

Creating Visual Browsers for Large-Scale Online Auctions

Mao Lin Huang and Quang Vinh Nguyen

*Faculty of Information Technology, University of Technology, Sydney,
PO Box 123 Broadway, NSW 2007, Australia*

Wei Lai

*School of Information Technology, Swinburne University of Technology,
P.O. Box 218, Hawthorn, VIC 3126, Australia*

1. Introduction

The Internet has dramatically brought more and more together buyers and sellers to the electronic marketplaces. Despite a rapid growing of interest and researches on the internet-based online e-commerce systems, the design of efficient mechanisms for navigating online product catalogues is still quite limited, especially for online auctions.

This chapter focuses on the investigation of information visualization approaches to the design of interactively visual interfaces that could guide the user to navigate efficiently through the online product catalogue for finding particular auction items the user wants in a large-scale online auction site. Our main concern is on its capability for helping shoppers to navigate and analyze online product information. Specifically, we concern about how well the application of these visualization techniques could speed up the process of information retrieval and activities taken for online auctions.

In the chapter, we will discuss the requirements raised for running online auctions as well as the technical issues on the design of graphical user interfaces and how we could use these graphical interfaces to help users navigate through the large on-line auction sites. We will introduce a very efficient visualization technique called *EncCon* as well as the design of graphic attributes that can be used to present the domain specific attributes of the auction items and the relational structures among these items, and these graphic presentations will provide users with a clear map showing the possible paths to the target items. We will demonstrate the effectiveness of our techniques by illustrating an online auction prototype that simulates the ordinary auction activities with the assistance of visualization.

1.1 Navigation of Product Catalogues in Online Auction

Over the past few years, electronic commerce (or e-commerce) has emerged as a dramatic new model of business (Bakos 1998). One of the greatest potentials of e-commerce is its ability to bring the effectiveness and unprecedented massive scale of buyers and sellers from all over the world. This property benefits both sides so that the buyers might have greater product diversity with potentially lower prices, and the sellers are able to reach a greater numbers of potential customers (Hahn 2001). At any time, through the online shopping stores (or auction websites), customers can learn more about the products, buy goods with electronic cash, and even have information goods delivered over the network. On the other hands, suppliers can reduce the overhead costs by investigating less in physical stores and distribution channels (Kim 1999).

An important precondition to the success of e-commerce systems, or specifically online auctions, is the construction of appropriate customer interfaces, from which online product catalogues can be retrieved, is

one of the key elements. Many extensive research projects have been done on both components of the online product catalogue including the content management and the catalogue interface. For content management, a number of products have been developed and used at commercial website such as *CardoNet*, *Interwoven*, *OnDisplay*, *Poet Software*, *Vignette*, etc (Neumann 2000). Various methods that support product search and navigation have been developed for catalogue interface such as those systems in (Callahan & Koenemann 2000), and (Huang & Zhang 2002).

Currently, a majority of commercial auction websites, including ebay.com, ubid.com, bidz.com, best-bidz.com and others, provide users with both the basic click-through navigation scheme, which is based on HTML pages, tables and lists, and add-on navigation aids. These auction websites typically categorize products as hierarchies so that users can retrieve auctioned items by hierarchically clicking on a number of HTML pages. The add-on navigation aids aim to provide navigation functions customizable to each user's need, such as search engines and personalized recommendations. In addition, multiple views of lists are usually used for the ease of seeking interesting items. These views include 'Current' (i.e. the default view of all items), 'New Today' (i.e. the new items posted today), 'Ending Today' (i.e. the items ended today), and others.

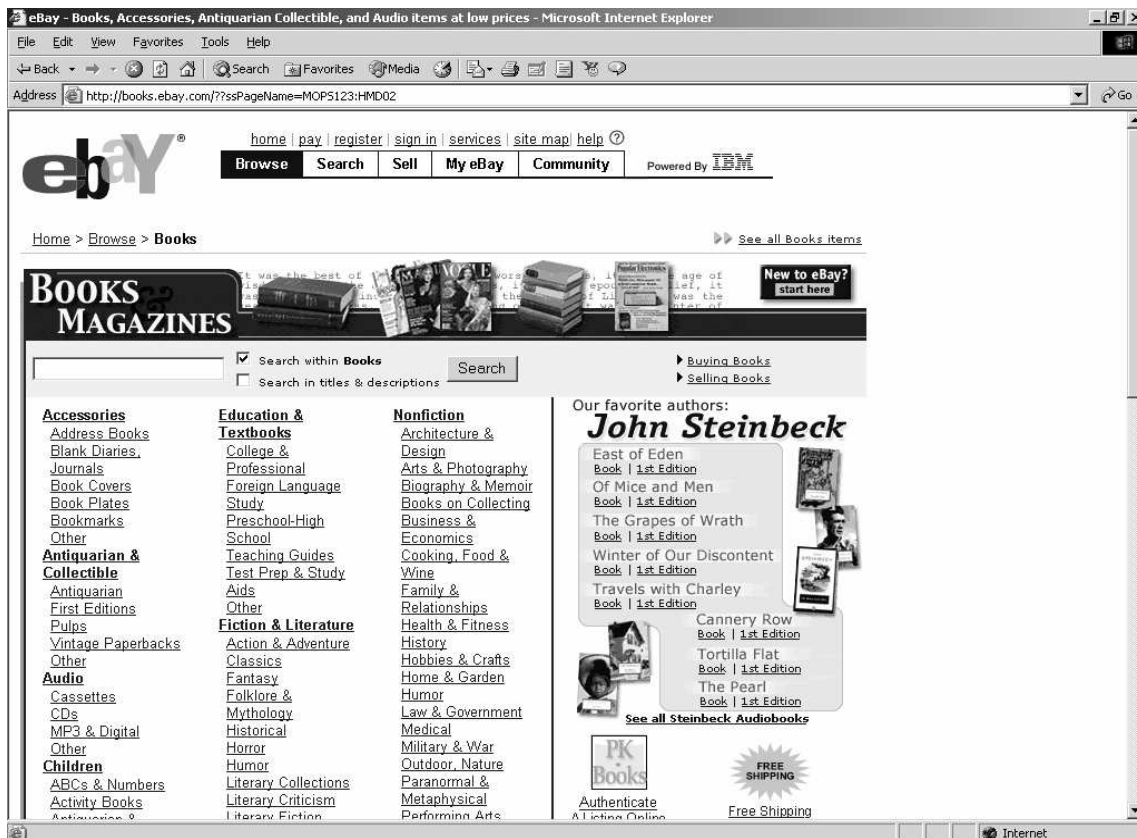


Figure 1. An example of the traditional text-based interface for online auction.

(Source from: http://www.ebay.com, accessed 20/03/2004).

Although the available navigation techniques can effectively assist sellers/buyers in searching and accessing product information over the *World Wide Web*, they mostly use the text-based interface that allows users to navigate by clicking-through several pages via URL links. Therefore, it could be difficult for the users to perceive the overall view of structure of the product hierarchy by reading these textual lists. Figure 1 shows an example of a text-based interface that is used on eBay's online auction website for browsing the product catalogue. The add-on navigation aids from those dominant commercial auction websites could help users to quickly locate interesting items. However, these aids, including search engines and personalized recommendations, might not be comprehensible enough for users who would like to do further research and analysis on several thousands items. Consequently, graphical visualization and navigation techniques are necessary for commercial auction websites so that they can effectively provide an overview of several thousands of items as well as quickly navigate to particular items.

Some newly developed visualization approaches have been proposed and implemented to enhance the presentation of product hierarchies for navigation. They aim to improve the readability, understandability, and comprehension of underlying hierarchical structure and to reduce the cognitive overhead for understanding the structure. These techniques are primarily for two-dimensional graph/tree visualization techniques to display and navigate the product catalogues. The technical detail of these visualization techniques can be found at (Huang & Zhang 2002), (Lee, Lee & Wang 2001), and (Inxight). However, there is still little research on the visual navigation of online auctions.

This chapter describes a new visualization approach for navigating large-scale online product catalogues of online auction stores. The visualization technique uses *EncCon tree* layout algorithm that displays the entire product hierarchy as well as a small portion of focused sub-hierarchy. We also introduce a new attributed visualization model that can be used to visualize the relational data with many associated attributes. We split the visualization model into two mappings between the abstract data and physical pictures, Geometric Mapping and Graphical Mapping. Users can browse through the entire product catalogue via *layering view* technique. A prototype was developed to demonstrate the effectiveness of this visualization technique in the area of online auction.

2. The Framework of Visual Online Auction Store

The proposed *visual online auction store* consists of several components. Within the scope of this research, we consider only the display and navigation components of the online auction. Figure 2 shows the components and interconnections among them in the context of online auction.

- **Product database** is a relational database used to store product information, including all data fields, attributes, and bidding information associated with a particular product that is available for auctioning. We used a MySQL database in our implementation.
- **Product Catalogue** is a content management system that assembles, indexes, aggregates and normalises product information from the product database, and quickly distributes the product catalogue information.
- **Catalogue Visualization** is a visual navigational interface that automatically displays the entire product catalogue's hierarchy, including categories, subcategories, and products. This component employs the *focus+context* visual layout and navigation mechanism [citations] that allows users not only to view the entire product hierarchy, but also to interactively browse down to a particular auctioned item.
- **Product Detail Display** is a web page generated on the server side by a particular scripting language, PHP, in our implementation, to show all the appropriate information of the selected

product. This page also displays the product's bidding information, and it allows the authenticated bidder to input the bidding price for the product.

2.1 The Procedure of Online Auction

- **The Role of Costumer:** our costumer is defined as an Internet User who accessed to our auction site for either obtaining information or conducting business as a seller or a buyer of a particular item for auction. For those costumers who want only to obtain some information about the auction we classify them as non-members. In contrast, we classify others who want to participate in the auction business process as members of the website.
- **Navigation/Bidding Session** this procedure allows users, including members & non-members to freely browse through a multi-level hierarchical structure of items for viewing and bidding at the site. This session allows registered members to browse, search and bid for items. Non-member visitors are only allowed to browse and search the items. In the item list page, a summarized description of found item will be displayed, including: Item Name, Current Bid (A\$), Starting Date, Closing Date and Number of Bids. Each item has a link to another page showing its complete information.
- **Sell Session** this session is for registered members only. It is used to place items for online auction. This page includes an online form allowing sellers to enter the details of their items, including Item Name, Item Image File, Starting Price, Reserved Price, Ending Date and Description of the Item, and add them under a particular Category and Subcategory. The information will then be saved into the database. The image file(s) will be saved somewhere in the server.
- **Registration Session** this session allows a visitor to register on the system by filling and submitting an online form. The Registration Form includes Username, Password, Confirm Password, First Name, Last Name, Email Address, Phone No, Street Address, Suburb, State, Postcode, Country and Credit Card Information. A confirmation email will be send to the user if her/his application is accepted. Once the registration is completed successfully, the user then becomes a formal member of the site.
- **Member Session** this session is for registered members only. In this system, a member can play both roles: the seller and the buyer, i.e. places items for selling or places bids for buying. The member session includes several functions allowing users to browse and bid on items, place books for selling, update his/her personal information, and change their password.
- **Administration Session** this session includes all functions for managing the system. The administration includes the following functions: Managing the members, including viewing the information of all members, and removing particular members, a function that checks the end-date for all items. If an item reaches its close date, an email will be send to the seller with information telling her/him whether the highest bidding price reaches the reserve price. If the highest bidding price is higher than the reserve price, then it will give the details about the potential buyers and the highest bidding price, and the money that she/he has to pay for the commission (This amount is 10% of the value of the sold item). An email will also be sent to the highest-price bidder to congratulate them for winning the item. It will also contain information about the seller. However if the highest bidding price does not reach the reserve price, then it will ask the seller to reset a new reserve price for the next period of auction.

3. Dynamic Visualization of Online Auction's Product Catalogue

The visualization of the product catalogue for online auction is implemented using the Java programming language. This visual browsing window does not replace entirely the traditional text-based interface, but provides extra assistance to users. In addition, the size of this applet window can be adjusted to suit each user's preference.

EncCon tree algorithm was used to lay out the structure of the product catalogues. In this visualization, nodes are used to represent the objects (such as categories, subcategories and auction items), while edges are used to present relationships among the objects or the relations among auction items and categories.

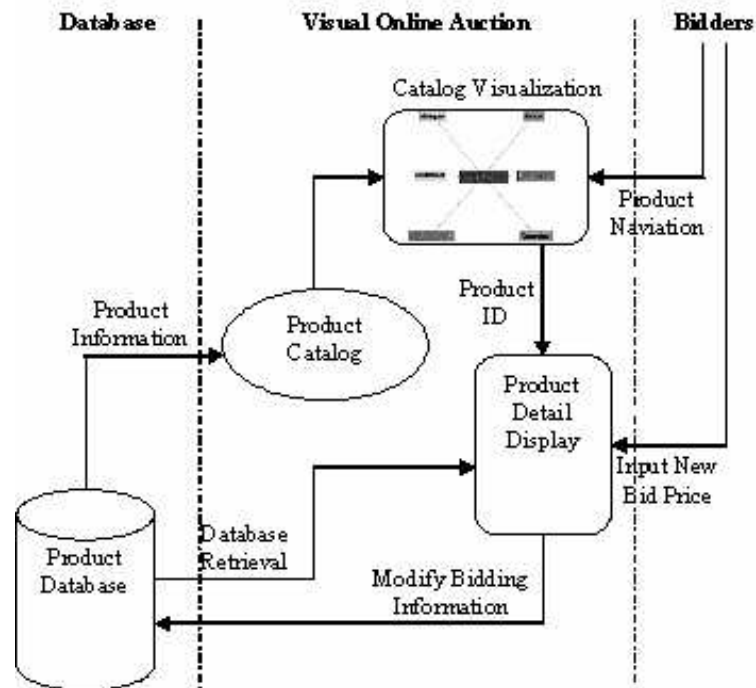


Figure 2. The framework of a visual online auction store

There are several alternative approaches in the design of a navigational structure for online auction sites. The navigational structure can be either *breadth-oriented* or *depth-oriented*. The *breadth-oriented* structure has the advantage of guiding users to their target item with the minimized number of mouse clicks, while *depth-oriented* structure enables the user to browse through more specific sub-category of interesting items effectively. However, the *depth-oriented* navigational structure requires more intermediate levels of retrieval (Hahn 2001). Although the use of appropriate navigational structure purely depends on the nature of applications, our auction prototype system uses *breadth-oriented* structure in its implementation. On the other hand, the navigational structure can be either single-only or multiple hierarchies. The use of multiple hierarchies may increase the chance of locating a target item of interest, but it often confuses the user because of its inconsistency through the site. The navigation structure used in our implementation is a single-only hierarchy.

It is desired that the chosen layout algorithm takes its advantage of geometric space efficiency, speed, and aesthetics. The above features and advantages of our layout technique ensure the capability of handling large or very large scale visualization with several levels of hierarchical views, i.e. a complex online auction's product catalogue with thousands of auction items. In other words, this could improve the scalability of the traditional interface. The layout also provides an overview of the entire category. This helps users have a better understanding of the overall structure of the product catalogue. Figure 3 shows an example of the visual navigational window (that displays all categories, subcategories and auction items) and the main window for browsing online auction's product catalogue.

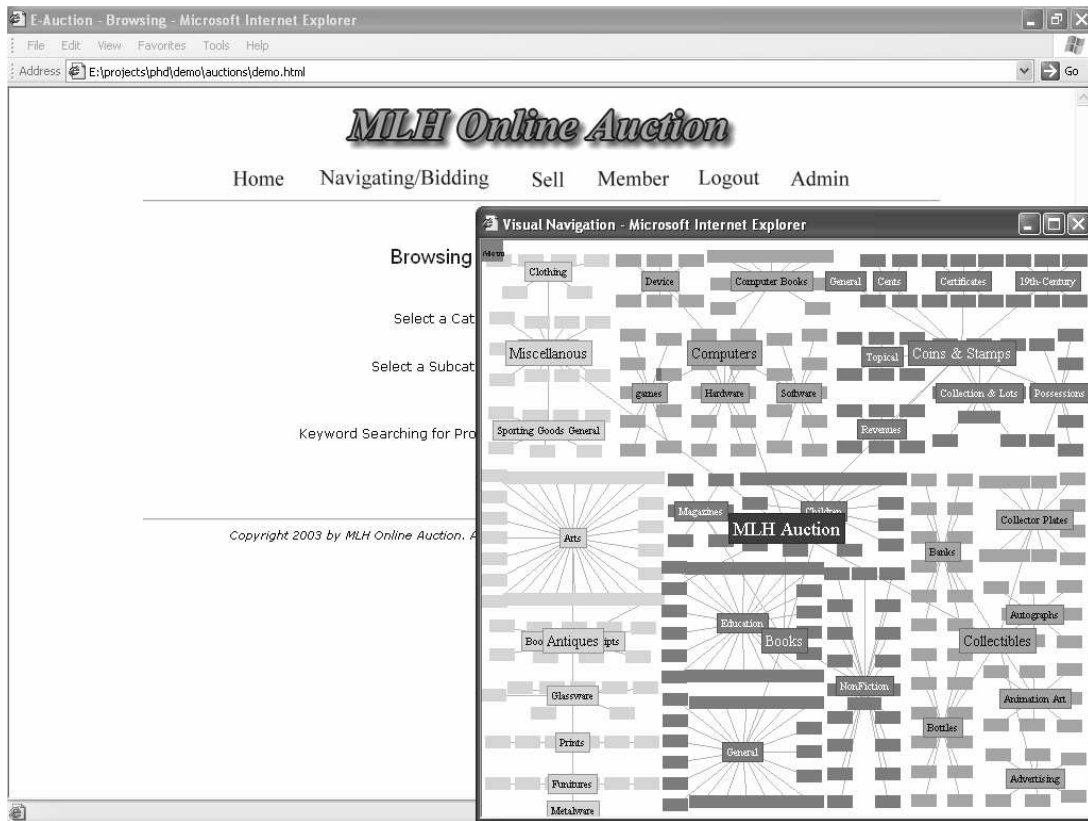


Figure 3. An example of the visual navigation window and the main window of the online auction's prototype.

We use our new *layering view* technique for navigating product catalogues. In the visualization, the layout of the overall context is overlapped with the layout of focused sub-hierarchy. Specifically, a semi-transparent technique is employed to display these dual views in the same geometric area, in which the focused sub-hierarchy is called the *detail-view* and the context of the hierarchy is called the *global-view*. This allows users to explore the entire hierarchy quickly by moving around the sub-hierarchies. Each visual interaction is accommodated by an animation in order to preserve the mental-map of the user during the navigation. In more detail, there are two states of the visualization: *normal* and *context*. Normally, the users' attention is on a particular sub-catalogue from the *detail-view*, and when the *context* state turns on, users' attention moves to the content of the *global-view*. At the *normal* state, the selected sub-hierarchy is displayed with no transparency, while the context is partly transparent and is displayed in brighter color (see Figure 5). At the *context* state, the context is brought from the back to the front and displayed with no

transparency, while the detail information is sent from the front to the back and displayed with brighter and partly transparent color (see Figure 6). These two views can be shifted interactively by using a left mouse-click on the background of each layer.

The visualization uses different colors to present items and subcategories of different categories. The categories and subcategories are also presented with bold boundary to identify auction items within the domain. These displays aim to improve the clarity of the visualization. The system also provides a mechanism to highlight the new products, ending-today products, and others. This aim to improve the overall display where the users can easily find the special items through the product catalogue (see Figure 4). We also provide an interactive menu allowing users to adjust the display to their preferred styles. When the mouse is moving over a node, the sub-hierarchy of the focused node is highlighted to emphasize the selection (see Figure 4). In addition, if the focused node is an auctioned item, brief information of this product will be displayed. This property reduces the navigation time since the users can quickly view information of the item from the visual navigation window. In our prototype, the brief information includes current bid price, starting date and closing date (see Figure 7). Finally, from the focused item, the bidders can also double-click on a particular product node in order to display all of the information associated with that auction item in the main window (see Figure 8).

Figure 4 shows a global view of a product catalogue of the prototype system, MLH Online Auction. From this figure, we can quickly identify a new product at the 'Computers-Games' categories. This item is highlighted by being painted with darker color at their front-end. The figure also indicates that the user is focusing on the category 'Computers'. Figure 5 shows the next display when the node 'Computers' is selected. You can see from the display of this figure that the subcategories and product of the category 'Computers' are enlarged and occupied the entire screen, while the previous context view in Figure 4 is reduced and sent to the background with semi-transparency. Figure 6 is the display when the global-view active state is selected. One can see that the display is reversed and that the context is brought from the background to the front and displayed with full colors, and the detail-view is sent from the front to the back and displayed with semi-transparent colors.

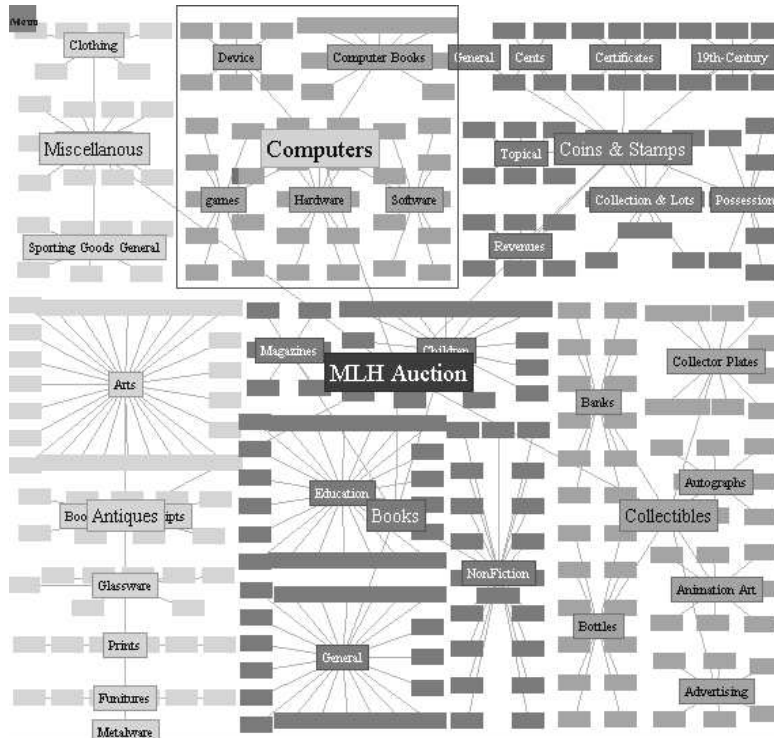


Figure 4. The global view of the entire product catalogue.

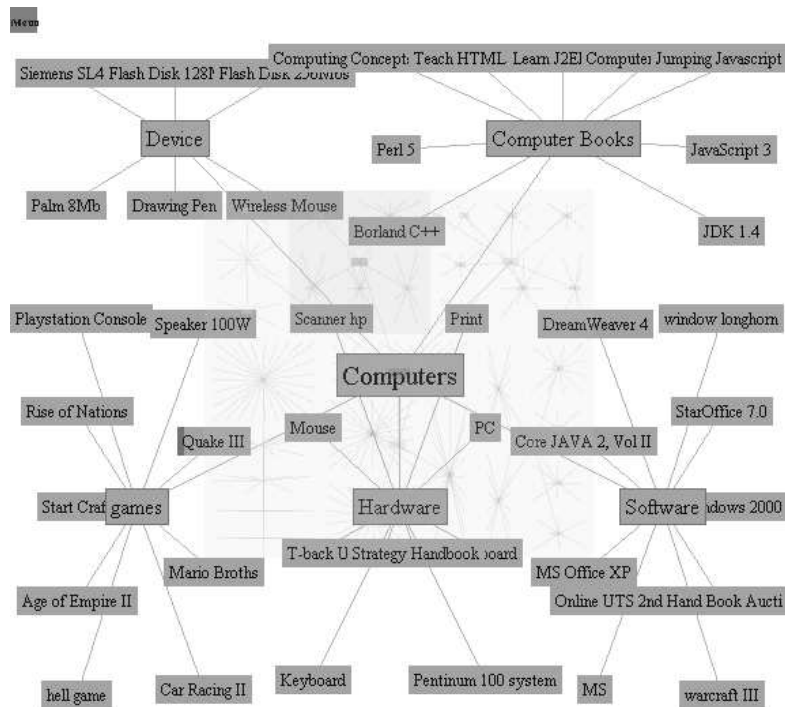


Figure 5. The display of all subcategories and auctioned items belonging to the category “Computer”.

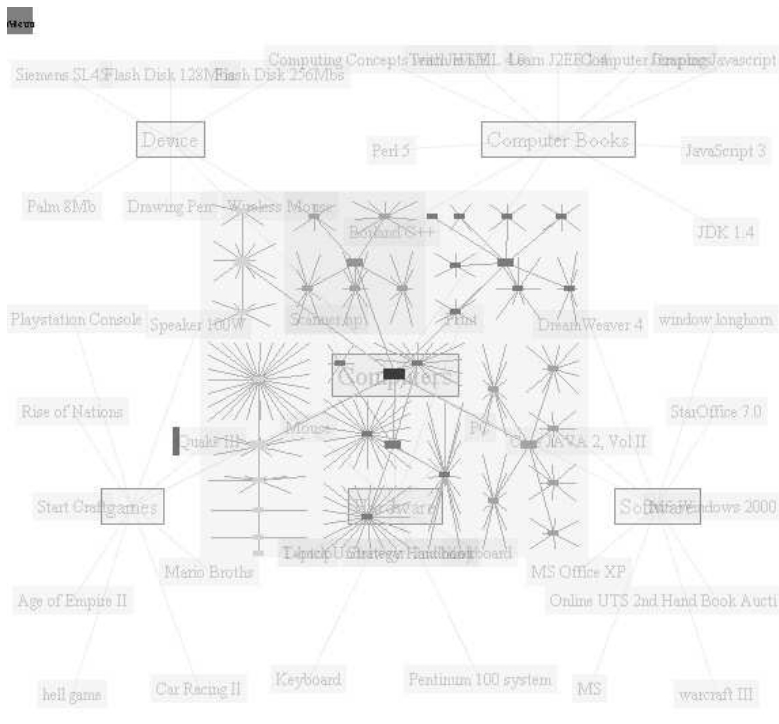


Figure 6. The display when the global-view is switched to active state.

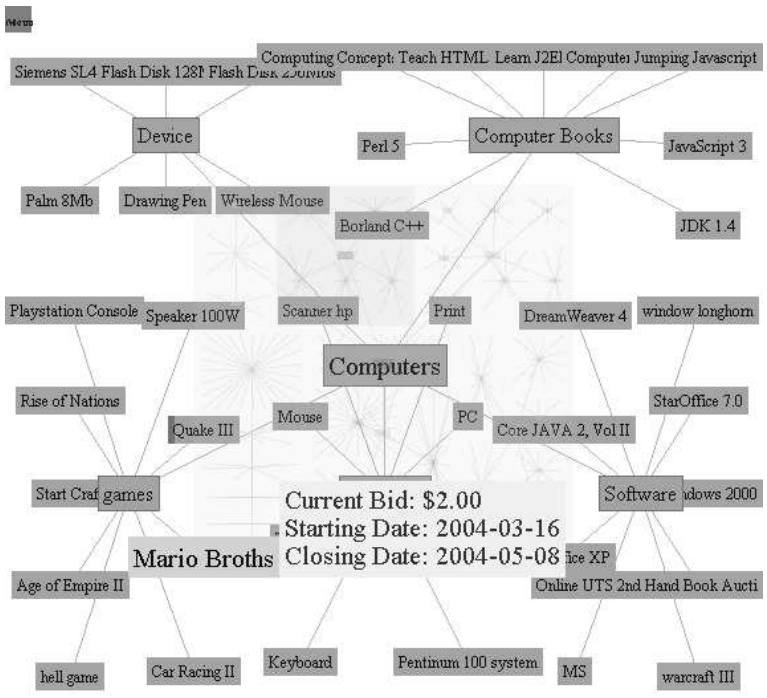


Figure 7. The display when the mouse is over a product. The system pop-ups a layer to show more detail of the auctioned item.

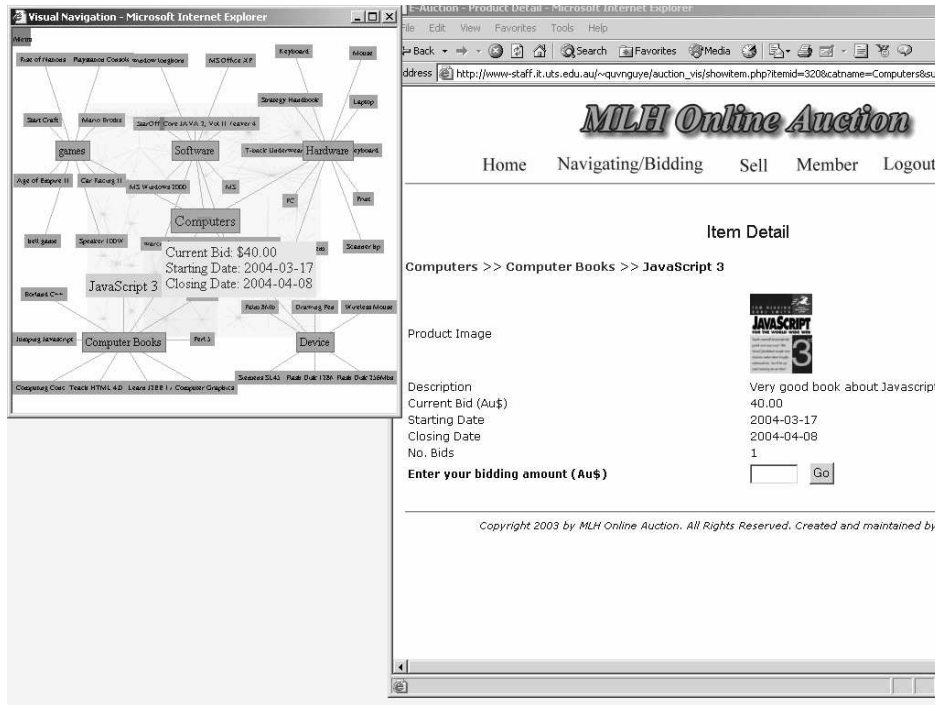


Figure 8. An example of the display of both visual navigation and the main window when the bidder double-clicks on the item “JavaScript 3” from categories “Computers-Computer Books”.

4. Attributed Visualization

Traditional techniques of relational information visualization (Battista, Eades, Tamassia, and Tollis 1999, Eades 1984) concern only the viewing of the abstract data and relations amount data items. They use *graphs* to model relational structures: the entities are *nodes*, and the relationships are *edges* (sometimes called *links*). For example, the structure of an Online Auction site can be modeled as a graph: the nodes are auction items, and a catalogue relationship is represented as a directed edge.

These graphs are typically drawn as diagrams with text at the nodes and line segments joining the nodes as edges. These drawings provide a readable and comprehensive graphic format of the relational structure for easy understanding.

The goal of the traditional *graph drawing* is to convert the abstract relational structure into a 2D/3D geometric space for visualization. This geometrical mapping involves the appending of two geometric attributes (x_a and y_a coordinates) to each *node* a , and four geometric attributes ($(x_a, y_a), (x_b, y_b)$) to each *edge* (a, b) in a graph. There are many applications in which the traditional graph drawing methods can be used to convey information, such as *family trees*, *software design diagrams* and *web site-maps*.

However, the data we want to visualize in the real world is much more complex than those can be simply presented in a 2D/3D geometric plane. For example, in the drawing of a web graph, an *edge* can only be used to present a hyperlink with four geometrical attributes ($(x_a, y_a), (x_b, y_b)$), and it cannot present the

communication protocol (such as *http*, *ftp*, *mailto* or *telnet*) associated with the hyperlink. The pure geometric presentation of a web graph does not show any further detail about a particular HTML document, such as the *access frequency* and *the connectivity* of the document.

Thus the traditional information visualization techniques and graph drawing methods tend to be inadequate to represent the relational data which has more attributes associated with it. We call this type of the data *Attributed Relational Data*.

Some alternative graph drawing methods have been proposed (Eades, Lai, and Mendonca 1994, Lai 1993, Kamada 1989, Kamada and Kawai 1988). They used a weighted graph model to address the problem of visualizing attributed relational data, such as the data describing the *E-mail Traffic* on the Internet (Eades, Lai, and Mendonca 1994). However, these methods still restrict the solutions of visualization within the domain of graph drawing, which concerns only *the geometric mapping* between the data and target pictorial representation. They do not take the advantages of rich graphics available on most of the PC/workstations to achieve the second mapping: *the graphical mapping* in which a set of rich graphical attributes is used to represent the domain-specific attributes of the data. Other systems, such as described by (Becker, Eick and Wilks 1995), concentrate on specific applications.

We are proposing techniques to visualize attributed relational data. We introduce our new visualization model that converts the attributed relational data into a target pictorial representation through two mappings: *the geometric mapping* and *the graphical mapping*. Our aim is to provide techniques that help human to distinguish the variety of attributes associated with the relational data through the use of graphical attributes.

This technique is implemented as a system for visualizing web graphs. It uses a force-directed graph drawing method (Eades 1984) to achieve the *geometric mapping*, and uses variety of graphical attributes to represent the metrics (attributes) associated with each *HTML document* and each *hyperlink*.

4.1. The Model of Attributed Visualization

A graph G consists of a finite set N of nodes and a finite set E of edges, where each edge is an unordered pair of nodes. A node μ is said to be *adjacent* to a node ν if (μ, ν) is an edge of G ; in this case, the edge (μ, ν) is said to be incident with μ and ν .

A *drawing* of a graph $G = (N, E)$ consists of two functions, $D_n: N \rightarrow R^2$ that associates a location $D_n(\nu)$ to each node $\nu \in N$, and a function $D_e: E \rightarrow C$ that assigns a curve $D_e(u, \nu)$ to each edge (u, ν) in E such that the endpoints of $D_e(u, \nu)$ are $D_n(u)$ and $D_n(\nu)$.

In practice, the nodes and edges have domain-specific attributes. For example, a hyperlink in a web graph has a protocol attribute (*http*, *ftp*, *mailto*, *etc*), and a node in a network may have a numerical attribute representing the amount of traffic passing through the node.

An attributed graph $A(G) = (A(N), A(E))$ consists of a finite set $A(N)$ of attributed nodes and a finite set $A(E)$ of attributed edges. Each attributed node $a(u) \in A(N)$ consists of $(u, DA(u))$ where $DA(u)$ is a set of domain-specific attributes associated with node u in the graph. Respectively for each attributed edge $a(u, \nu) \in A(E)$ we have that $a(u, \nu) = ((u, \nu), DA(u, \nu))$ where $DA(u, \nu)$ is a set of domain-specific attributes associated with edge (u, ν) in the graph.

In practice, the underlying graphics system has a set of available *glyphs* and a set of *linetypes*. Using this terminology, a drawing of a graph G maps each node u to a glyph g at a location $D_n(u)$ on the screen and each edge (u,v) to a linetype l with endpoints attached to the glyphs at $D_n(u)$ and $D_n(v)$.

Each glyph g (respectively each linetype l) has a set A_g (respectively A_l) of *attributes*. The attributes of a glyph include *shape*, *size*, and *colour*. The attributes of a linetype include *shape* (*Bezier*, *B-spline*, etc.), *thickness*, *linestyle* (*solid*, *dashed*, *dotted* etc.) and *colour*.

The attributes $\{a_g^1, a_g^2, a_g^3, \dots, a_g^x\}$ of a glyph g is a set of specific values of different types (*boolean*, *Integer*, *character*, etc) that associated with g . For example, suppose that a_g^i is a *shape* attribute associated with glyph g , we then can have a set of possible values, "*Rectangle*", "*Oval*", "*Curve*", "*Polygon*", etc of the *character* type assign to this attribute. If a_g^i is a *size* attribute, then we may assign a non-negative integer value (from 0 to 1024) to it as the actual number of pixels.

An *Attributed Visualization* (the underlying graphics) $AV(G) = (GLYPH(N), LINETYPE(E))$ of a graph G consists of a finite set $GLYPH(N)$ of *glyphs* (graphical entities) and a finite set $LINETYPE(E)$ of *linetypes* (graphical entities). Each glyph $g(u) \in GLYPH(N)$ consists of $(A_g, D_n(u))$ where A_g is a set of graphical attributes associated with g in the visualization, and the drawing $D_n(u)$ is a geometric location of u in a 2D plane with x, y coordinates associated with u . Respectively for each linetype $l(u,v) \in LINETYPE(E)$ we have $l(u,v) = (A_l, D_e(u,v))$, where A_l is a set of graphical attributes associated with $l(u,v)$ and the drawing $D_e(u,v)$ is a geometric curve representing the relation (u,v) in a 2D plane, which includes two endpoints for the curve (u,v) . Under this schema, the visualization problem becomes the problem of mapping between attributed graphs $A(G)$'s and attributed visualizations $AV(G)$'s.

5. Translation of Data into Pictures

In this section, we describe a framework for visualizing attributed relational data. In this framework, the translation of data into pictures includes two steps: *the geometric mapping* and *the graphical mapping*. We present the conceptual model of the data translation, and implement this model by using Java.

5.1. The Transition Diagram of Attributed Visualization

Figure 9 illustrates a transition diagram for attributed visualizing. This diagram is made up of several components. These components are described below:

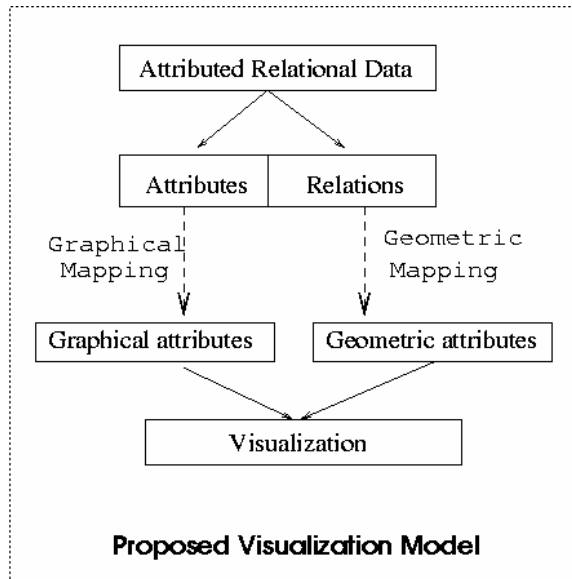


Figure 9. The Framework of Attributed Visualization.

- **Attributed Relation Data:** The real world relational data that we want to visualize. It includes a set of relationships among the data objects, and a set of domain-specific attributes associated with each data object and relationship (relation object).
- **Attributes:** A set of domain-specific attributes (properties) that are associated with a particular data object or relation object in a specific domain of the real world.
- **Relations:** Relationships among the data objects. As a certain type of the objects, a relation can associate a set of domain-specific attributes with it. These attributes determine the type of the relationship between two objects.
- **Graphical Attributes:** A set of graphical properties that are associated with a particular graphical object *glyph* or *linetype*, which is the graphical shadow of a particular data object or relation object.
- **Geometric Attributes:** A set of geometric properties that are associated with a data object /or a relation object. For example, in a 2D plane, a data object usually associates two geometric properties, x and y coordinates in a plane.
- **Visualization:** The final pictorial representation of attributed relational data. This includes representations for data, relational structure and domain-specific attributes that are associated with the data/relation objects. The basic elements of the visualization are *glyphs* and *linetypes*. Each glyph g (respectively each linetype l) represents a data object $v \in N$ (respectively a relation object $e \in E$). The attributes $A_g = \{a_g^1, a_g^2, a_g^3, \dots, a_g^x\}$ of a glyph g is used to represent the domain-specific attributes of a data object $v \in N$ that is graphically mirrored on g . Respectively the attributes $A_l = \{a_l^1, a_l^2, a_l^3, \dots, a_l^y\}$ of a linetype l is used to represent the domain-specific attributes of a relation object $e \in E$ where l is a graphical shadow of e in the visualization.

During the transition, the original data could have three different representations in each stage of the translation process.

The abstract graph representation

This level of the representation describes the abstract relational structure of the data objects and relations among the objects in the real world. These data objects are represented as a set of *nodes*, and these relations are represented as a set of *edges* in a graph model. Different applications of the relational data in the real world are translated into this universal representation.

The geometric representation

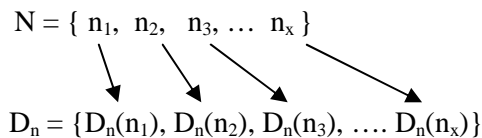
The second level of the representation is generated after the geometric mapping. The outcome of the geometric mapping is a geometric structure. A geometric structure means that each data object (node) is assigned with a geometric position and each relation object (edge) in the graph is assigned with two geometric positions, a *start-point* position and a *end-point* position. To make this geometric structure visible and allow users to view this geometric structure in a 2D plane, some simple graphical objects, such as *rectangles* and *lines*, are used to graphically display these geometric objects. This level of the representation is sufficient for the visualization, as the target representation, of *non-attributed relational data*.

The geometric + graphical representation

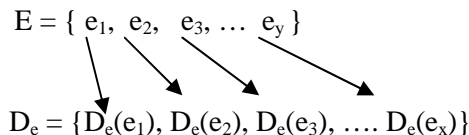
This level of the representation is the target representation - the actual pictorial representation for the visualization of *attributed relational data*. It is generated after the graphical mapping. The outcome of the graphical mapping is a picture. A picture consists of many graphical objects, and each graphical object can have many graphical properties (including geometric and graphical properties). These properties, such as *position*, *shape*, *size*, *color*, *brightness* and *z-coordinate*, can be used to represent the main features of the data/relation objects as well as the attributes associated with the data and relations in the real world.

5.2. The Geometric Mapping

The geometric mapping is the process of creating geometric representation of data. It converts the abstract data structure into a network with geometric *points* and *links* on a 2D plane. It assigns a geometric position (x, y) to each graphical glyph $g(v)$ that is the shadow of a data object v in a visualization. It also defines the *start-point* and the *end-point* for each graphical linetype $l(e)$ that is the shadow of a relation e in the visualization. Suppose that we have a finite set N of data objects and a finite set E of relation objects; then the actual geometric mapping from data set N to the drawing $D_n(N)$ can be illustrated below:



The actual geometric mapping from relation set E to the drawing $D_e(E)$ can be illustrated below:

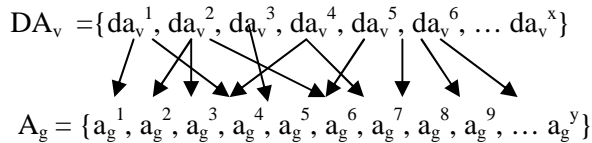


These mappings are one-to-one which means that there is only one geometric shadow created for each data object v and each relation object e after the mapping process. It assigns a location $D_n(v)$ to each data object v and assigns a curve $D_e(e)$ to each relation object e .

5.3. The Graphical Mapping

The graphical mapping is the process of creating pictorial representation of the data. It converts a set of domain-specific attributes DA_v of a data object v (respectively a relation object e) into a set of graphical attributes A_g (respectively A_l) of a graphical object g (respectively l) for visualization. Note that g (or l) is a graphical shadow (representation) of data v (or relation e).

Suppose that v is a data object associated with a set DA_v of domain-specific attributes. There is a graphical object g that can have a set A_g of graphical attributes associated with it. The actual graphical mapping from v to its graphical shadow g can be done as illustrated below:



Respectively the graphical mapping from a relation object e to its graphical shadow l can be done in the same way.

These mappings are many-to-many mapping which means that there are more than one graphical attributes can be used to represent a domain-specific attribute da_v^i (respectively da_e^i) that is associated with a data object v (respectively a relation object e). This also means that a graphical attribute a_g^i can be used to represent more than one domain-specific attribute in the visualization.

5.4. Implementation on Auction Web Graphs

The incredible size of the online auction sites accompanied with its large access frequency, introduces great challenges for information discovery. The web has no navigation structure or any sort of complete index of the content available. The problem of navigation across the huge auction site within a minimal time for finding particular items never be perfectly solved.

One of the ways in which web site designers are trying to address this problem is by providing what is commonly called "site-maps". The idea of a web site-map is based on the geographical metaphor of map. It is used to provide the user with an overall visual picture of the contents of a web site so that the user can easily navigate and obtain the interested information.

Since web site mapping is essentially a process of visualization of the information content of a web site, various approaches are adopted based on human visualization and perception capabilities. Each approach or technique for web site mapping has adopted one or a combination of these capabilities hoping to exploit them to help the user in navigation and comprehending the contents.

In web site-maps, a HTML document can be presented as a *node* in the graph, and a hyperlink can be presented as an *edge* in the graph. However, most of these approaches are only focusing on the pure geometric representations, rather than the graphical representations, of web graphs and they usually just assign some very simple graphical objects *glyphs* (perhaps some simple rectangles of the same size) to the *nodes* with the same graphical properties (such as *size*, *color* and *shape*) for visualization. These simple graphics are unable to represent the domain-specific attributes associated with the auction item, such as *the access frequency*, *bidding frequency* and *the connectivity* of an item page.

Therefore the user gains no knowledge about the domain-specific attributes of nodes (auction items) and edges (relationships), which are very important to the user where she/he is surfing through the visual structure of a auction catalogue graph.

We apply our new attribute visualization to create auction site-maps. We want to use a set of graphical attributes A_g (respectively A_l) associated with a glyph g (respectively l) to represent a set of domain-specific attributes DA_v (respectively DA_e) of an auction document v (or a hyperlink e), where g (or l) is the graphical shadow (representation) of a HTML document v (or a hyperlink e).

5.5. Graphical Attributes Associated with glyphs & linetypes

In our implementation, each glyph g in a web site-map has a set

$A_g = \{a_g^1, a_g^2, a_g^3, a_g^4, a_g^5, a_g^6\}$ of six graphical attributes. They are:

- $a_g^1 \rightarrow$ size of the graphic entity
- $a_g^2 \rightarrow$ background color
- $a_g^3 \rightarrow$ shape of the nodes
- $a_g^4 \rightarrow$ brightness
- $a_g^5 \rightarrow$ highlight/shadows
- $a_g^6 \rightarrow$ Z-coordinate at overlaps

In a web site-map, each linetype l has a set $A_l = \{a_l^1, a_l^2, a_l^3, a_l^4\}$ of four graphical attributes. They are:

- $a_l^1 \rightarrow$ length
- $a_l^2 \rightarrow$ thickness
- $a_l^3 \rightarrow$ brightness
- $a_l^4 \rightarrow$ Z-coordinate at crossings

5.6. Creating Pictorial Representation of Auction Graphs

The actual pictorial representation of a web site-map is generated after the *graphical mapping*, in which a set of domain-specific attributes associated with each web object v (or e) in the graph is mapped to a set of graphical attributes associated with graphical objects *glyphs* (or *linetypes*) in the visualization. This graphical mapping is a many-to-many mapping as illustrated in Figure 10.

Figure 11 shows the auction web graph. This graph is drawn using an *EncCon* algorithm (Nguyen and Huang 2005) (a kind of space-efficient drawing), without applying Attributed Visualization technique. In the visualization, we can only see the relationships (links) among the nodes (items), and cannot see any domain-specific attributes of auction objects in this visual site-map. Therefore, the user gains no knowledge about the properties (attributes) associated with these objects (items & links) in an auction site-map while she/he is browsing through the map.

Figure 12 shows the similar layout of an auction web graph as shown in Figure 11. However, it uses graphical attributes to represent the domain-specific attributes associated with web objects. This gives the user some ideas of what nodes are more important in the auction localities and worthwhile to have a look.

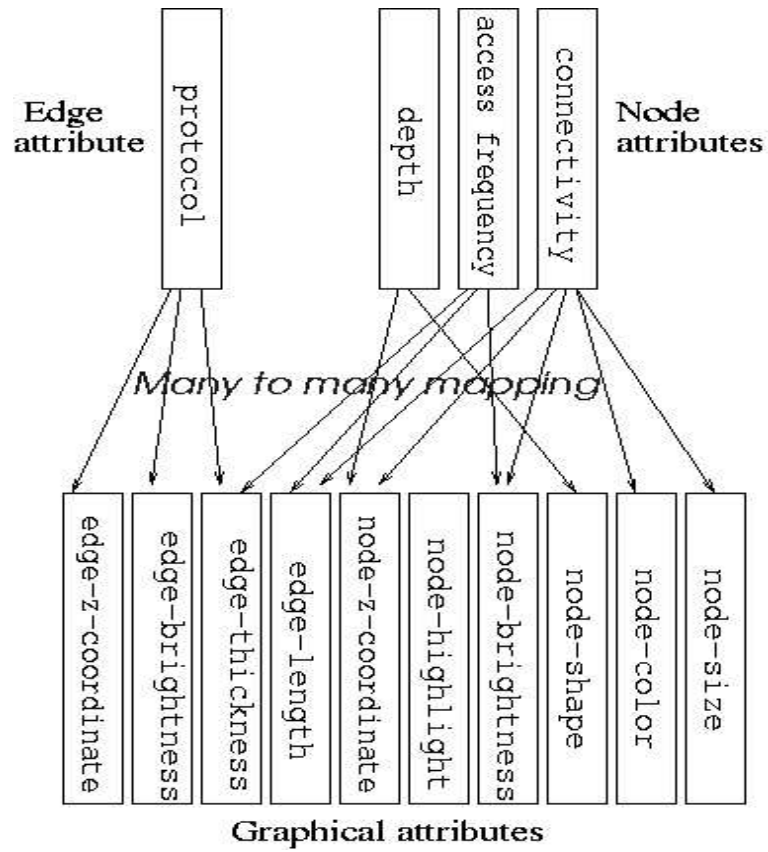


Figure 10. The Graphical Mapping.

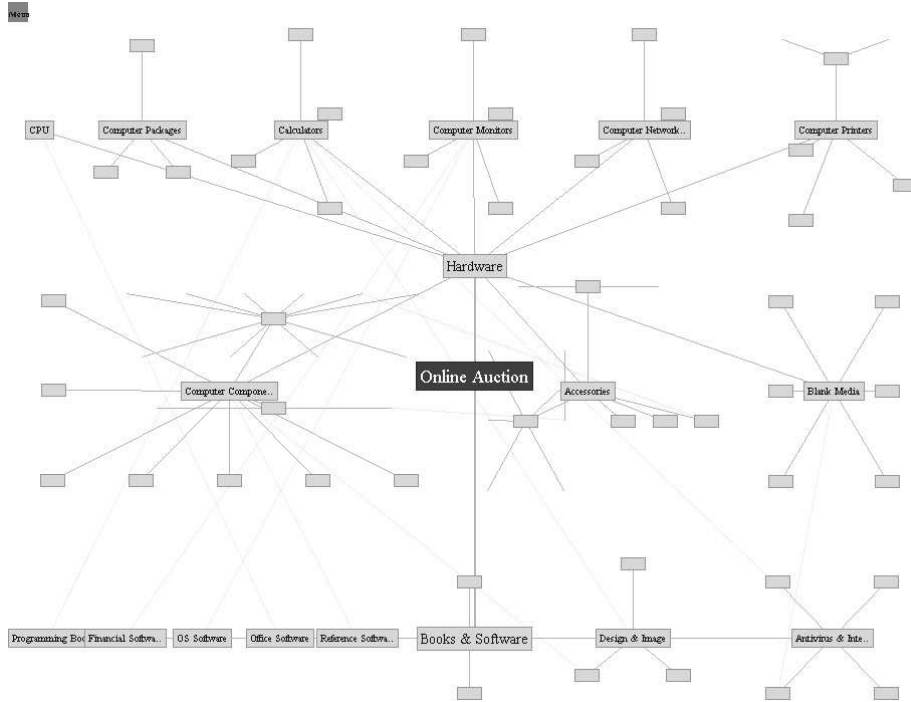


Figure 11. A screen of the original graph visualization of an auction website.

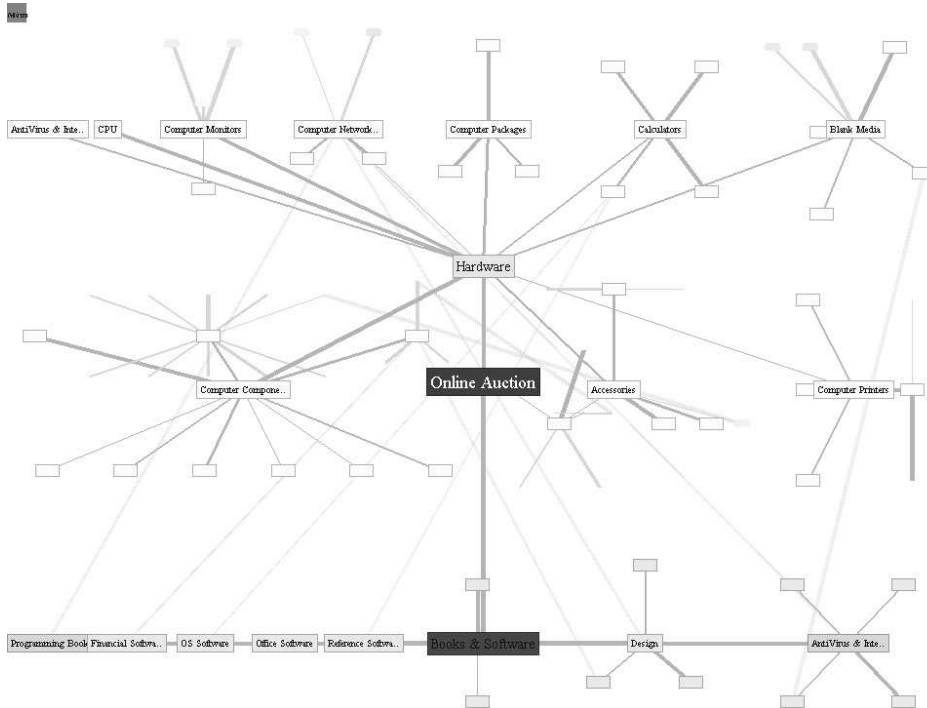


Figure 12. A screen of an attributed visualization of the same auction site as shown in Figure 5.2.

6. Conclusions

This chapter has presented a new dynamic visual user interface that appears as an additional window in the auction site for assisting the online auction process. This visual interface enables users to view and browse the auction catalogue with a large number of the auction items. A new *focus+context* viewing technique called *layering view* was employed to handle the overlapped display of both the *context view* and the *focus view*. This visualization combines the *EncCon tree layout* and the *layering view* method to provide users with a visual aid for the fast and effective product navigation.

In addition to the *EncCon* layout method, we also use a new attributed visualization to visualize not only the relational structure of the data, but also the domain-specific attributes that are associated with the data. The visual representations of these attributes are very important for users, who are browsing through the visual structure, in understanding of the data objects and the relationship objects that are represented as a set of *glyphs* and *linetypes* in a picture. Pictures produced by attributed visualization are much meaningful than those produced by traditional information visualization techniques.

The use of attributed visualization for auction website mapping can improve the quality of visual auction catalogue browsing. Under this scheme, the user can gain the knowledge about each node (item) in the auction web locality from the visual map. This will directly help the user in making up her/his decision of where to go next for finding particular items, while he/she is interactively browsing through the website via the visualization.

References

- BAKOS, Y. (1998): The Emerging Role of Electronic Marketplaces on the Internet, *Communications of the ACM*, 41(8): 35-42.
- BATTISTA, G., EADES, P., TAMASSIA R. & TOLLIS, I. (1999): *Drawing Graphs: Algorithms for the Visualization of Graphs*. USA, Prentice Hall.
- BECKER, R., EICK, S. and WILKS, A. (1995): Visualizing Network Data. *IEEE Transactions on Visualization and Graphic* 1(1): 16 - 28.
- CALLAHAN, E. and KOENEMANN, J. (2000): A Comparative Usability Evaluation of User Interfaces for Online Product Catalog, *In 2nd ACM Conference on Electronic Commerce (EC-00)*, ACM Press, pages 197-206.
- EADES, P. (1984): A Heuristic for Graph Drawing. *Congressus Numerantium* 42:149-160.
- EADES, P., LAI, W. and MENDONCA, X. (1994): A Visualizer for E-mail Traffic. *In 4th Int. Conf. Proc. Pacific Graphics'94 / CADDM'94*, pages 64-67.
- HAHN, J. (2001): The Dynamics of Mass Online Marketplaces: A Case Study of an Online Auction, *In SIG-CHI on Human factors in Computing Systems*, ACM Press, pages 317-24.
- HUANG, M.L. AND ZHANG, K. (2002): Navigating Product Catalogs Through OFDAV Graph Visualization, *In International Conference on Distributed Multimedia Systems (DMS'2002)*, Knowledge Systems Institute, pages 555-561.
- Inight. (2004): Hyperbolic Browser (Start Tree) for Online Shopping Stores, <http://www.inight.com> (accessed 20/03/2004)
- LAI, W. (1993): Building Interactive Diagram Application. Ph.D. thesis. University of Newcastle, Australia.
- KAMADA, T. (1989): Visualizing Abstract Objects and Relations. *World Scientific Series in Computer Science* - vol 5.

- LEE, J., WANG, P. and LEE, H.S. (2001): A Visual One-Page Catalog Interface for Analytical Product Selection. *In Electronic Commerce and Web Technologies, Second International Conference, EC-Web 2001*, Springer, pages 240-249.
- KAMADA, T. and KAWAI, S. (1988): Automatic Display of Network Structures for Human Understanding. Technical Report 88:007. Department of Information Science, Faculty of Science, University of Tokyo.
- KIM, J. (1999): An Empirical Study of Navigation Aids in Customer Interfaces, *Behaviour & Information Technology*, 18(3): 213-224.
- MACKINLAY, J. (1986): Automating the Design of Graphical Presentations of Relational Data. *ACM Transactions on Graphics* 5(2): 110 - 141.
- NEUMANN, A. (2000): A Better Mousetrap Catalog, *Business 2.0*, pages 117-118.
- NGUYEN, Q.V. and HUANG, M.L. (2005): EncCon: An Approach to Constructing Interactive Visualization of Large Hierarchical Data, *Information Visualization Journal*, 4(1): 1-21.