

Exporting mathematical document collections to PDF

Zdravko Beykov

*Computer Science
Jacobs University Bremen
Campus Ring 1
28759 Bremen
Germany*

*Type: Guided Research Proposal
Date: March 9, 2008
Supervisor: Prof. Michael Kohlhase*

Executive Summary

Nowadays, we are living in a world where information is all around us. The internet is giving us the opportunity to explore a tremendous amount of data. Much of our difficulty lies in how to locate this data and with the advancement of technology this proves no easy task. Even when one finds the place where the information he searches for resides, he might not be able to narrow it down. Even if he succeeds in narrowing it down, the problem of understanding the idea closely linked to this information could arise. If some terms used in the explanation of the information are not present, then the user has to go through the process of locating each unknown node of this dependency separately and this would cost him a lot of time.

The idea of the project is to ease the quest of finding the relevant data one searches for and include all the dependencies of the document that the user is unfamiliar with before the main idea. This whole info would be exported to a PDF file that would be ready for printing. Thus, the end product would be nothing more and nothing less than what one needs for understanding a certain topic.

1 Introduction

The project would ease the extraction of printable data from a mathematical document collection in the semantic markup language OMDoc^[1]. It will be embedded into the SWiM wiki^[2] as a Flash Applet and would let the user choose the dependencies that would have to be included in the final PDF, which would result from an XSL transformation^[3]. This project could be helpful for students learning a certain topic by filling gaps of their knowledge of pre-requisite information.

2 Statement and Motivation of Research

This guided research would answer the question of how to utilize extracted mathematical information from the semantic wiki SWiM by including all the needed dependencies nodes into a single PDF file. Also, it will address the logical interconnection between semantic documents in order to get only relevant to the user data. This would be convenient, since in the end the user would have a complete PDF with only the needed info. It will be of great help for the learning of the material.

The project would rely on several outside resources. The first one is the OMDoc-HTML XSL stylesheet that would be used for displaying the selected nodes in HTML. Currently, this resource is available. The next one would be the OMDoc-TeX in the new XSL 2.0 format. It will be needed for converting all the dependencies into a simple TeX file that would later be transformed into PDF via LaTeX^[4]. The newest version of this file is not yet ready and therefore it will be worked on to the needed extent. The last needed resource would be the complete OMDoc version of the General Computer Science lecture notes used in Jacobs University for the course. Currently, they are available for the first part of the course, and soon the second part will be ready too.

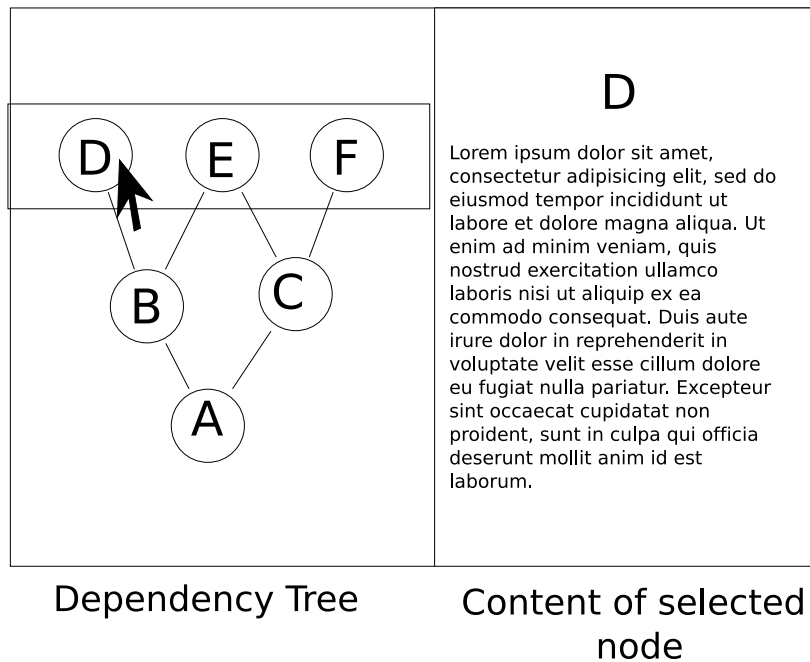
3 Planned Investigation

Dependency in an OMDoc document could mean several things like definition, proof, symbol, recommended-to-read, etc dependency. There could be numerous types, and, moreover, the SWiM wiki let's you even define custom ones. Each type of dependency would be shown, they would be represented in a different way but have the same functionality in the applet. Since all users are assumed to have different knowledge, no automated way of expanding dependencies would be used. Each user will decide for himself which dependencies to include and which not.

The idea of representing the dependencies as a graph, in which they will be nodes, would require a graphic environment that could be embedded in a website. Thus,

Adobe Flash would be suitable for the job, as it is very popular and very well supported on a number of different operating systems. Moreover, Flash has XML handling functions which would ease up dealing with OMDoc, which would be needed for finding the dependencies of the document.

The initial design would feature two frames. On the left side there would be a graphical representation of a tree. In it, the nodes would be the documents and the edges would be the dependencies.



The root of the tree (the initially selected document), would show its dependencies' nodes in a rectangle above itself. If the user decides to include one of them, then this node would lower itself graphically, get out of the rectangle, and its dependencies' nodes will appear in the rectangle above. Each time a node from the graphic is selected, its contents are displayed in the right frame which will be HTML. This can very easily be achieved using the browser's capability of XSL transformation. Both the XML(OMDoc) and its XSL files are available, so this task looks quite straightforward to implement. However, the communication between the Flash applet and the HTML frame that would contain this document would have to be investigated further since at first no simple solution can be seen.

The concatenation of OMDoc files into a single file would require some testing to come up with a visually pleasant solution. It should be logically structured, showing clearly the transition between each dependency node. Then, this whole document would have to be converted into a single PDF. LaTeX would be used as a version for transforming OMDoc to Tex files (the input files for LaTeX) will soon be available. At this stage, time will be spent to further develop the existing, but not

yet finished OMDoc-TeX XSL stylesheet. Also, there will be another difficulty as the transformation would have to be executed on a server, yet the Flash Applet runs in client mode. A client-server communication would have to be established for the transferring of the concatenated document to the server that would run LaTeX, and then offer the resulting PDF for download.

Lastly, as any software project, once it is released, it still has to be maintained. Moreover, this project is somewhat innovative, therefore it would most probably undergo many changes. The user's feedback would be collected in order to see if any features are clearly desired to be modified or added.

4 Evaluation Criteria

As we are dealing with a project that will be used by users of the semantic wiki SWiM, they can be the evaluators of the project. The people using the applet could leave comments on their impressions. The most important aspect would be the usefulness of the applet. The number of visits and/or created PDF's could also be tracked. The more people using the applet, the more significant its value would be. The opinions of the users would matter even more, as one could suggest improvements of current version of the project.

5 Timeline

March 12, 2008: Finish investigating SWiM dependencies graph flash applet.

March 18, 2008: Finish reading about Flash's basic graphic methods.

March 26, 2008: Finish implementing finding, extracting dependency nodes from an OMDoc in SWiM document.

April 6, 2008: Finish graphical dependencies implementation.

April 15, 2008: Finish implementing applet-html communication procedures.

April 18, 2008: Decide on the structure of the concatenated document.

April 25, 2008: Finish working on the OMDoc-TeX stylesheet.

May 5, 2008: Ready alpha version.

May 15, 2008: Finish testing the applet, release final version.

6 References

[1]. Michael Kohlhase. *The OMDoc document format*. OMDoc – An open markup format for mathematical documents. (83-217) [Version 1.2]. Number 4180 in LNAI. Springer Verlag, 2006.

- [2]. Christoph Lange. SWiM A Semantic Wiki for Mathematical Knowledge Management. Technical Report. Jacobs University. <http://kwarc.info/projects/swim/pubs/tr-swim.pdf> 2007
- [3]. Michael Kay. The Extensible Stylesheet Language Family (XSL). W3C Group <http://www.w3.org/TR/xslt20/> 2008
- [4]. Michael Kohlhase. *Transforming OMDoc by XSLT Stylesheets*. OMDoc – An open markup format for mathematical documents. (235-240) [Version 1.2]. Number 4180 in LNAI. Springer Verlag, 2006.