Collapsible Cylindrical Trees: A Fast Hierarchical Navigation Technique

Raimund Dachselt, Jürgen Ebert
Dresden University of Technology, Department of Computer Science
{raimund.dachselt,juergen.ebert}@inf.tu-dresden.de

Abstract

This paper proposes a new visualization and interaction technique for medium-sized trees, called Collapsible Cylindrical Trees (CCT). Child nodes are mapped on rotating cylinders, which will be dynamically displayed or hidden to achieve a useful balance of detail and context. Besides a comprehensible three-dimensional visualization of trees, the main feature of CCT is a very fast and intuitive interaction with the displayed nodes. Just one click is needed to reach every node and perform an action on it such as displaying a web page. The CCT browsing technique was developed for interaction with web hierarchies but is not limited to this domain. We also present sample implementations of CCT using VRML, which show the usefulness of this intuitive tree navigation technique.

Keywords: *visualization, web navigation, hierarchy, interactive tree, sitemap, 3D graphics, VRML, XML*

1. Introduction

With the widespread use of internet technologies and the availability of huge information spaces on the world wide web, fast navigation becomes one of the crucial tasks. Since hierarchical structures like organization or web structures, product catalogues or book chapters are omnipresent, they need to be visualized on the web, too. Moreover, a fast access of web pages, products, or content is of high importance besides the presentation. It can be noticed, that structured information of this type is often organized in a strictly hierarchical format or can be easily converted to trees. That is why visualization and navigation of trees is the focus of this work. Web page hierarchies shall serve as a prominent example of a structured representation of hierarchical data.

A number of web navigation techniques were developed, among them sitemaps in different formats to facilitate overview and quick navigation within web pages. The spectrum reaches from simple textual enumeration to traditional menus or even sophisticated hyperbolic visualizations. In the field of three-dimensional graph and tree visualization various interesting approaches were suggested, usually to display large hierarchies with the added benefits of the third dimension. However, these techniques are only sparingly used for real navigation tasks in the world wide web.

On one hand this was due to the lack of powerful 3D graphics capabilities of consumer platforms in the past. Meanwhile this has changed, though the problem of various proprietary 3D formats and plug-ins still prevents easy access to 3D internet applications. On the other hand the traditional 3D visualization techniques are often optimized for huge hierarchies and thus difficult to navigate for non-expert users.

To overcome the complexity of these solutions and still take advantage of the capabilities of three-dimensional visualization and interaction paradigms, we developed a comprehensible 3D navigation technique called Collapsible Cylindrical Trees (CCT). It allows a very fast navigation in medium-sized hierarchies. Though it was developed especially for web applications it is not limited to this application domain. As opposed to many other approaches our goal has not been to get insight into complex hierarchies but to quickly navigate through trees and perform an associated action on a node such as opening a web page.

This paper is organized as follows: The second section relates our work to previous approaches and motivates the development of our technique. Then, we introduce CCT in detail, followed by a discussion of typical application scenarios of this novel technique in section 4. Insight into implementation details is given in section 5, followed by a conclusion and outline of future work.

2. Related Work

Much research has been dedicated to the visualization of trees as a special type of graphs. Most of the work concentrated on the challenge to display large hierarchies in a comprehensible form. As Herman et al. describe it in [4], these approaches tried to overcome the disadvantages of visualizing large hierarchies, like reduced performance, viewability or usability. In our opinion especially the last disadvantage, lack of intuitive interaction, must be solved not only for huge but first of all for medium-sized hierarchies. Looking for example at common menu systems or program menus we notice, that they often provide very long lists and therefore need long vertical mouse movements to reach deeper nested entries. We tried to eliminate these deficiencies in the CCT approach with mapping different labels on a cylinder. This known idea is a useful amendment to two-dimensional menu systems, where vertical space is either limited or too many menu entries are presented at once. A recent example showing this technique are mobile phones with limited displays.

Classical tree layouts are top-down or left-to-right views, H-trees, radial views, balloon views, tree-maps, onion graphs, and various modifications based on these forms. The survey [4] lists detailed references to these approaches. A few of them do not use screen space efficiently [5]. Our CCT approach combines three traditional layouts to maximize use of available display size: a top-down view for the first tree level with a radial approach for siblings of a node and a left-to-right view for children of a branching node. Approaches like tree maps [5] focus on the content information of nodes and criticize the text only approach of other node and link diagrams [4]. We suggest that both aspects, clear textual description of nodes and additional graphical cues to content should be provided by a visualization. Moreover, with web page hierarchies the internal or branching nodes are equally important to the leaf nodes and must be displayed in the same meaningful way. Approaches like tree maps which basically display leaf nodes fall short in this respect.

Various 3D approaches were developed to display larger hierarchies or reduce visual clutter, such as cone trees [12] and augmented solutions. Some of them are based on 2D tree layouts, like the information cube [11]. Other problems occur with 3D visualizations, like object occlusion, difficult view adjustment [4], performance problems, or bad readability of 3D texts [8]. We took these problems into consideration while developing CCT and cautiously used 3D graphics. In addition to that many of the 3D approaches concentrate on conveying a sense of structural relations, whereas in our approach the interaction with nodes and high-quality text display play a more important role.

The horizontally oriented Navigation Cones in LyberWorld [3] improved on text readability and influenced our solution. Their successor, visual trees, is a generalization of the LyberWorld search trees. They solve the problem of too many sibling nodes with the help of a spiral display instead of evenly distributing them at the circular base of a cone. Although spiral trees are very elegant, occlusion still occurs and space remains limited. The CCT concept of *endless cylinders* tries to improve on that.

There also exists a variety of Focus+Context extensions to 2D and 3D visualizations like the hyperbolic browser [7] or hyperbolic cone trees for large directed graphs [10]. Fig. 1 shows a hyperbolic web sitemap of our research group, the Heinz-Nixdorf Endowed Chair for Multimedia Technology (MMT).

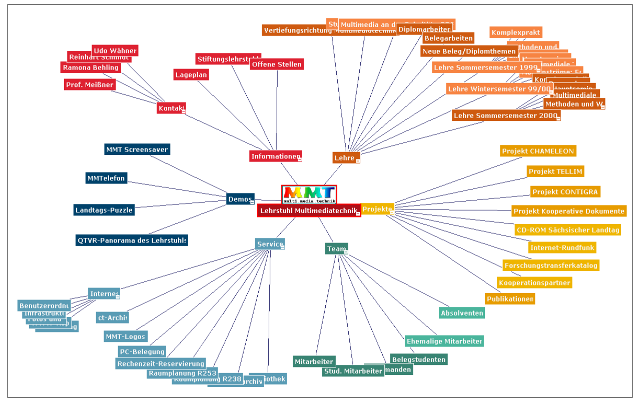

Figure 1. Hyperbolic sitemap of the MMT group

The problem with hyperbolic displays is again interaction. Due to the always changing position of the nodes it is very hard to build up muscle memory and avoid unnecessary mouse movements.

The review of Benford et al. [2] considers three-dimensional visualization techniques of the world wide web. Among them is a 3D hyperbolic visualization of sections of the World Wide Web [9] based on cone trees. The Harmony Internet Browser included a 3D visualization and navigation facility called information landscape, where document hierarchies were represented as blocks mapped onto a plane [1]. These and other approaches are again aimed at visualizing large hierarchies, are more difficult to comprehend in terms of 3D structure and clearly focus on structures and revealing relationships among web pages instead of fast navigation and interaction. We tried to solve the cognitive difficulties of 3D navigation in the CCT approach, where one of the goals was to use traditional 2D pointing devices.

3. Collapsible Cylindrical Trees

To overcome the problems mentioned above we developed a new visualization and interaction technique for medium-sized hierarchies. The idea of CCT is not to visualize an entire tree hierarchy at once but only to display the first two levels of a tree plus the chosen path down the hierarchy. Thus information overload is avoided and a quick overview guaranteed. Local detail and global structure are satisfactorily balanced. This section introduces the underlying model and the CCT features.

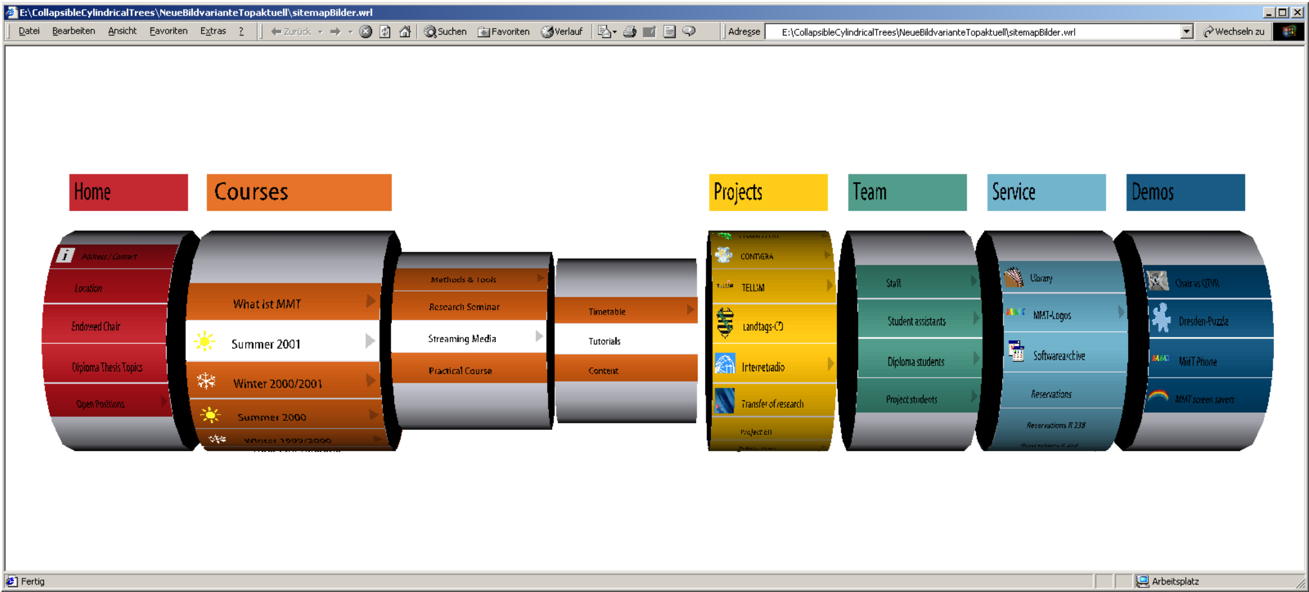

Figure 2. A web sitemap of the MMT group using the CCT visualization with current depth of 3.

3.1. The Model

Given a tree with a finite list of nodes, every node n has an associated set of attributes, which must contain a label and may contain an appropriate action, command, or URL. In addition to that, other attributes like color, size, or icon can be added depending on the semantics of the node. Every intermediate or parent node p has a set of children C. Let d be the depth of the tree and l the level within the hierarchy, where $l=0$ denotes the root node, and $l=1$ the children of the root node etc.

For every p (except the root node) a cylinder is constructed, on whose surface all nodes of C are displayed with their label on *facets*. The number of cylinder facets num_f is fixed to a certain value, e.g. $num_f = 20$. Facets are evenly spaced covering the whole cylinder, so with $num_f = 20$ every facet covers 18° of the cylinder's surface. Facets are independent 3D objects, providing a basic color, text label, and icons for other node attributes. Each node of C is associated with a facet. If the number of children is smaller than num_f, unnecessary facets are removed. For usability reasons and as opposed to cone trees the child facets are not evenly distributed around the circular base. If the number of children exceeds num_f, all num_f facets are used. The remaining nodes of C can be mapped to facets on the fly as needed (see section 3.2).

The basic idea of CCT are nested cylinders according to the telescope metaphor. That means, for every node p listed on a cylinder with level $l=i$ a new cylinder is constructed for C of p, which has a smaller radius and belongs to the level $l=i+1$. All smaller cylinders of level $l=i+1$ are nested and hidden within the $l=i$ cylinder and can be pulled out to the right of the parent cylinder or collapsed as necessary. That means, only one path of the hierarchy is visible at once, represented by a number of ever decreasing cylinders showing all siblings of the corresponding level (see Fig. 2). So the basic layout is easy to comprehend with the x-axis representing tree depth, whereas siblings of C are displayed along the y-axis.

The second idea is the parallel display of all cylinders for $l=1$ in a horizontal fashion like being put on a stick. This increases usage of screen space. As soon as one of the $l \geq 1$ cylinders is pulled out by the user, it would interfere with its right sibling cylinder. That is why all other cylinders except the expanded cylinder and its parent cylinders are squeezed or horizontally scaled to accommodate the same screen space as before. The cylinder width of the current path is accumulated and subtracted from the available screen width. The result is divided by the number of all remaining $l=1$ cylinders to calculate the new scale and translation values for them. Since cylinders of the current path are displayed with their normal width, they provide the desired details. All other scaled cylinders provide context, at the same time being accessible for further navigation. If the maximum compression of the sibling cylinders of $l=1$ is reached as a result of deep hierarchies, the cylinders of the current path can be collapsed too, starting from the left side (the oldest parent). As soon as the user goes up the hierarchy, the cylinders are stretched again.

3.2. Navigation and Interaction

As already mentioned, besides tree visualization one of the goals of CCT is performing an action associated to a node. This can be opening a file, executing a program, displaying a web page etc. This is done with a simple click of the mouse on the facet representing the node.

Due to the cylindrical mapping of nodes not all items may be visible at once. So the rotation of cylinders is one of the possibilities for navigation. For that purpose the mouse is just vertically moved in the upper or lower area of the cylinder, which causes its automatic rotation in the desired direction, thus revealing the hidden nodes. Rotation speed increases at the edges of the cylinder to allow quick navigation while maintaining fine control in the central area.

If the number of children exceeds num_f on the cylinder, the nodes in the back can be dynamically exchanged before becoming visible to the user. Theoretically there could be an unlimited number of children on the cylinder. We call the concept *endless cylinder*. It is an improvement to the layouts of cone trees or spiral trees, where the visualized trees can be viewed from different directions. This inherently limits the possible arrangements of nodes. Since with CCT the user's viewpoint is fixed and cannot be changed, we can apply the "magic" on the back side of the cylinders.

The second navigation technique is needed to go down or up the hierarchy. Branch node facets provide this tree expansion functionality. Again, no mouse click is needed. If the mouse cursor stops in the center of a cylinder on top of a *p* node, the corresponding cylinder containing all nodes of *C* becomes visible and is translated to the right. All other $l=1$ cylinders are scaled to maintain the overall size. To investigate the just appeared nodes the user only moves the mouse to the right and continues navigation on the sub-cylinder. If the mouse cursor is moved on any other cylinder, the cylinders are again collapsed in an animated way. All movements are smoothly animated.

The CCT navigation needs just one click of the mouse when the desired node is reached. The user then clicks on it to perform the associated action (see an example in Fig. 3). Every single node of a hierarchy can be reached with just short vertical and horizontal mouse movements and without pressing any button. Studies like [6] report that axis-aligned movements out-perform "off axis" movements especially when using the mouse for menu selection. The CCT navigation technique is also fast due to the equal size of all facets and the same width of all non-scaled cylinders, which is calculated according to the available space. So the user maintains a simple mental model, memorizes mouse movements and thus employs muscle memory for navigation.

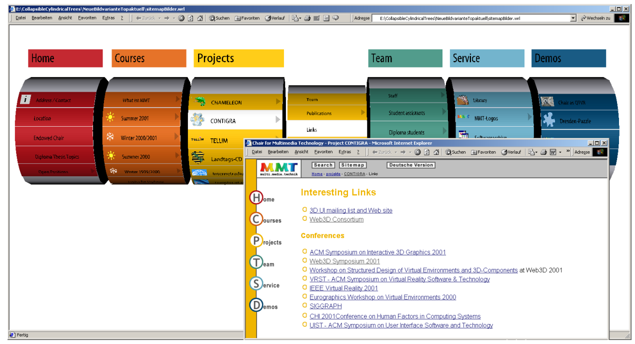

Figure 3. Display of a web page as an action associated to a node

3.3. Size

After having introduced the main characteristics of CCT we will explain what we mean with medium-sized trees. Due to the idea of presenting all $l=1$ cylinders in parallel, the number of root children num_{rc} is limited. For the sake of text readability it should not exceed approximately 7 nodes, depending on the desired text length. This value is derived from the assumption of an average text label length and typical screen width of 1024 pixel. Due to the *endless cylinder* concept every intermediate node *p* can have a theoretically unlimited number of children num_c. That means, all *p* nodes can have a high branching degree. However, if $num_c \leq num_f$, the implementation becomes easier and the display will be more comprehensible. The depth of the tree is also unlimited, although useful values are again suggested by the number of cylinders being displayed in parallel. We therefore recommend a value of $d = num_{rc} - 1 = 6$. To summarize, except the top hierarchy CCT clearly encourage breadth instead of depth in the tree. Research on menu performance suggested that breadth should be preferred over depth [13].

If you consider the recommended *d* and num_{rc} as well as $num_c \leq num_f$, then the maximum number of nodes in a tree would be calculated as follows:

$$maxnum = num_{rc} \cdot num_f^{\,d}$$

Given the recommended values $num_{rc}=7$, $d=6$, and $num_f = 20$ we get a theoretical maximum of $7 \cdot 20^6 = 4.48 \cdot 10^8$ nodes. The practical values are far lower, considering the balance of the tree and the fact, that leaf nodes not only occur at $l=d$, i.e. not every node within the hierarchy is a parent node. Practical values of a few hundreds to thousand nodes are typical for medium-sized hierarchies being displayable with the CCT visualization.

3.4. Presentation Aspects

This section describes the basic visual parts of the three-dimensional visualization and its degrees of design freedom. All root children are assigned to a facet or plate above the corresponding cylinders. They have the same parts as the normal node facets on the cylinder, but are completely independent in terms of geometry and appearance All cylinders consist of a body and the facets displayed on it. For the sake of a clear design and comprehensible visualization all cylinder and facet geometry is laid out identically (apart from the visual distortion in the scaled state). Facets are centered vertically on the cylinders. If their number exceeds $num_f/2$, the first facet is aligned near the top of a cylinder.

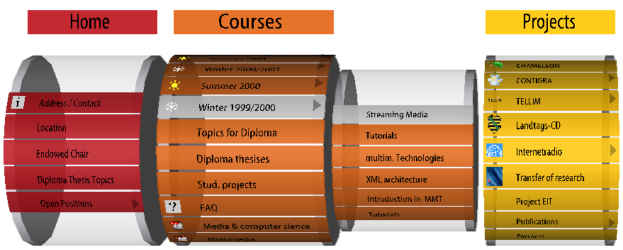

Figure 4. Modified visual attributes. Notice the increased number of facets on a cylinder.

A facet consists of a 3D text, a parent node indicator (e.g. a triangle like in typical menus), and a preview picture or item to the left of the text entry. This can be a thumbnail of the document being represented by the node, a node type indicator, a picture of a product, a visual icon for file size, ownership etc. For the text we recommend a condensed font like Arial Narrow. According to the attributes of every single node the text label, facet color, and preview picture are changed. Various meanings can be coded with facet color. In the sitemap example of Fig. 2 every node has the color of the main category. In Fig. 5 color of node facets alternates for a faster overview. Color can also indicate accessibility of a link, ownership, frequency of visits, document type and many more. That means, although nodes are treated equally in terms of geometric layout, semantics can be conveyed in a variety of forms (form and color of the item segment; color, font type, and size of the text; display of icons or pictures).

As far as the cylinders are concerned their radius, width and appearance can be altered. Changing the radius, more or less facets can be displayed on the cylinder. So one can for example adjust the number of facets to the perceptional favorable number of 4-8 items [13]. Through the perspective distortion at the edge of the cylinders one can still notice more items. The width of the cylinder determines the text length of the node labels. It must be balanced with the radius and number of $l=1$ cylinders to achieve an aesthetically pleasing and usable visualization. Fig. 4 shows a detail of the CCT sitemap with changed design parameters.

Current tree depth is visualized with the help of slightly scaled child cylinders. This value can be adjusted as well. Moreover, geometry and color of the child cylinders can indicate their relationship to the parent. The current position within the hierarchy is shown by a highlight of the active facet, which remains visible while the child cylinder is expanded. Unobtrusive sound can enhance the feeling of elegant interaction with the tree.

4. Application Examples

The CCT visualization and navigation technique is well suited for structured hierarchical information like taxonomies, family trees, web sitemaps, organizational charts, product hierarchies, media collections etc. It probably does not support very large hierarchies like complex file systems or huge 3D scene graphs. CCT are not made for discovering structures and relations in hierarchies, but were especially developed for fast navigation and visual comprehension of the visual structure.

The first idea for CCT rooted in the development of a World Wide Web sitemap of our research group. A result is shown in Fig. 2 and 3, containing 6 top level hierarchies. The tree contains approximately 300 nodes and still allows smooth navigation. Compared to the hyperbolic visualization depicted in Fig. 1 it allows a better overview and a reproducible navigation. The small icons depict previews of people, project logos, or symbols like the sun for the summer semester or the PDF icon for respective documents. The corresponding web pages are opened in a separate window.

As a second example we implemented a content navigation, as one can find for online manuals, tutorials, specifications etc. We used existing content from a study on internet radio and developed a solution, where the CCT navigation is displayed in the top frame and the selected chapters in the content frame below it (see Fig. 5). The current path is always visible through the highlighted facets. The design is rather functional and without pictures to resemble typical tables of contents. With this solution we tried to optimize the usage of available screen space.

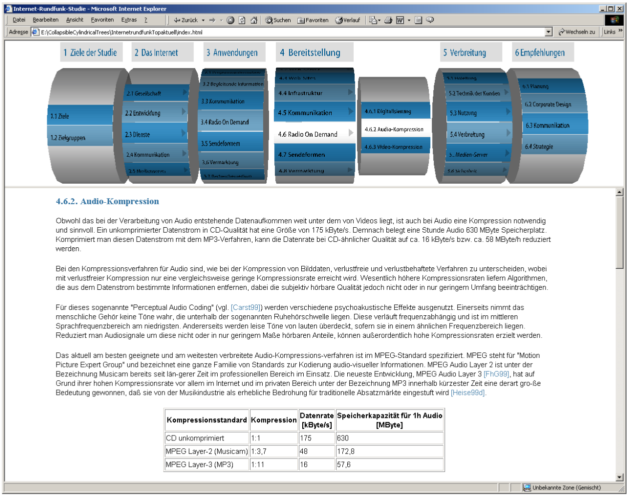

Figure 5. CCT used for content navigation of an online study on internet radio

4.1. Analysis

The two realized prototypes were not formally evaluated yet. However, early informal evaluation experiences show the suitability of this approach for quick navigation and interaction in medium-sized hierarchies. Moreover, people thought that they are fun to use. Due to the resemblance with traditional menu systems even users with no particular 3D or visualization experiences reported that they found CCT comprehensible. It seems reasonable to assume that almost no training is needed as opposed to other 3D tree visualization techniques. The fixed viewpoint, avoided information overload, short axis-aligned mouse movements, usage of muscle memory, and at most one needed mouse click are reasons for that.

CCT provide a compact and clearly arranged display of the first tree levels for a quick overview. Compared to menu systems more information are displayed at once and cylinders can contain more items than traditional menus. Compared to cone trees and similar techniques less information is displayed at the same time. We achieved a balance providing enough details but at the same time not burden the user with too many information, thus reducing the information load. Though 3D is visually exciting, the perspective and texture mappings degrade font quality, which is often a key aspect of information visualization [8]. With CCT both textures and text labels are always facing the user which guarantees maximum display quality and readability.

The most noticeable disadvantages of CCT are the restricted number of root children and the limited display of hierarchies, i.e. only the current navigation path is visible besides the two topmost levels. We compensated for that with a quick and elegant navigation.

4.2. Possible Enhancements

Looking at the content navigation example, screen space is efficiently used. If the full screen area is available like for a sitemap, an improvement would be the display of several cylinder bars in a vertical fashion like in a matrix. For every child *p* of the root node one row of cylinders could be constructed (level 1), where each cylinder is one of the children of *p* (level 2) and contains the grandchildren on its facets (level 3). This enhancement allows to view three levels at once, and still permits the navigation in the sub-levels.

Another technique to improve on the number of root children is a toroidal or donut arrangement of the top level cylinders instead of the linear display. The perspective distortion of the cylinders in the current implementation already hints at this idea.

5. Implementation

For the implementation of the sample CCT applications we used the Virtual Reality Modeling Language (VRML) [14] as the current standard for 3D graphics on the web.

5.1. VRML Representation

We defined three VRML prototypes for the CCT visualization. One contains the cylinder geometry together with invisible feedback areas in the upper and lower region of the cylinder. Touch sensors are associated with it and generate *move_up* and *move_down* events. The second PROTO contains the geometry of one single facet, the 3D text, the parent node indicator, and the textured face for the icon. All configurable attributes are defined as fields of the prototype, such as the color or text. The third prototype *mainFacet* is very similar to the facet PROTO, it contains the facet geometry and text for the root children (the plate above each cylinder).

The VRML file itself contains a transformation hierarchy with all cylinders and their parameterized facets, using the prototypes with the EXTERNPROTO statement. In addition to that a number of *TimeSensors*, ROUTE-Statements and an external JavaScript were defined for event handling. The *move_up* and *move_down* events of a cylinder activate the *TimeSensors*, which in turn call the corresponding functions of the script to rotate the cylinders.

Every facet has an associated sensor, which calls a script function that activates a timer to expand or collapse the corresponding sub-cylinder. At the same time all other cylinders receive events to adjust themselves and their sub-cylinders. After expanding or collapsing a cylinder its width is distributed to the parent, which calculates the width of all sibling cylinders. They in turn calculate the new positions.

5.2. Internal Tree Representation

It would be extremely difficult to develop the VRML representation manually. That is why we decoupled the tree structure and the 3D representation and automatically generate the VRML files from the internal representation of any hierarchy. A simple XML [16] grammar was created to code the trees:

```
<!ELEMENT NODE (NODE*)>
<!ATTLIST NODE
     label CDATA #REQUIRED
     target CDATA #IMPLIED
     color CDATA #IMPLIED
     icon CDATA #IMPLIED>
```

This grammar can be easily extended with other attributes or node types. Imagine for example a document type attribute containing values like text, image, video etc.

5.3. CCT Architecture

The CCT architecture is depicted in Fig. 6. At present we use a Java program to convert the XML document to the VRML file. Using the Document Object Model (DOM) the tree file is parsed and converted into an internal node hierarchy. A converter class then generates the VRML file with the help of the PROTOs and automatically creates the *TimeSensors* and ROUTE-Statements. After that the VRML file can be displayed in a web browser using a common VRML plug-in.

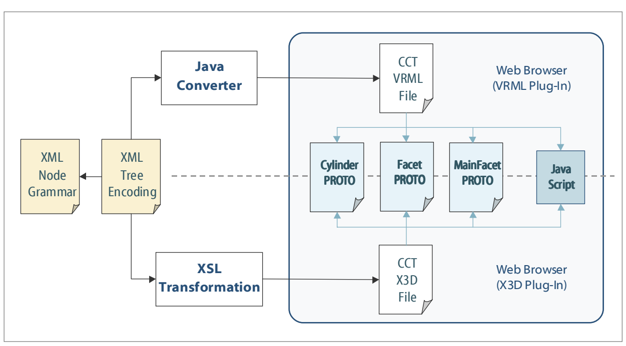

Figure 6. The CCT architecture: current and future implementation at the top and bottom

The successor of VRML97 will be X3D (Extensible 3D) [15]. Besides other improvements it provides an XML encoding for VRML. Therefore the more elegant solution would be to transform the XML tree file into an X3D representation with the help of the Extensible Stylesheet Language (XSL) [17]. The XSL Transformations Language (XSLT) allows the direct transformation of the XML document into the X3D file as shown in the lower part of Fig. 6. The second solution will be implemented as soon as stable X3D browsers will be available. At the moment the processing part is triggered manually after every change of the hierarchies, though it could been automated and realized on the fly with the help of a crawler process.

5.4. Performance

Due to the impressive advances in 3D graphics hardware and software the display of the sample trees with about 300 nodes is easy-to-use with an acceptable frame rate on a 750 MHz PC with a mid-range graphics accelerator. For an average hierarchy of 200 nodes approximately 40 cones, 200 facets, and more than 1000 VRML routes are being created. So the routes and the event handling are rather the bottleneck of this implementation. An advantage of the CCT approach is, that most of the cylinders can be hidden within their parent cylinders. This way they do not need to be drawn at all. At most $num_{rc} + d-1$ cylinders and $num_{rc} + (num_{rc} + d-1)\cdot num_f$ facets must be displayed at once.

If many preview pictures or icons are displayed, the performance slows down depending on the available texture memory and 3D graphics capabilities of the systems. A high-end graphics PC smoothly displays a fully textured hierarchy.

6. Conclusion and Future Work

We introduced a new visualization and interaction technique for medium-sized trees. Collapsible Cylindrical Trees allow an overview about the first two levels of a hierarchy plus the display of a navigation path within the tree. Thus the visualization balances detail and context through dynamically expanding and collapsing sub-hierarchies, represented by cylinders with facets for every child. We implemented a fast navigation technique, where only one click is needed to reach every single node and perform an action on it. The solution is suitable for being used on the web and allows aesthetically pleasing tree visualizations. Our sample applications show, that CCT and interaction with it are easy to comprehend.

The implementation must be refined and improved, since it is only a proof of concept yet. A formal evaluation and usability tests are a necessary continuation of our work. CCT performance could be measured in comparison to traditional menu systems or to other three-dimensional web navigation techniques. It will be interesting to find out, how CCT behave with more complex hierarchies and to implement the suggested enhancements.

References

[1] K. Andrews, "Visualizing Cyberspace: Information Visualization in the Harmony Internet Browser", *Proceedings of the IEEE Symposium on Information Visualization*, 1995, pp. 97-105.

[2] S. Benford, I. Taylor, D. Brailsford, B. Koleva, M. Craven, M. Fraser, G. Reynard, and C. Greenhalgh, "Three dimensional visualization of the World Wide Web", *ACM Computing Surveys*, Vol. 31, 4es, Dec. 1999, Article 25.

[3] M. Hemmje, C. Kunkel, and A. Willet, "LyberWorld - A Visualization User Interface Supporting Fulltext Retrieval", *Proceedings of ACM SIGIR '94*, 1994, pp.249-259.

[4] I. Herman, G. Melançon, and M.S. Marshall, "Graph Visualization and Navigation in Information Visualization: A Survey", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 6, No. 1, January/March 2000, pp. 24-43.

[5] B. Johnson and B. Shneiderman, "Tree-maps: A space-filling approach to the visualization of hierarchical information", *Visualization 1991*, pp. 284-291.

[6] G. Kurtenbach and W. Buxton, "The limits of expert performance using hierarchic marking menus" *Proceedings of the conference on Human factors in computing systems INTERCHI '93*, April 1993, pp. 482-487.

[7] J. Lamping, R. Rao, "The Hyperbolic Browser: A Focus + Context Technique for Visualizing large hierarchies", *Journal of Visual Languages and Computing, 7(1)*, 1996, pp. 33-55.

[8] J.D. Mackinlay, "Opportunities for Information Visualization", IEEE Computer Graphics and Applications, Vol. 20, No. 1, January/February 2000, pp. 22-23.

[9] T. Munzner and P. Burchard, "Visualizing the Structure of the World Wide Web in 3D Hyperbolic Space", *Proceedings of the VRML '95 Symposium*, 1995, pp.33-39.

[10] T. Munzner, "H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space", *Proceedings of the IEEE Symposium on Information Visualization (InfoViz '97)*, 1997, pp. 2-10.

[11] J. Rekimoto and M. Green, "The Information Cube: Using Transparency in 3D Information Visualization", *Proceedings of the Third Ann. Workshop Information Technologies & Systems* (WITS '93), 1993.

[12] G.G. Robertson, J.D. Mackinlay, and S.K. Card, "Cone trees: Animated 3D visualization of hierarchical information", *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems '91*, New York: ACM Press, 1991, pp. 189-194.

[13] Shneiderman, B., "Designing the User Interface: Strategies for Effective Human-Computer Interaction", 3rd edition, Addison-Wesley, 1998.

[14] The VRML Consortium Inc., "The Virtual Reality Modeling Language – International Standard ISO/IEC 14772-1:1997", 1997, http://www.web3d.org/technicalinfo/specifications/vrml97/index.htm

[15] Web3D Consortium, "X3D: The Virtual Reality Modeling Language - International Standard ISO/IEC 14772:200x", http://www.web3D.org/TaskGroups/x3d/specification/

[16] Extensible Markup Language (XML) http://www.w3.org/XML/

[17] Extensible Stylesheet Language (XSL) http://www.w3.org/Style/XSL/