Navigating Product Catalogs Through OFDAV Graph Visualization

Mao Lin Huang

Department of Computer Systems
Faculty of Information Technology
University of Technology, Sydney
NSW 2007, Australia
maolin@it.uts.edu.au

Kang Zhang

Dept of Computer Science
Box 830688, MS EC31
University of Texas at Dallas
Richardson, TX 75083-0688, USA
kzhang@utdallas.edu

Abstract. This paper describes the development of a 2D interactive visual interface for navigating online product catalogs that are commonly used for B2C e-commerce. We introduce an e-commerce framework for *Visual Online Shops* that use a 2D dynamic graph visualization as an interface, allowing buyers to interactively navigate through the large product hierarchies with a sense of information space. We discuss the applications of several dynamic visualization techniques, such as *Image-Map* and *OFDAV browser* that can be used to assist buyers in navigating large product information spaces dynamically and help them in selecting suitable products through the mouse clicks of appropriate graphical nodes in the visualization. The prototypes are written in *Java, JavaScript, PHP* and *MySQL* that simulate the online shopping experiences. They are applicable to any e-commerce online purchasing application.

1 Introduction

One of the major elements for online e-commerce is the online product catalog. It provides sellers with a content management system that stores, indexes, aggregates, normalizes, and distributes product information. It also provides potential buyers with an interactive interface that offers a multimedia representation of the product information as well as classification and navigation services.

There were many research works have been done for both aspects of the online product catalogs. For content management, a number of commercial products are developed, such *as CardoNet, Interwoven* and *OnDisplay*, and are used by many e-commerce web sites [1]. For catalog interfaces, many methods supporting product search and navigation have been developed [2]. These product information navigation methods are sometimes referred to as *shopping metaphors*. The usability of several product selection mechanisms was studied in [3, 2]. Also, the impact of different shopping metaphors on the effectiveness of online stores in terms of click-through and conversion rates was analyzed in [4, 5].

While these product catalogs can effectively assist the seller/buyer in managing, searching and accessing product information through the WWW, they usually do not provide a 2D graphic user interface that gives buyers a sense of information "space" when he/she is exploring the large product hierarchy. Instead they only provide buyers a series of textual lists placed in separate pages. Each list in a page shows only one level of the product hierarchy (see Figure 1). The buyer has to click through many pages to move down/up the product hierarchy in finding appropriate products he/she need. Thus, the entire structure of the product hierarchy was split into many small pieces and it was very difficult for buyers to percept the overall structure of the

product hierarchy by reading these textual lists. In fact, the effectiveness of this navigation mechanism in terms of click-through and human cognitive process was lower.

In this paper, we discuss the use of several graph visualization techniques to enhance the presentation of online product catalogs and address the above problems. We aim to develop high-quality catalog interfaces in terms of readability, understandability and comprehension by upgrading a broken 1D interactive space (a series of textual lists placed in many pages) to an integrated 2D interactive space (a overall navigational graph visualization). We used several dynamic viewing techniques, such as *Image-Map* and *OFDAV* [6] techniques to address the "small window" problem allowing buyers to navigate large product catalogs dynamically.

Figure 1: 800 Web Mall – A typical example of using traditional online product catalog for online e-business that provides buyers with a series of textual lists placed in separate pages for navigation. Each list in a page shows only one level of the product hierarchy.

This 2D visual structure of the product catalog greatly addresses the *product navigation problem* and allows buyers to visually browse through the entire product catalog with a sense of "Information Space".

The *OFDAV* technique [6] can be used to explore the huge product hierarchy online (especially for exploring distributed product catalogs that are partially unknown). It can quickly track the subset of the huge product hierarchy based on the

focus of the user, and provides buyers with a series of dynamic visual maps for guiding the online shopping activities.

The *Image-Map* technique, on the other hand, provides a click-free environment that reduces the click-through rate to zero during the navigation. This is because *Image-Map* supports the rollover function that only requires the movements of mouse pointer in viewing different parts of the product hierarchy.

2 The Framework of Visual Online Shop

The proposed *Visual Online Shop* is made up of several components. These components and interconnections among them can be described in Figure 2; details are below:

- **Dynamic Visualization:**

  A navigational visual presentation that automatically displays a sequence of the subset of product catalog following the buyer's orientation. It adapts several visualization techniques, such as *Image-Maps* and *OFDAV* – an on-line exploratory visualization technique that allow buyers to interactively navigate the large product hierarchy incrementally by swapping of views. It addresses the "small window" problem.

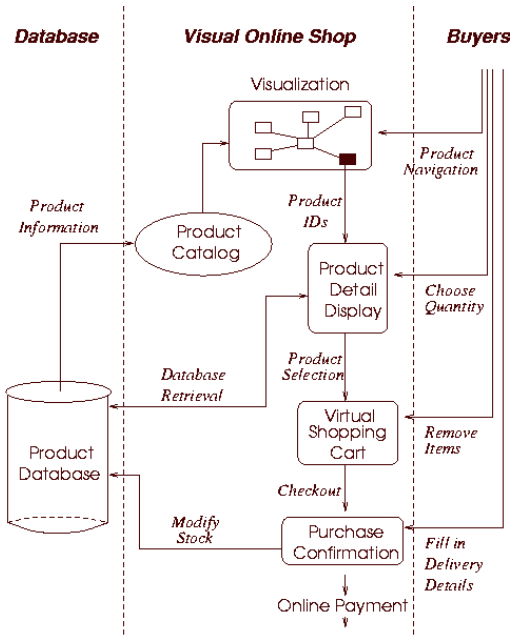


Figure 2: The framework of Visual Online Shop.

- **Product Database:**

  A relational database used to store product information, including all data fields and attributes associated with a particular product that are available for sale in the online shop. In our implementation, we used a MySQL database.

- **Product Catalog:**

A content management system that assembles, indexes, aggregates and normalizes product information from the product databases, and quickly distributes the product information.

- **Product Detail Display**

  A web page generated on the server side by a scripting language (we use a PHP script in our implementation) It retrieves the appropriate product entry from the database table in corresponding to the mouse-click on a particular graphic node in the visualization. It then displays selected attributes of one or more products in the page. Each product displayed has an associated "Add" button and an associated “Quantity” text field. The “Quantity” field allows buyers to fill in the number of product they want to purchase, while the “Add” allows buyers to add selected product to the virtual shopping cart.

- **Virtual Shopping Cart**

  The core element of the shopping-cart model of e-commerce that could be written in one of the server-side scripts, such as PHP, ASP or JSP. This shopping cart is responsible for controlling the buyer selection of products and the checkout operation. It shows how many products are already chosen and the total value of the chosen products in the cart so far.

  Buyers can empty the shopping cart (i.e. clear all selections made so far) by clicking on a "Clear" button, and remove items from the shopping cart by clicking on the "Remove" button. Users can also complete their shopping session by clicking on a button labeled "Checkout".

- **Purchase Confirmation**

  The component displays a purchase form asking buyers to fill in their delivery details (including name, address, suburb, state, country and email address).

  All these fields must be completed for the order to go ahead. The user completes the transaction by clicking on a button labeled "Purchase". The details of the order are sent via email to the email address given on the form. It then retrieves the product database, re-calculate the product stock and modify the in-stock field of those relevant product entries.

- **Online Payments**

  In this project, online payment is not a research focus. Therefore, we are not going to discuss it in further detail. There are many existing online payment systems are available, and the implementation for each one is different. In this framework, we use a *dummy* function that can be replaced with an interface to a chosen online payment system.

3 Dynamic Visualization of Product Catalogs

3.1 Image-Map Browser

Image-Map Browser is a visualization method that uses the techniques of *DHTML* and *JavaScript* to dynamically display a sequence of interactive pictures (image files). These pictures are merged into a large visual structure of the product hierarchy that allows buyers to navigate through it and select the products they need by clicking on some appropriate graphic nodes.

Under this scheme, A large visual category hierarchy of products is created in corresponding to the contents and structures of Product Catalog. However, since the

window size for display is limited, we are usually unable to display the entire visual structure (a large image) in one screen. Therefore, we divide this large visual category hierarchy (image) into *N+1* small images where *N* is the number of *the category/sub-category nodes* in the visualization. We then enable the rollover feature to all category/sub-category nodes using *DHTML, CSS* and some *JavaScript* functions to achieve the swapping between images. In particular, we use *onMouseover* and *onMouseout* event handlers to implement the rollover effect.

We define each small image as a clickable image map that contains a part of the entire product category. This allows the buyer to click on a particular *product node* (a leaf node in the lowest level of the hierarchy) in the image to retrieve the product database through a server-side script. The <MAP> tag defines the logic name of a small image, while the <AREA> tags define the mouse-sensitive region for each *product node*.

To ensure that only one image is displaying on the screen at a time, we use advanced layered structure to generate Image-Map. We define *N+1* layers, and each layer has a <DIV> tag associated with that defines the content of this layer. These small images are overlapping each other, but only one is "visible" and others are hidden which means “invisible”.

- **An Example**

Suppose that we are required to develop an Online Grocery Store in which the products for sale can be divided into five categories: *Frozen-Food*, *Fresh-Food*, *Beverages*, *Home-Health* and *Pet-Food*.

The layout of browser window (main page) is divided into three frames, the *Left Hand Frame*, the *Top Right Hand Frame* and the *Bottom Right Hand Frame*, as shown in Figure 3.

The Left Hand Frame displays the Image-Map Browser that shows a category hierarchy of products available in the shop. The default image shows only two top levels of the products hierarchy as shown in Figure 4.

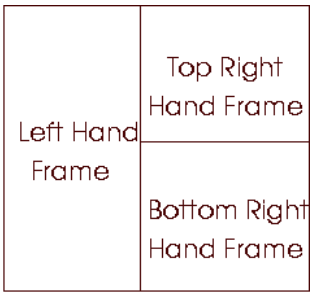


Figure 3: The window layout.

Figure 4: The default image showing the top level of the product hierarchy.

When the mouse pointer moves over to a particular category node *Fresh-Food* then the image displayed changes to a more detailed graph showing the substructure of this particular category as shown in Figure 5.

At the lowest level of the hierarchy, individual nodes are "live". They represent particular products and they are live links to a *PHP* file. When a buyer clicks on a particular *product node* it will bring up a *PHP* file showing the details of the product in the Top Right Hand Frame.

The Top Right Hand Frame displays a web page generated on the server side by a *PHP* script. It retrieves the appropriate product entry from the *MySQL* product table and then displays selected attributes of the product in a web page generated by the script. Each product displayed has an "Add" button that adds the selected product to the buyer's virtual shopping cart, which is displayed in the Bottom Right Hand Frame.

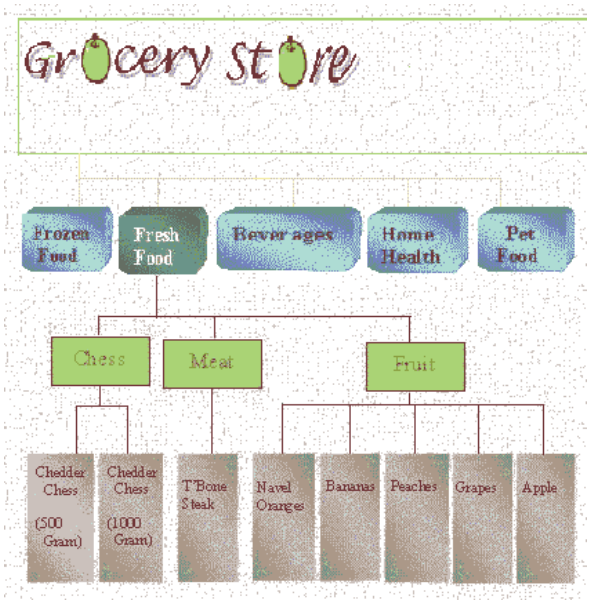


Figure 5: A display of substructure of product hierarchy under the *Fresh-Food* category.

The Virtual Shopping Cart shows the buyer how many products are already chosen and the total value of their selections so far. Buyers can empty the shopping cart (i.e. clear all selections made so far) by clicking on a "Clear" button. Buyers can also complete their shopping session by clicking on a button labeled "Checkout". When the Checkout button is clicked, the system checks that the buyer actually has products in the shopping cart, and if so it executes a *PHP* script whose results are shown in the Top Right Hand Frame. (If there are no products in the shopping cart a simple message is displayed in the shopping cart frame).

The Checkout component displays a purchase form asking the buyer to fill in their delivery details (including name, address, suburb, state, country, and email address). All these fields must be completed for the order to go ahead. The user completes the transaction by clicking on a button labeled "Purchase". The details of the order are sent via email to the email address given on the form.

3.2 OFDAV Browser

OFDAV Browser [6] is a 2D interactive graph visualization written in Java. It can be used to incrementally visualize a subset of the entire category hierarchy (visualization) of products by updating the visualization online. Exploration of the entire category hierarchy proceeds by changing of viewing frames. The transformation between two viewing frames is done smoothly through two types of animation: *Layout animation* and *Fade animation*. This greatly reduces the extra cognitive effort spent by buyers in re-forming the mental map of the picture after each transition.

OFDAV Browser has two working modes "Navigation" and "Select". Under the "Navigation" mode, you may explore the other parts of the category hierarchy (visualization) by selecting new focus nodes in viewing frames.

Under the "Select" mode, however, you can activate a *PHP* script appearing in the top right hand frame of the page by clicking on a graphical node (representing a product) at the lowest level of the hierarchy. The *PHP* script will then retrieve the appropriate product entry from the *MySQL* product table and display selected attributes of the product.

- **OFDAV Visualization**

Graph Information visualization concerns viewing relational data where the underlying data model is a graph. Most existing information visualization techniques have problems presenting very large data (such as product hierarchies) with millions or perhaps billions of nodes. The major problems can be summarized as following:

- Static visualization techniques predefine or pre-compute the layout. In many cases, the whole huge graph is not known. The online product catalog is updated automatically from time to time. So it may be impossible to pre-compute the layout of the whole visual structure.
- Pre-computation of the overall geometrical structure (global context) of a huge graph (with, perhaps, billions of nodes) is very computationally expensive. Many graph layout algorithms have super-linear time complexity, and in practice are too slow for interactive graphics if the number of nodes is larger than a few hundred.
- Pre-computation of the layout itself poses another problem. Since views are extracted from a predefined layout, changing views is a geometrical operation, not a logical operation. The user naturally thinks in terms of the logical relations in the application domain, not in terms of the synthesized geometry of the layout;

thus logical navigation of the product catalog through the entire visualization by using static techniques is difficult.

Online Force-Directed Animated Visualization (OFDAV) [6] is a new Graph Visualization technique introduced in 1998 that addresses the above problems. It uses an exploratory model of visualization allowing users to navigate huge graphs that are partially unknown, see Figure 6.

This technique provides a major departure from traditional visualization methods. It does not predefine the geometry for whole graph at once; instead it incrementally calculates and maintains a small local visualization (part of the global context) on-line, corresponding to the change of the user's focus. This feature enables the user to logically explore the huge graph without requiring the whole graph to be known.

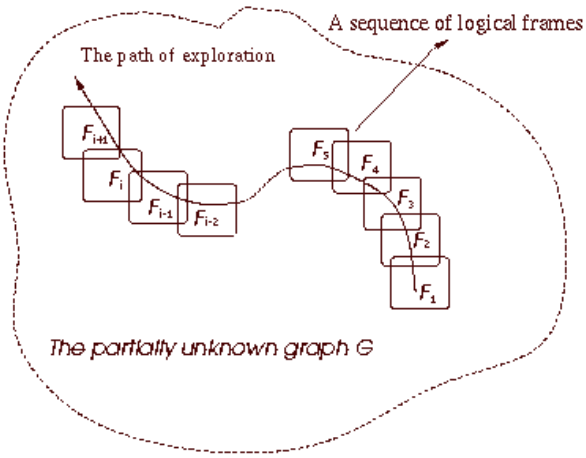


Figure 6: Online Visualization Model.

In *OFDAV*, the user's view is focused on a small sub-graph of a huge graph *G* at any point in time. The visualization of this sub-graph is called a *logical viewing frame* and is defined by its *focus* nodes. Conceptually, the focus nodes form a first-in-first-out queue with user's highest interest focus. The viewing frame is updated smoothly following the changes of the user's interested focus. The user can change focus by selecting another node in the viewing frame, but we do not anticipate the user's selection. However, we do assume that the user can always discover the neighborhood of the focus node.

It uses a *force-directed* graph drawing algorithm [6,7] to draw the sub-graph of *G* and the logical neighborhood around this sub-graph. The logical neighborhood of the focus nodes gives users a clear idea of where they are in the product hierarchy and helps them to decide where they should go next in a large information space.

It uses multiple animation to guide the user between views, reduce the cognitive effort and preserve the mental map of views.

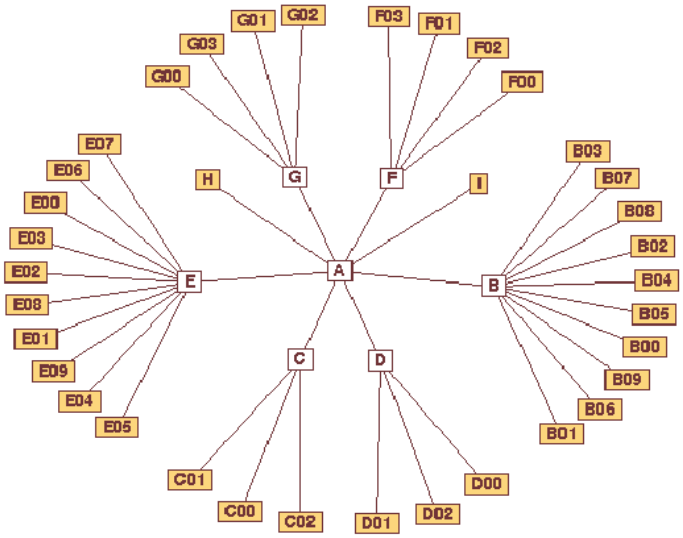
It adopts a graphical *history* tail that contains a sequence of previous focus nodes. It traces the sub-graphs that the user has visited. This assists in backtracking through the large product hierarchy.

- **An Example**

To illustrate how the system works to navigate the large product hierarchy, a simple example session is presented here.

In the previous example, we used an *Image-Map* Browser to assist the navigation of product hierarchy for an On-line Grocery Store. In this example, we are required to use an *OFDAV* Browser to explore a very large product hierarchy created for an e-Market. The window layout defined in Figure 3 remains the same.

The system first displays an initial viewing frame showing the top three levels of the product hierarchy at the beginning of the navigation (see Figure 7). It then updates the frame on-line as the buyer moves the focus by clicking on a succession of new focus nodes. We see that nodes of the initial frame are within 2 hops from the root



node “A” that are presenting top level categories and sub-categories of the products available in the e-Market.

Figure 7: An initial viewing frame showing the top three levels of the product hierarchy that is drawn by a modified spring algorithm.

Figures 8 shows the incremental change to the next viewing frame when the buyer selects the node labeled “F01”. We see that the node “F01” becomes a new focus node added into the focus queue; its immediate neighborhood appears into the frame and the old focus node “B” is deleted from the focus queue and its neighborhood disappears from the current frame.

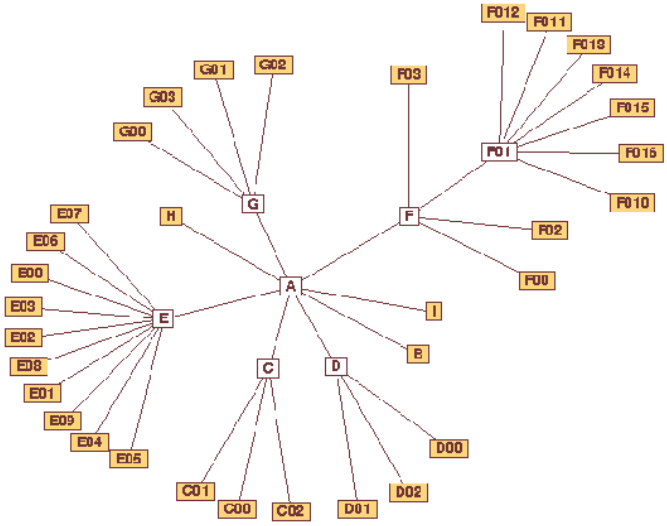


Figure 8: Navigating the product hierarchy on-line by changing the focus node to “F01”.

Figure 9 shows one frame further on after the buyer has selected the node ``F012". We see some new nodes appearing and three old nodes disappearing.

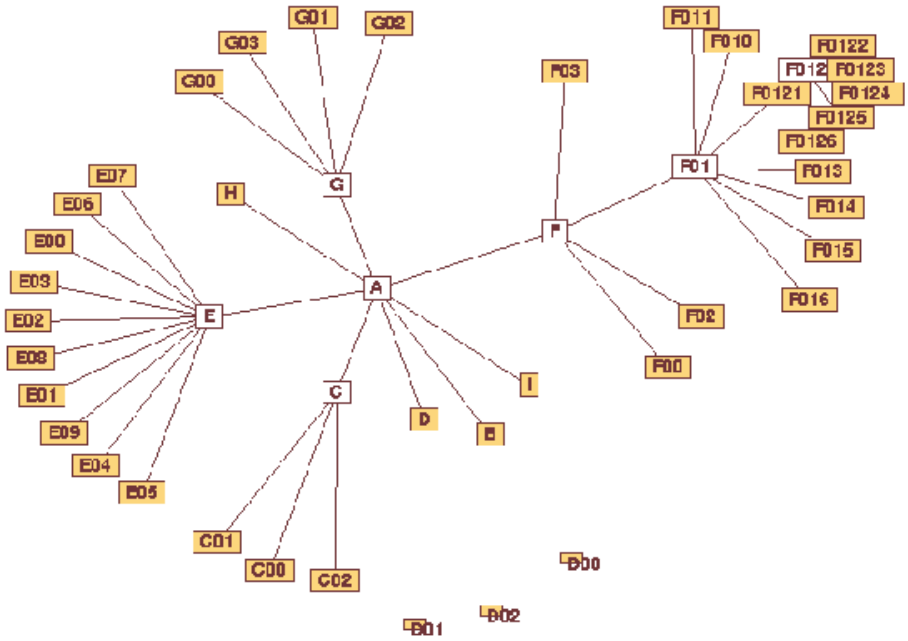


Figure 9: An example of nodes appearing and disappearing. We use fade animation to implement the addition and the deletion of nodes.

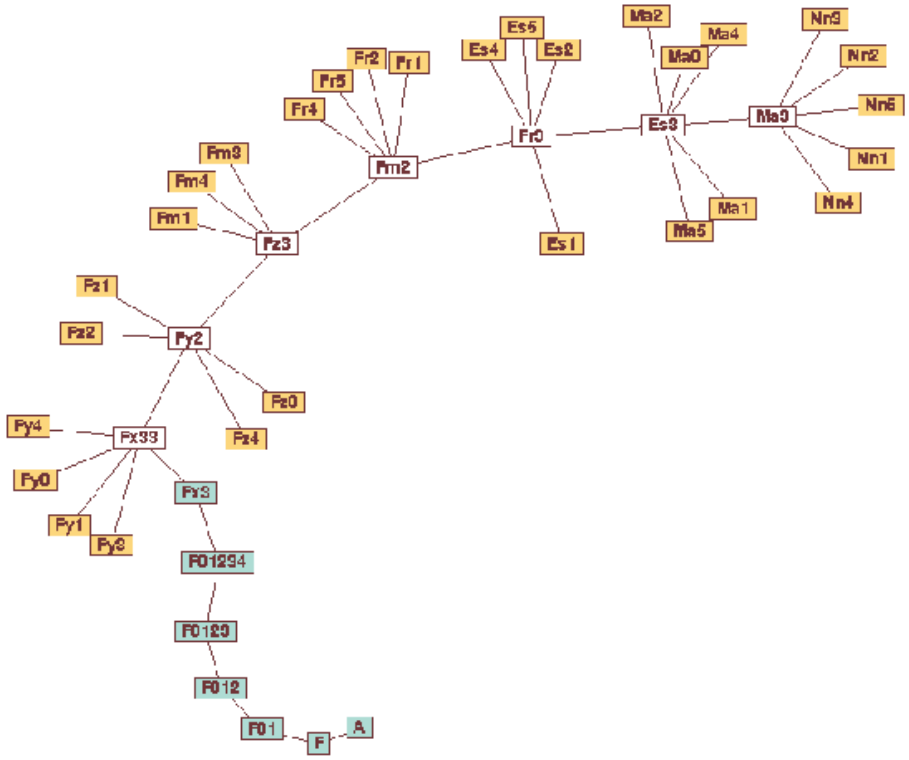


Figure 10: An example of viewing history nodes that indicate the location of where you are in a large product hierarchy.

Figure 10 shows the visualization ten frames further into the buyer's exploration of product catalog. Notice the history tail consisting of the previous focus nodes labeled ``A", ``F", ``F01", ``F012", ``F0123", ``F01234"and ``Fx3".

4 Conclusions

In this paper, we presented a new framework for the visualization of online product catalogs that provide an integrated 2D interactive space allowing buyers to visually navigate the product hierarchies with a sense of information space.

The traditional presentation of product catalogs provides buyers only a series of textual lists placed in several separate pages for navigation. This costs an increase of buyer’s cognitive effort spent in learning the overall structure of product hierarchies during the navigation. This navigation mechanism also requires a high rate of click-through for the navigation, as the navigational structure of the product category was implicitly presented in many separate pages. The parent-child relationships in the

product hierarchies were presented by embedded hyperlinks that the navigator cannot see directly.

In this paper, we attempted to use graph visualization techniques to address the above problem. We discussed the applications of several visualization techniques, such as *Image-Map* and *OFDAV browser* that can be used for dynamically navigation of product information.

However, this is just a beginning and there are a lot of research works in the future. We will conduct a usability study on this topic soon. We are also interested in investigating other visualization methods that could be more appropriate to be applied in the fields of e-commerce and B2B e-business. This will happen in the near future.

References

1. A. Neumann, "A Better Mousetrap Catalog", Business 2.0, February 2000, pp 117-118.
2. M. Stolze, "Comparative Study of Analytical Product Selection Support Mechanisms", *Proceedings of INTERACT 99*, Edinborough, UK, 1999.
3. E. Callahan and J. Koenemann, "A Comparative Usability Evaluation of User Interfaces for Online Product Catalog", *Proceedings of the 2nd ACM Conference on Electronic Commerce*, 2000, pp. 197-206.
4. J. Lee and M. Podlaseck, "Using a Starfield Visualization for Analyzing Product Performance of Online Stores", *Proceedings of the 2nd ACM Conference on Electronic Commerce*, 2000, pp. 168-175.
5. J. Lee, M. Podlaseck, E. Schonberg, and R. Hoch, "Visualization and Analysis of Clickstream Data of Online Stores for Understanding Web Merchandising," *International Journal of Data Mining and Knowledge Discovery*, 5(1), January 2001, Kluwer Academic Publishers.
6. M. Huang, P. Eades and J. Wang, "On-line Animated Visualization of Huge Graphs using a Modified Spring Algorithm", *Journal of Visual Languages and Computing*, 9(6), pp. 623-645, September 1998, Academic Press.
7. P. Eades, "A Heuristic for Graph Drawing", *Congressus Numerantium*, 42: 149-160, 1984.