Large XML Document Manager and Visualizer

J. P. Merienne and C. Jacquemin

LIMSI-CNRS and University of Paris 11, BP 133, 91403 Orsay Cedex, France
{Jean-Philippe.Merienne,Christian.Jacquemin}@limsi.fr

Abstract

This work presents an interactive Graphical User Interface used to display and manipulate collection of large XML documents. It relies on a geometrical model that combines hierarchical and sequential representations. In this interface, XML structures are displayed on an inclined plane in a 3D environment. This interface provides users with a visualization that speeds up searches in collection of documents through simple interaction techniques. Document browsing is facilitated through display space optimization and direct manipulation of document structures. Besides, this approach tries to take advantage of 3D by engaging human 3D spatial cognitive capabilities while using 2D interaction to not disrupt user's habits. A user evaluation for document access is reported that shows that our interface tends to be faster than 2D interface in data searching while taking advantage of the user's natural attraction to 3D interface.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [User Interfaces]: Graphical user interfaces (GUI)

1. Introduction

The increase of easily accessible documents on the Web contributes to the expansion of markup languages for documents containing structured data. XML[?] is designed to deliver structured content over the Web. Users must be able to view XML documents as quickly and easily as HTML documents and take advantage of the structure and the semantics of XML markups. Managing and modifying XML documents, can then be an interesting user interface design problem. What modes of interaction can be offered for access and visualization of large structured documents such as corporate documentation or technical and scientific reports.

3D-XMI (3D XML Management Interface) is a new user interface used to display and process collection of XML documents. It is designed for rapid manual searches, easy management of the document's structure and takes advantage of human spatial memory (i.e. the ability to remember where we put something). In our current prototype we focus on the data model given by the Document Type Description (DTD) DocBook. It offers a wide range of structures for the description of books, manuals, theses, and corporate documents. Each XML document is converted into a geometrical model that combines hierarchical and sequential representations. The user can freely zoom and navigate in the geometrical model. Interactions, such as the merging of two XML documents, use simple 2D techniques and common pointing devices (like the mouse and the keyboard).

In this paper we first describe the conversion of XML documents into hierarchical tree structures through the DOM library and the generation of a geometrical model. We describe in detail the interactions and the operations in the visualization process. In order to test the efficiency of human spatial memory, we report on a user study that compares *3D-XMI* with two concurrent models: a 2D interface, *Cone-Tree* a full 3D interface.

2. XML Encoding

In order to facilitate the visualization and the processing of XML documents, they are first converted into tree structures through the Xerces library[?]. The Document Object Model (DOM) defines the logical structure of documents and the way in which they can be accessed and manipulated. The user, via the visualization interface, can browse and interactively modify document structures. When the user wants to save her changes, the interface calls the DOM parser in order to translate the tree structures into XML documents that conform to the DocBook DTD[?]. Such a generic framework could of course be extended to other XML document specifications (see Figure 1).

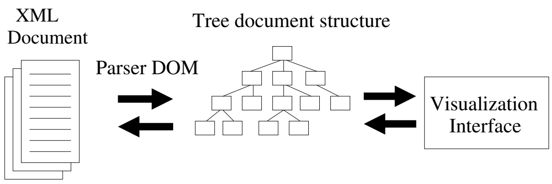

Figure 1: *The XML interaction.*

3. 3D XML Management Interface

In *3D-XMI*, each XML document has a geometrical model in a 3D environment. Like the *Data Mountain*[7] all models are placed on a inclined plane and are draggable to any place in this plane. A model is made of a set of labels that represent the parts of the document : paragraph, chapter, section, subsection... Each label is visualized at two levels of granularity. Color and title identify the label in the geometrical model. The full textual content of the label is displayed in the foreground when the item is selected. When a document is first loaded, all the labels are displayed sequentially.

The order of the labels in the sequence is the same as in the book: for example, the first title and paragraph in the book are the two-frontmost labels of the list. When a label is selected and opened via a mouse click, the path in the tree that leads from the root to the selected label is highlighted. Then the sequential form is soothly transformed into a tree through an animation, in order to facilitate the user's orientation and to keep a continuity between the two forms of the model. Figure 2 shows a combination of the tree and sequential forms.

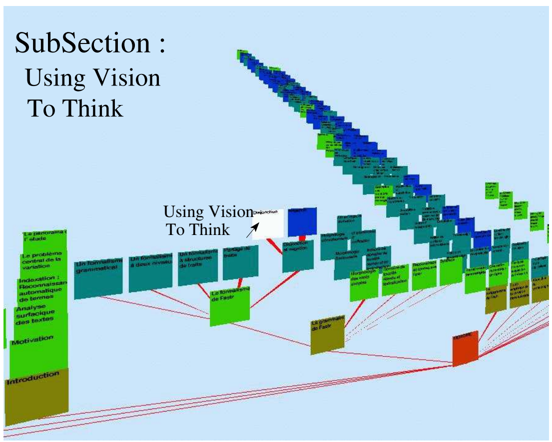

Figure 2: *Combination of sequential and hierarchical representations (Label contents manually added).*

The advantage of combining sequential and hierarchical visualization is to give a compact view of the document and to help users speeding up searches on the models by viewing all labels on a single display. The point of view is affected by a left rotation instead of being screen aligned. A study on 3D-text legibility[5] shows that with a sufficient rendering quality, small texts can be substantially rotated before reading performance suffers. The rotation of labels in our model allows us to increase the number of displayed labels without affecting readability.

3.1. Human Spatial Memory Capabilities

SpaceTree[6], *Cone-Tree*[8], or *Tree-map*[4] are examples of interfaces designed for 2D/3D-visualization and browsing. Since our interface is dedicated to the management of sets of XML documents, one of our primary motivations for the design of *3D-XMI* is to rely on natural human spatial memory and perceptual skills. Our pre-attentive ability to recognize spatial relationships based on 3D depth cues (like perspective views and occlusion) makes it possible to place labels at different distances and understand the spatial relationships of the models without thinking about it. We hope that the 3D depth of our interface and free users' interactions can take advantage of these capabilities.

3.2. Interaction

The most important feature of *3D-XMI* is to offer a rich environment for accessing and modifying XML document structures. First a selected substructure can be dragged with the mouse on a inclined plane. We believe that the user's act of directly placing structure chunks where she wants can support spatial memory. Moreover the user can freely navigate on the inclined plan with the arrow of the keyboard. She can then zoom in or out of a label in order to have a global view of the nodes. Navigating in all directions helps the user to understand better the 3D space by using parallax and 3D orientation[9]. In the sequential form, one of the problems is the occlusion of the labels. Selecting a label located at the farthest end of a list is very difficult. This difficulty can be avoided by crushing all the labels of this sequence that are in front of the selected labels (see Figure 3). This mode of interaction evokes a deck of cards. As described in Bernsen's taxonomy[1], we believe that this type of analogy can intuitively help users to manipulate our interface and avoid label occlusion.

Last, parts of a document can be copied and pasted into another structure. For this kind of action, we prefer to use a common 2D interaction model. For example to cut a part of a document, a label is selected and then the structure is split into two new structures. After such a manipulation, the XML marks are transformed in order to remain consistent with the DocBook DTD. For example, if the first node of a newly cut structure is a section and has four subsection children, the section is transformed into a book structure and the

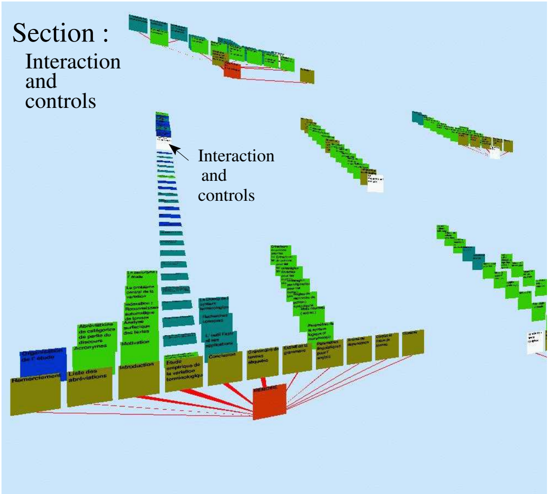

Figure 3: *Label occlusion avoided through label crushing.*

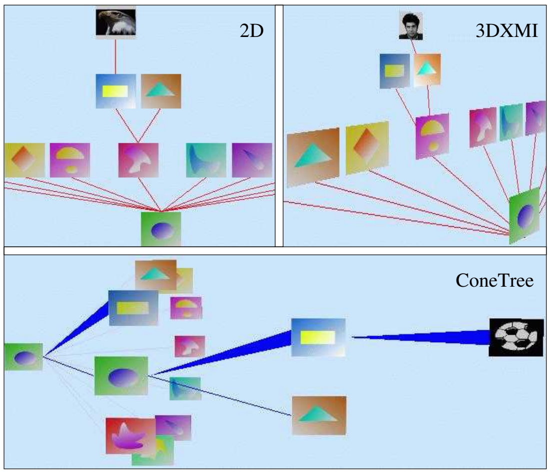

Figure 4: *Truncated views of 2D, 3D-XMI, Cone-Tree interfaces.*

subsections are transformed into chapters. Conversely, if a document is pasted as the child of a section of an other document, the book structure becomes a section of level 2 and the chapter a section of level 3. Figure 3 shows the display after several structural cut and paste actions on an XML document.

4. User Study

Since inclined plane perspectives such as *Data Mountain* stimulate spatial memory[7], users can take advantage of spatial cognitive abilities when they can organize freely their workspace. Similar capacities are used in our interface to manage XML documents. But is this spatial memory applicable when users search information in the tree structured model? In order to answer this question, we have conducted a user study that compares the three types of tree-browser interfaces shown in Figure 4. A simple 2D tree structure interface, a striped version of *3D-XMI* (without sequences to not compare too dissimilar features) and a striped version of the 3D interface *Cone-tree*[8]. Evaluations of *Cone-tree* already exist[2] and *Cone-tree* has already been used for searching in textual documents [3]. We however test the *Cone-tree* interface because we are not aware of any study focusing on the spatial memorization of such embedded conical layouts. The main purpose of our experimental evaluation is to show if the depth of the two 3D interfaces help users to remember structures by stimulating their spatial memory. The analysis of variance is performed on subjective variables and time to retrieve information.

Ten users have taken part in this study. Five users are experts with 3D interface manipulation. The five other users are only introduced to 3D interface manipulation. Subject ages range from 12 to 32 years, and all have normal or corrected-to-normal vision. They are 2 females and 3 males in expert group and 3 females and 2 males in initiated group. The study runs on a 2.4 GigaHz Pentium 4, with a 512 MB memory, a 17” display and an ATI radeom 9700 Pro GPU.

4.1. Experiment

The purpose of the experiment is to ask users to retrieve *information labels* (leaf nodes) by browsing the tree structure through the three interfaces. Clicking on a a *structure label* (an intermediary node) opens the sub-nodes and reclicking on it closes its subtree. In the three interfaces, the underlying tree structure is the same. Structure labels have different colors and textures with different shapes. These textures have no meaning by themselves; they are used to help the user by providing her with various visual clues. The information labels that must be retrieved by the users are pictures of animals, celebrities, or objects. In order to test the memory capacity of the subject, we arrange the interface so that only one of the paths that lead from the root to the selected label is shown at a time. The interactions with the models are the same in the three interfaces, except the fact that the two 3D interfaces have commands to zoom in and out the model.

Each subject has to test the three interfaces. The order in which the subjects test the interface are balanced. The test protocols are the same in the three tests. First all the labels in the tree structure are shown to the user and we inform her that she has to remind information labels. Once the labels of the model have been shown, the user is required to find ten information labels. For the balance of the experiment, the order of texture labels (animals, celebrity, object) is changed after each user test. To conclude the tests the user answers a questionnaire that focuses on three topics: perceived easiness, effectiveness, and pleasantness. Each answer is scored on a 1 to 3 scale.

4.2. Retrieval Time

Result analysis reveals a statistically reliable effect of interface on retrieval times ($F(2/12)=31.182$, $p<0.001$). The main finding is that the simple *Cone-tree* slows down information access when compared with the 2D interface ($F(1/6)=27.399$, $p=0.002$) and the simple version of *3D-XMI* ($F(1/6)=44.548$, $p=0.001$). *3D-XMI* is only slightly faster than the 2D interface. The statistical results tend to be significant ($F(1/6)=5.344$, $p=0.06$).

4.3. Subjective Ratings

The answers to the questionnaire show that the 2D interface is the easiest interface followed by the simple version of *3D-XMI* ($F(2/12)=109.68$, $p<0.001$). The simple *Cone-tree* seems to be the most difficult interface to use. For effectiveness, the results show that there are no significant differences between 2D interface and our simple version of *3D-XMI*. The results also show that in term of effectiveness, the simple *Cone-tree* is worse than the other two interfaces ($F(2/12)=31,92$, $p<0.001$). Finally the results on pleasantness ($F(2/12)=59.55$, $p<0.001$) indicate that the simple *Cone-tree* is the most pleasant interface, followed by our striped interface and last, the 2D interface (see Figure5).

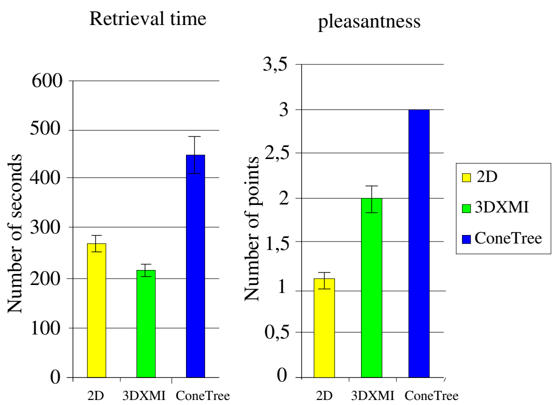

Figure 5: *Average retrieval times and Average pleasantness.*

5. Discussion and Future Work

The user study reported here does not demonstrate that spatial memory is useful for accessing information through tree structures. It however demonstrates that *Cone-tree* is not efficient even though users find it very pleasant. The lack of spatial references and the rotation of the cones disorientate users. User satisfaction is certainly due to the rotation animation and to the original 3D design of this interface.

On the contrary the 2D interface seems to be efficient for manual searching tasks but does not have the favor of users. The effectiveness of the 2D interface certainly comes from the user practice of this type of interface. The user study also suggests that *3D-XMI* seems to be a good alternative because it combines effectiveness and pleasantness. The lack of user practice for this type of interface probably minimizes the effectiveness of spatial memory.

Further work includes the addition of graphical device like label shadows on floor and walls, and audio information to help users understand better the spatial configuration.

6. Acknowledgements

We would like to acknowledge Stéphanie Buisine and Frédéric Vernier for their helpful advice.

References

1. N. Bernsen. A revised generation of the taxonomy of output modalities. In *Proceedings of AMODEUS number RP5-TM-WP11, 1994*, page 13, 1993. 2
2. A. Cockburn and B. McKenzie. An evaluation of cone trees. In *People and Computers XV (Proceedings of the 2000 British Computer Society Conference on HumanComputer Interaction)*, University of Sunderland, 2000. Springer-Verlag. 3
3. M. A. Hearst and C. Karadi. Cat-a-Cone: an interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. In *Proceedings of SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval*, pages 246–255, Philadelphia, US, 1997. 3
4. B. Johnson and B. Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings IEEE Visualization'91*, pages 284–291, IEEE, Piscataway, NJ, 1991. 2
5. K. Larson, M. van Dantzich, M. Czerwinski, and G. G. Robertson. Text in 3D: Some legibility results. *ACM Press New York, NY, USA*, pages 145–146, 2000. 2
6. C. Plaisant, J. Grosjean, and B. Bederson. Spacetree: Supporting exploration in large node link tree, design evolution and empirical evaluation. In *INFOVIS 2002*, pages 57–64, IEEE Symposium on Information Visualization, 2002. 2
7. G. Robertson. Data mountain: Using spatial memory for document management. In *Proceedings ACM Symposium on User Interface Software and Technology (UIST'98)*, pages 284–291, IEEE, Piscataway, NJ, 1991. 2, 3

8. G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone trees: animated 3D visualizations of hierarchical information. In *Proc. Human factors in computing systems conference*, pages 189–194, 1991. 2, 3

9. U. Wiss and D. Carr. A cognitive classification framework for 3-dimensional information visualization. 2