e. to make the user view be independent of specific e-learning standards so as to eliminate the need for the author to understand the standards to which the course being created is conformant.

According to these ideas, we have designed and implemented CANOE (Course Assembly and NOrmalization tool for E-learning). The objective of the tool is to assist users, especially those who are not familiar with e-learning standards, to build standardized courses quickly and easily. The requirements of CANOE include:

- It should support SCORM 1.2.
- It should offer the ability to normalize an existing legacy course into its SCORM 1.2 conformant equivalent.
- It should be able to assemble existing learning resources (conformant or not) into a SCORM 1.2 conformant course.
- It should provide a friendly user view such that authors who are unfamiliar with SCORM can also complete the task of building standardized courses with ease.
- It should be intelligent to some extent to maximally reduce the workload of the user.

3.2 Design Issues

To meet the above requirements, CANOE is designed with the following characteristics.

3.2.1 A High-Level User View

As one of its features, CANOE provides its users with a high-level user view. With standard-specific details being abstracted away, this high-level user view frees the author from knowing the underlying document formats prescribed by SCORM. The high-level user view has been designed according to the following rationales:

- It should make the tool easy-to-use and -understand for users, especially those who have very little knowledge of the SCORM specification.
- It should be designed for easy and efficient mapping between the user view and physical storage.

In a word, the abstract granularity of the user view should be controlled at a level that both the first and second points above should be satisfied.

3.2.1.1 Simplified SCORM Content Model

To make it easier to understand, the SCORM content model is cautiously tailored in CANOE by omitting the component Asset. Though this approach may somewhat go against with the best practice [5], it is the tradeoff for simplicity and rapid development. In the simplified content model, there are only two components: SCO and Content Aggregation. To avoid using jargon directly from SCORM, we use the terms *Page* and *Unit* in the user interface to refer to SCO and Content Aggregation in the content model respectively. In this way, from the perspective of users, the content structure of a course is simply a tree made up of *page nodes*

https://doi.org/10.23106/dcmi.952107912

and *unit nodes*. Thus the process a user builds a course can be identified as a series of actions that begins with an empty tree, and proceeds to insert *unit* and *page nodes* consecutively, during which corresponding metadata is associated with each node.

3.2.1.2 User-friendly Metadata View

Metadata annotation is an important but arduous task in creating a course. In an attempt to make this work easier for the user, the following techniques are used:

– Have the metadata entries well organized on the user view so that data *is structured in a meaningful way for the user* [6].

– Allow the user to predefine default values for some of the metadata entries, especially those that have the same value on almost every new node.

– Extract useful information, if available, from the learning content to fill in metadata entries. For example, we can extract information from HTML elements like TITLE and META, etc.

– Check the validity of inputs automatically and give hints to the user when an invalid value is encountered in a metadata entry.

– Automatically maintain the consistency of metadata entries that exist in more than one place.

3.2.2 Normalization for Learning Resources

In CANOE, normalization is classified into two categories according to the level of granularity of the object being processed: SCO-level normalization and course normalization.

3.2.2.1 SCO-Level Normalization

SCO-level normalization is transparent to end users, since it is automatically invoked once a learning resource is selected to be added into a course. The purpose of SCO-level normalization is to embed in a SCO, when necessary, the script required for interacting with an LMS and to sanitize the HTML page for being a SCORM conformant SCO.

3.2.2.2 Course Normalization

Course normalization is used to export a SCORM conformant course from an existing legacy course. Compared to SCO-level normalization, this process is more complex and difficult. The primary difficulty is how to generate course content structure automatically from an existing course. After investigating many existing courses, we found out that in spite of having been created without conforming to certain well-known standard, they are not absolutely irregular. Instead they usually have their own rules. Among these rules, two kinds of information may be useful in generating the content structure: file naming rules and inter-page links. CANOE allows the user to select from the following two modes when performing course normalization.

1) Generating Content Structure According to File Naming Rules

If the file naming rules of an existing course reflect the content structure of the course, the user may perform course normalization with this option. In this mode, the tool attempts to find a proper position for each node to be inserted by contrasting the name of the associated file against the file naming rules. Take it as an example, let us suppose a course has the file naming rules that the first two digits in the filename denote which chapter it is in and the following two digits denote the section number. For units, if exist, the unit number is followed with a leading underline ('_'). In this way, an HTML page with the filename *0201_1.htm* will be recognized as the first unit in section one of chapter two. If this mode is selected to perform course normalization, the user needs to submit the file naming rules to the tool in the form of a regular expression. For the example above, it should be *(\d{2})(\d{2})[_\d]{0,}.htm*.

2) Generating Content Structure According to the Inter-page Links

If the inter-page links of an existing course are hierarchical (for example, the main page of the course contains all links to the entry pages of every chapter which again contains all links to entry pages of every section in it, and so on.), then this mode can be effective in generating content structure. In this mode, the user needs to specify the main page of the whole course when performing normalization.

After the mode being selected and necessary parameters being submitted, the tool begins the actual normalization process. The main steps of this process include:

(1) Explore and analyze the course to be normalized and generate the content structure using an appropriate algorithm according to the mode being selected.

(2) Create a new empty course that is conformant to SCORM and proceed to add unit and page nodes consecutively based on the content structure that has been acquired.

(3) Perform SCO-level normalization for each page node automatically.

(4) Generate metadata and update manifest file for each node.

4 A Related Algorithm

Due to the limited space of this paper, we merely outline the algorithm to generate content structure according to the inter-page links.

```
private void generateTree_IPL(String szFileName,
      Node oParent, int nLevel) {
   Document oDoc = parseFile2DomDoc();
   NodeList oLinkList = oDoc.getHyperLinkNodes();
   for(int i=0;i<oLinkList.getLength();i++) {
      if(checkValidHref(oLinkList.item(i))){
          var sFileName = getRefFile(oLinkList.item(i));
          if (Exists(sFileName) && !Added(sFileName)){
             Element oTemp =
                m_treeDoc.createElement(“node”);
             oTemp.setAttribute(“FileName”, sFileName);
             oTemp.setAttribute(“level”, nLevel);
             oParent.appendChild(oTemp); }}}
   Node oSibling = oParent.getNextSibling();
   generateTree_IPL(getFileName(oSibling),
      oSibling, nLevel);
   Node oChild = oParent.getFirstChild();
   generateTree_IPL(getFileName(oChild),
      oChild, ++nLevel); }
```

5 Conclusion and Future Work

In this paper, we introduced the design and implementation of CANOE 1.0. Compared to other course tools [4, 7, 8], CANOE features the ability to perform legacy course normalization without presumption on the tool with which the course has been created. The tool has been put into use in Southeast University and has successfully normalized several legacy courses into SCORM 1.2 conformant equivalents. It is proved to be cost effective to build SCORM conformant courses with the help of normalization that CANOE offers. We believe that course normalization is one of the important alternatives to acquire standardized courses and we hope our work can contribute in this area.

In future work, we will extend CANOE to support IMS, SCORM 2004 [9] and other promising e-learning standards. Besides we will consider the support of normalizing courses in a collaborative manner. This will be very useful in situations where a legacy course was constructed by multiple authors.

Acknowledgement

This paper is jointly supported by Key Technologies R&D Programme with project no. 2001BA101A08-01 and JSNSF with project no. BK2003001. The authors are also very grateful to Ningsheng Jian, Wennan Shen, Dongdong Zheng for their efforts taken in the implementation of CANOE, and to many others who make the tool become reality.

References

1. IEEE LTSC. IEEE Standard for Learning Object Metadata (LOM). http://grouper.ieee.org/LTSC/wg12/.
2. IMS Global Learning Consortium, Inc. http://www.imsglobal.org/.
3. ADL Technical Team. SCORM 1.2. http://www.adlnet.org/
4. Microsoft Learning Resource iNterchange 3.0. http://www.microsoft.com/elearn.
5. Learning Systems Architecture Lab, Carnegie Mellon University. SCORM Best Practices Guide for Content Developers 1st Edition. http://www.lsal.cmu.edu/lsal/expertise/papers/presentations/pfiisd20040216/pfiisd20040216.pdf
6. D.R. Rehak, N. Pasini and W.H. Blackmon. Scissors, Scotch® Tape, Post-its®, Magic Markers® and Colorforms®: "LO-Tec" Tools (and Toys) for Creating Learning Objects. http://www.lsal.cmu.edu/lsal/expertise/papers/notes/lotec05052003/lotec05052003.html.
7. Reload Editor. http://www.reload.ac.uk/.
8. THESIS 2.0. http://www.thesispro.com/
9. ADL Technical Team. SCORM 2004 Overview. http://www.adlnet.org/.

https://doi.org/10.23106/dcmi.952107912