Adaptive Hypermedia Content Authoring using MOT3.0

Jonathan G. K. Foss and Alexandra I. Cristea

Department of Computer Science, University of Warwick,
Coventry, CV4 7AL, United Kingdom
{J.G.K.Foss, A.I.Cristea}@warwick.ac.uk

Abstract. Ideally, in a web environment, each person should receive the exact information that he/she needs. If the needs of the users vary across a large number of people, then the task of creating sections of information, and putting those together in the right order can be daunting. This tasks falls to the author of adaptive hypermedia and this is the reason why research into better tools, that automate part of the authoring tasks, or make individual authoring tasks easier, is continuing. In this paper we present MOT3.0, which is an improved version of our main authoring tool, MOT.

Keywords: authoring of adaptive hypermedia, MOT3.0, MOT, AHA!

1 Introduction

Adaptive hypermedia allows content to be displayed differently to diverse users depending on the needs of that particular user. This content needs to be authored in such a way that sections of it can be shown or hidden according to various conditions.

Creating the perfect way of authoring for adaptive content is still an unsolved problem. This paper tackles this issue, by presenting an improved version of an existing authoring tool.

MOT (My Online Teacher) is a collection of authoring tools for adaptive hypermedia, which has been gradually developed since 2000 [8]. MOT is currently, to the best of our knowledge, one of the only authoring tools for adaptive hypermedia that is *general purpose*, in the sense that it can author for various adaptation engines, as well as for various topics (educational or otherwise). The way it does this is via an export language for the static content specifications, called CAF [9], and for the adaptation specifications, called LAG [9]. Any delivery system capable of reading these languages can play adaptive content created by MOT. Previous research has created software to export content from MOT to AHA! [11], WHURLE [2] and the commercial system learning management system (LMS) Blackboard [1].

The MOT tool adheres to the separation of concerns principle, thus separating reusable types of components, as advocated by many adaptation frameworks (e.g., AHAM [10], WebML [5], GAHM [18], XAHM [4], LAOS [7], GAM [12]).

One of the main features of MOT [20] is that it provides a way of separating the content, which is stored in domain maps, from the actual lesson goals, which are stored in goal maps. This technique is implemented according to the LAOS [7]

adaption framework. MOT facilitates the authoring of both domain maps and goal maps, whilst the other layers of the LAOS framework are edited by other tools.

MOT is aimed at being easy to use and expressive. An evaluation of the current implementation of MOT (MOT1.0) [20] confirmed this to some extent, by showing that users understood the theory behind domain maps and goal maps. However, the evaluation also showed that the user interface of MOT1.0 is insufficient. Additionally, the functionality of creating adaptive content is appropriate, but there were important static, non-adaptive content creation standards that were not imported to MOT.

Thus, the ***main goals*** of the research presented in this paper are to:

- Revisit MOT from the point of view of *authoring usability*
- Revisit MOT from the point of view of *reuse of existing content*, be it adaptive or, as in most cases, non-adaptive
- Create, based on these revisits, a new iteration of the MOT system, MOT3.0

The remainder of the document is organized as follows. In section 2 we discuss the improvements we have made to the functionality of MOT. Section 3 discusses follow-up research, and we conclude in section 4.

2 Improving My Online Teacher

In the following, we outline each issue that has been revisited and improved, with respect to authoring usability and reuse of non-adaptive content. It is also important to mention that MOT3.0 is not a simple extension of MOT1.0, but is a complete rewrite of the previous system, keeping desired functionality and improving on any bugs, insufficient functionality, etc. found on the way. MOT1.0 was written in Perl[1], whilst MOT3.0 is written entirely in PHP[2].

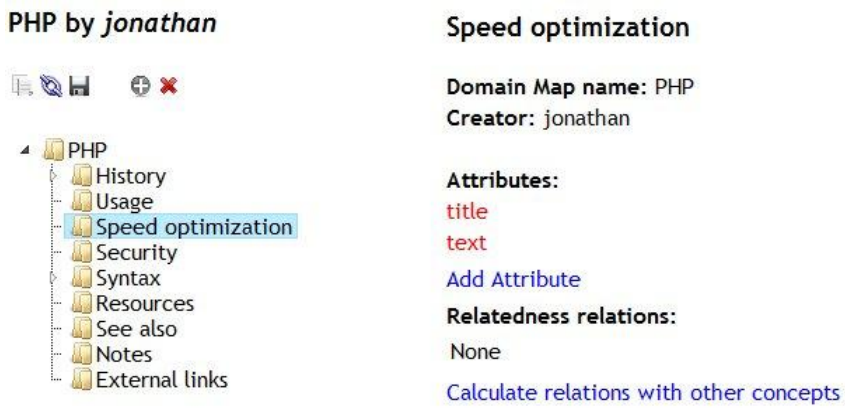

Fig. 1. The AJAX-based layout of MOT3.0

[1] http://www.perl.com/
[2] http://www.php.net/

Previous versions of MOT have used frames to divide the screen. For instance, when viewing a concept, the left-hand side of the screen shows the concept map, whilst the right-hand side shows the concept and its attributes. For MOT3.0 it was decided to use AJAX to synchronize data between the web server and the browser. This prevents the browser from needing to refresh the entire page every time the user manipulates a piece of content, and therefore provides a smoother user experience (see Figure 1).

In both versions of MOT, each concept contains a title attribute and optionally many other attributes (e.g. text, introduction, question, keywords). Content - or links to content - is stored within these attributes. Goal Maps arrange content into groups of sublessons. A sublesson is expressed as a link to a specific concept attribute. MOT3.0 adds a rich text editor from the Yahoo User Interface[3] library to provide a WYSIWYG (X)HTML editor to the user when editing concept attributes.

In MOT1.0, to reorder a map hierarchy involves clicking a cut link next to the source concept, then clicking on a paste link next to the target concept. This can be confusing for new users, since the target position of the node can be ambiguous. To solve this problem, MOT3.0 uses the JavaScript based component jsTree[4] for domain map and goal map hierarchies. This allows information to be updated using AJAX, and also enables the user to drag-and-drop concepts/sublessons to change the structure of the hierarchies. The drag-and-drop functionality is also used to allow users to copy/link concepts between domain maps.

Previous versions of MOT have provided limited search functionality. MOT1.0 provided a page that simply listed all the concepts in the system, together with their keywords. The user could then use their browser's 'Find' function (Ctrl+F) to search within the page. MOT3.0 adds a new search tool, with the results displayed to the user via AJAX. The search functions are used to help the user with copying and linking between domain/goal maps.

2.1 Compatibility with Other Systems and Standards

To maintain backwards compatibility with previous implementations of MOT, MOT3.0 is designed to use the same database schema as MOT1.0, for the moment. More importantly, for compatibility with AHA! [11], WHURLE [2] and the Blackboard LMS [1], the ability to transfer to and from CAF XML files [9] into the MOT database has also been retained in MOT3.0. This will allow easy integration to future developments of adaptive systems. Currently, one such target is GALE, the GRAPPLE FP7 STREP adaptation engine [14], which is still under construction, and is a follow-up of the AHA! [11] adaptation engine.

Moreover, other file formats that can now be used with MOT include standard formats for e-learning content, such as IMS-CP [15] and SCORM [21], and standard formats for testing and questionnaires, such as IMS-QTI [16]. These file formats are compatible with non-adaptive systems such as Sakai [19] or Moodle [17]. These import facilities are based on the import scripts created for MOT 2.0 [13].

[3] http://developer.yahoo.com/yui/
[4] http://www.jstree.com/

2.2 Importing from MediaWiki

MediaWiki[5] is a collaborative authoring tool, which provides a simple to use interface for authors. MediaWiki articles are written in WikiText, which provides basic formatting such as hyperlinks, references, tables and headings. Headings are used to denote different sections of an article. MOT3.0 is now able to download the contents of any named MediaWiki article, e.g., from Wikipedia[6], and split the article into different concepts, according to the structure of the article.

2.3 Importing from Microsoft PowerPoint

If adaptive courses are to be created, it is easier for educators to start with existing material, instead of creating content from scratch. Many existing educational resources are currently authored in Microsoft PowerPoint (PPT) slides[7]. MOT introduces a new method of importing content from presentations. PowerPoint files can be uploaded to the MOT3.0 server, where they are converted to HTML and JPEG files using OpenOffice.org[8]. From the HTML files, MOT can create a concept for each slide. Each slide contains an HTML version of the text on the slide (stored in a text attribute), an image attribute with a JPEG of the slide, and a notes attribute. A goal map is also generated. Currently, the hierarchies created by the PowerPoint importer simply contain a root node with a child node for each slide. Future research will investigate the use of the title of each slide to determine a concept hierarchy. For instance, if a presentation contains 5 consecutive slides with the same title, these 5 slides can be grouped together.

3 Follow-up and Future Research

Comments provided during an evaluation of the system have provided inspiration for the next iteration of MOT3.0. For instance, follow-up research is already investigating ways of ensuring users are able to quickly and intuitively create domain map hierarchies. One suggested method would be to add a function that allows the user to type concept names, separated by new line and tab characters, which will allow them to quickly describe an entire concept map based on subject headings. This would help the user to quickly create multiple concepts ‘at once’.

[5] http://www.mediawiki.org/
[6] http://www.wikipedia.org/
[7] http://office.microsoft.com/powerpoint
[8] http://www.openoffice.org/

We also propose to investigate crawling links in a Wikipedia article, and including any related articles in the resulting domain map. The question here is what the ideal depth of this import is (e.g., links in the article are of depth 1, links of links are of depth 2, and so on). It is clear that it should be smaller than six, because this may end up with a very large part of Wikipedia, as there are clearly parallels to the degrees of separation between people, which is six [6]. Links between articles on Wikipedia provide a way of inferring a relationship between these articles. It would also be useful to create a keywords attribute for each concept. These improvements could be implemented using similar techniques to that of DBPedia[12], which is a community effort to extract structured information from Wikipedia, and also VisWiki[13], which is using mind maps [3] to visualize Wikipedia topics.

Furthermore, in addition to the MediaWiki and PowerPoint import facilities, current research is investigating other document formats that can be separated into different concepts, and therefore used within adaptive hypermedia content. For example, HTML pages could be interpreted using a similar algorithm to the MediaWiki importer.

4 Conclusion

This paper has described the latest implementation of the My Online Teacher system, MOT3.0. Methods of extracting content from other non-adaptive sources have also been discussed.

There are however, some areas that need more research. This paper has suggested some areas that still need to be improved.

This paper has also introduced new tools for creating domain/goal maps for adaptive hypermedia from two non-adaptive sources – MediaWiki, and Microsoft PowerPoint. We have also proposed some ways of improving these importing functions, allowing more comprehensive domain maps and goal maps to be created.

For authoring tools in particular, due to the great effort the authoring process takes, it is vital to have greater acceptance of the tools that we propose to the users. Thus, research into methods of improving the functionality and usability of such tools is an important concern for the present. This paper both presents some answers, as well as raises new questions that, whilst originating in one system, are of greater importance to the community at large, as they reflect issues which any authoring system would need to take into account in order to be successful.

Acknowledgments. The work presented is funded by the GRAPPLE EU FP7 project.

[12] http://dbpedia.org/
[13] http://www.viswiki.com/

References

1. Blackboard Learning Management System, www.blackboard.com/
2. Brailsford, T.J.; Moore, A.; Stewart, C.D.; Zakaria,M.R.; Choo, B.S. & Davies, P.M.C. (2001) Towards a Framework for Effective Web-based Distributed Learning. Poster, 10th International World Wide Web Conference, Hong Kong, 1-5 May (2001)
3. Buzan, T. , Buzan, B.: The mind map book, BBC Active, ISBN: 1406612790 (2006)
4. Cannataro, M. Cuzzocrea, A. Mastroianni, C. Ortale, R., Pugliese,: A. Modeling Adaptive Hypermedia with an Object-Oriented Approach and XML. In Proc. of WebDyn'02, Honolulu, Hawaii (2002)
5. Ceri, S. Fraternali, P., Bongio, A.: Web Modeling Language (WebML): a modeling language for designing Web sites, Computer Networks, 33(1-6) 137-157 (2000)
6. Chuo, D.: E-mail Study Corroborates Six Degrees of Separation, Scientific American, August 8 (2003) http://www.scientificamerican.com/article.cfm?id=e-mail-study-corroborates (last accessed: 28/06/09)
7. Cristea, A. & De Mooij, A. LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators, WWW'03, Budapest, Hungary (2003)
8. Cristea, A. I., Okamoto, T., Cristea, P. D.: MyEnglishTeacher - An Evolutionary, Web-based, multi-Agent Environment for Academic English Teaching, Congress of Evolutionary Computation, CEC 2000, July 16-19, San Diego, USA, published on CD (2000)
9. Cristea, A.I., Smits, D., De Bra, P.: Towards a generic adaptive hypermedia platform: a conversion case study, Journal of Digital Information (JoDI), Special Issue on Personalisation of Computing & Services, 8(3) (2007)
10. De Bra P., Houben G.-J. & Wu H.: AHAM: A Dexter-based Reference Model for adaptive Hypermedia. Proceedings of the ACM Conference on Hypertext and Hypermedia (1999)
11. De Bra, P., Smits, D., Stash, N.: Creating and Delivering Adaptive Courses with AHA!, Proceedings of the first European Conference on Technology Enhanced Learning, EC-TEL 2006, Springer LNCS 4227, pp. 21-33, Crete, October 1-4 (2006)
12. De Vrieze P.T., Van Bommel P., Van der Weide T.: A Generic Adaptivity Model in Adaptive Hypermedia, AH 2004. Eindhoven, The Netherlands (2004)
13. Ghali, F., Cristea, A.I.: Evaluation of Interoperability between MOT and regular Learning Management Systems", 3rd European Conference on Technology-Enhanced Learning EC-TEL. 2008. Maastricht, The Netherlands, Springer, pp. 104 - 109, ISBN: 978-3-540-87604-5, DOI: 10.1007/978-3-540-87605-2_12 (2008)
14. GRAPPLE FP7 STREP project, http://www.grapple-project.org/ (last accessed: 28/06/09)
15. IMS Content Packaging (IMS-CP), http://www.imsglobal.org/content/packaging/ (last accessed: 28/06/09)
16. IMS Question and Test Interoperability Specification, http://www.imsglobal.org/question/ (last accessed: 28/06/09)
17. Moodle, http://moodle.org/ (last accessed: 28/06/09)
18. Ohene-Djan J. A: Formal Approach to Personalisable, Adaptive Hyperlink-Based Interaction. PhD thesis, Dept. of Computing, Goldsmiths College, Univ. of London (2000)
19. Sakai, http://sakaiproject.org/portal (last accessed: 28/06/09)
20. Saksena, M., Cristea, A.: Towards more efficient generic semantic authoring for adaptive hypermedia, Proceedings of the Adaptivity, Personalization and The Semantic Web Workshop at Hypertext 2006, 17th ACM Conference on Hypertext and Hypermedia, ACM Press, pp. 11-20, 23-25 August, Odense, Denmark, DOI: http://doi.acm.org/10.1145/1149933.1149936
21. Sharable Content Object Reference Model (SCORM), http://www.adlnet.org/Technologies/scorm/default.aspx (last accessed: 28/06/09)