Dynamic Layout Management in a Multimedia Bulletin Board

Hyunmo Kang, Ben Shneiderman
Department of Computer Science,
Human Computer Interaction Laboratory,
University of Maryland at College Park
College Park, MD 20742 USA
{kang, ben}@cs.umd.edu

Gregory J Wolff
Ricoh Innovations, Inc.
California Research Center
2882 Sand Road Suite 115
Menlo Park, CA 94025 USA
wolff@rii.ricoh.com

Abstract

This paper proposes a novel user interface to manage the dynamic layout of multimedia objects in the Multimedia Bulletin Board (MBB) system. The MBB has been designed and implemented as a prototype of an asynchronous communication system that enables rich communication and collaboration among users of multimedia objects such as text, image, moving picture, sound, voice, web, office document, and other files. The layout properties of the multimedia objects on a board (e.g. x-y position, size, z-order, partial occlusion, explicit and implicit links, etc.) show important and useful information on the user dynamics occurring within a board. However, a fixed layout created and edited by multiple users may prevent users from recognizing and identifying useful information. This paper resolves this problem with a novel user-controlled layout strategy made visible with dynamic layout templates (DLT). Users can reorganize the objects to extract meaningful information related to time, source, geographic location, or topic.

Keywords: Multimedia Bulletin Board (*MBB*), asynchronous communication, collaboration, graphical user interfaces, layout management, dynamic layout template (*DLT*)

1. Introduction

The three systems most commonly used for facilitating asynchronous information flow within a group are electronic mail, the online bulletin boards, and searchable storage archives [8]. As information technologies have evolved, the capability of these systems has evolved through four generations of the type of message contents: text only, text plus attachment, rich text and embedded graphics, and embedded pointers to database objects [11]. The *Multimedia Bulletin Board* (*MBB*) was designed and implemented as a prototype of the next generation asynchronous information system that enables rich communication and collaboration among people using multimedia objects such as text, image, moving picture, sound, voice, web, document, and miscellaneous files. The main *MBB* display window with a control bar (Figure 2) is implemented as an ActiveX control and placed on a web page so that users can easily manipulate the boards with web browsers.

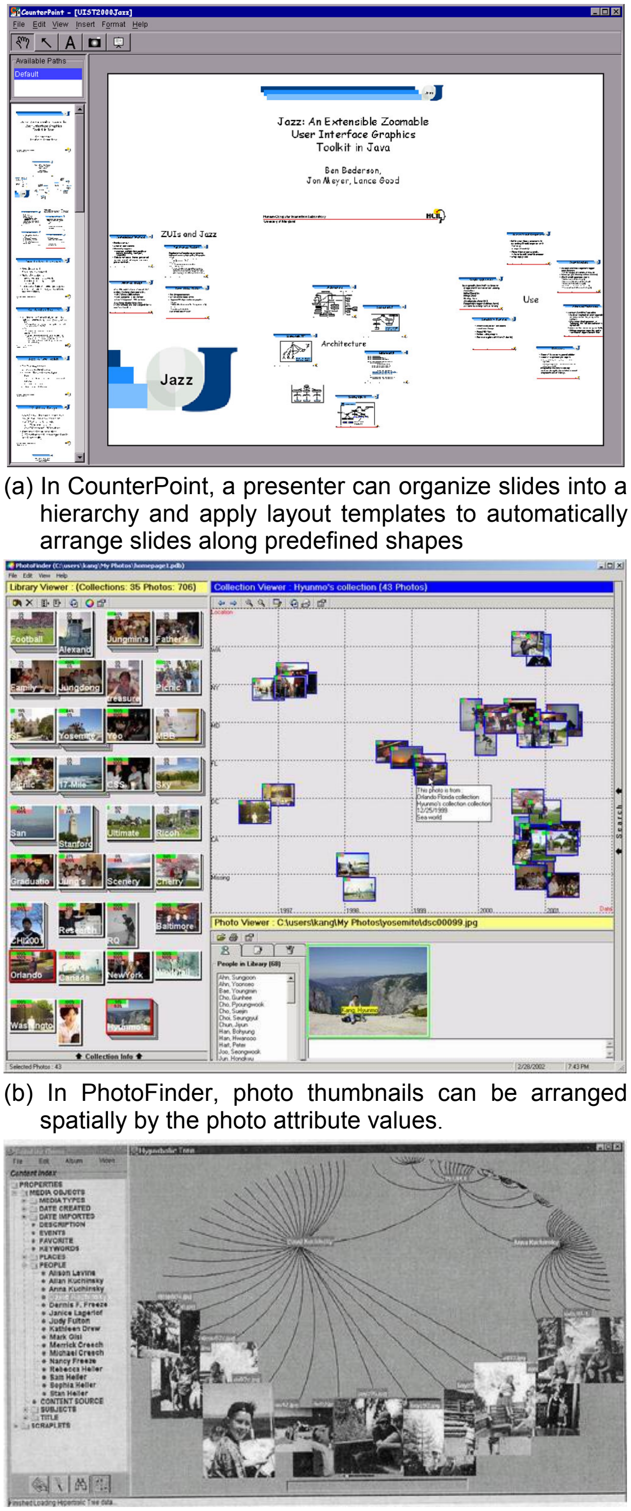
Figure 2 shows an example of the *MBB* messages on the topic of "Visiting San Francisco" composed by multiple users. One user triggered the discussion by asking for some good places to visit, such as restaurants and clubs. Several users answered the question and made comments on the posted messages with maps, images, web pages, moving pictures, texts, and so on. The layout properties of the multimedia objects such as x-y position, size, z-order, partial occlusion, explicit links, spatial distances among the objects, etc. represent useful and important information on the user-dynamics related to the specific story threads or the discussion topics on the board. However, the fixed layout (manually created by multiple users) may prevent users from extracting and identifying information. For example, users might be interested in such tasks as "I'd like to see the linear list of restaurants that people recommended with their related web pages and pictures", "I'd like to see all the landscape pictures posted, placing them around the San Francisco area map with reasonable size without overlapping", "I'd like to see all the messages posted on this board in chronological order using a weekly calendar", "I'd like to see how two specific people interacted with each other in this board", and so on. There is no easy way to perform these tasks with the fixed layout of a board, and even if the layout is not fixed it is a tedious and time-consuming job for users to change the whole layout manually every time they need a different layout. In practice, users create meaning by organizing the space [7]. There are many ways to lay out the same set of messages. There is no "right" layout; rather, the layout is very personal, has meaning for the individual who creates it. Therefore, the main challenge was to design and implement a novel user interface that makes it easier for end-users to restructure the layout of existing multimedia messages on the board and create new layouts of messages. As the number of the messages on the board increases the need and benefit of restructuring the layout of the board grows.

2. Related Work on Spatial Layout

With email lists, users expect to be able to reorganize their linear textual displays in ascending/descending order according to date received, sender name, topic, size, etc. With multimedia information, designers are beginning to understand what forms of two-dimensional displays are needed and explore novel possibilities. For example, MS Powerpoint allows only a row-by-column sequential ordering of slide thumbnails, but newer tools, such as Counterpoint [4] enable richer possibilities. Counterpoint users can have circular, oval, or hierarchical displays that reveal the structure of a presentation and allow greater flexibility in preparing and presenting a slide show (Figure 1(a)). Photo browsers are also expanding from row-by-column displays that are found in ACDSee [13], PhotoSuite [14], and web tools such as Ofoto [15]. PhotoFinder [10] offers four collection viewers that allow user control of size and placement or automatic placement by ranking, date, numbers of people, etc (Figure 1(b)). Canon's ZoomBrowser offers a dazzling spiral display that is attractive to many viewers and the FotoFile [1] uses a hyperbolic tree for photos (Figure 1(c)). Other innovations include 3-dimensional positioning of photos [6] and user fly-throughs that are appealing for some users, but disorienting for others.

Some studies focused on the model for window management and layout in desktop interfaces. Elastic windows [5] presents a browsing interface with hierarchical window organization and multiple window operations that allows users to organize web pages and restructure the information on the screen. Bell B. et al. [3] presents a general approach to the dynamic representation of 2D space that is well suited for tiled user interface layout. Beaudouin-Lafon et al. [2] presents several techniques that improve window management by extending the metaphor of overlapping windows. However, most researchers have addressed window and space management to maximize visibility and usability rather than constructing the new meaning through dynamic layouts of the content objects. Video Manga [9] system shows a good example of constructing new meaning through the spatial layout. It presents methods for automatically creating pictorial video summaries that resemble comic books. The selected keyframes of each video segments are sized by importance, and then packed into a pictorial summary.

Microsoft's Data Mountain [7] shows another possibility of using spatial layout for document management. In this system, users freely arrange document thumbnails on an inclined 3D plane textured with passive landmarks to take advantage of human spatial memory.



(a) In CounterPoint, a presenter can organize slides into a hierarchy and apply layout templates to automatically arrange slides along predefined shapes

(b) In PhotoFinder, photo thumbnails can be arranged spatially by the photo attribute values.

(c) In FotoFile, photo thumbnails can be arranged with hyperbolic tree for photo browsing

Figure 1. The Example Applications of Spatial Layout

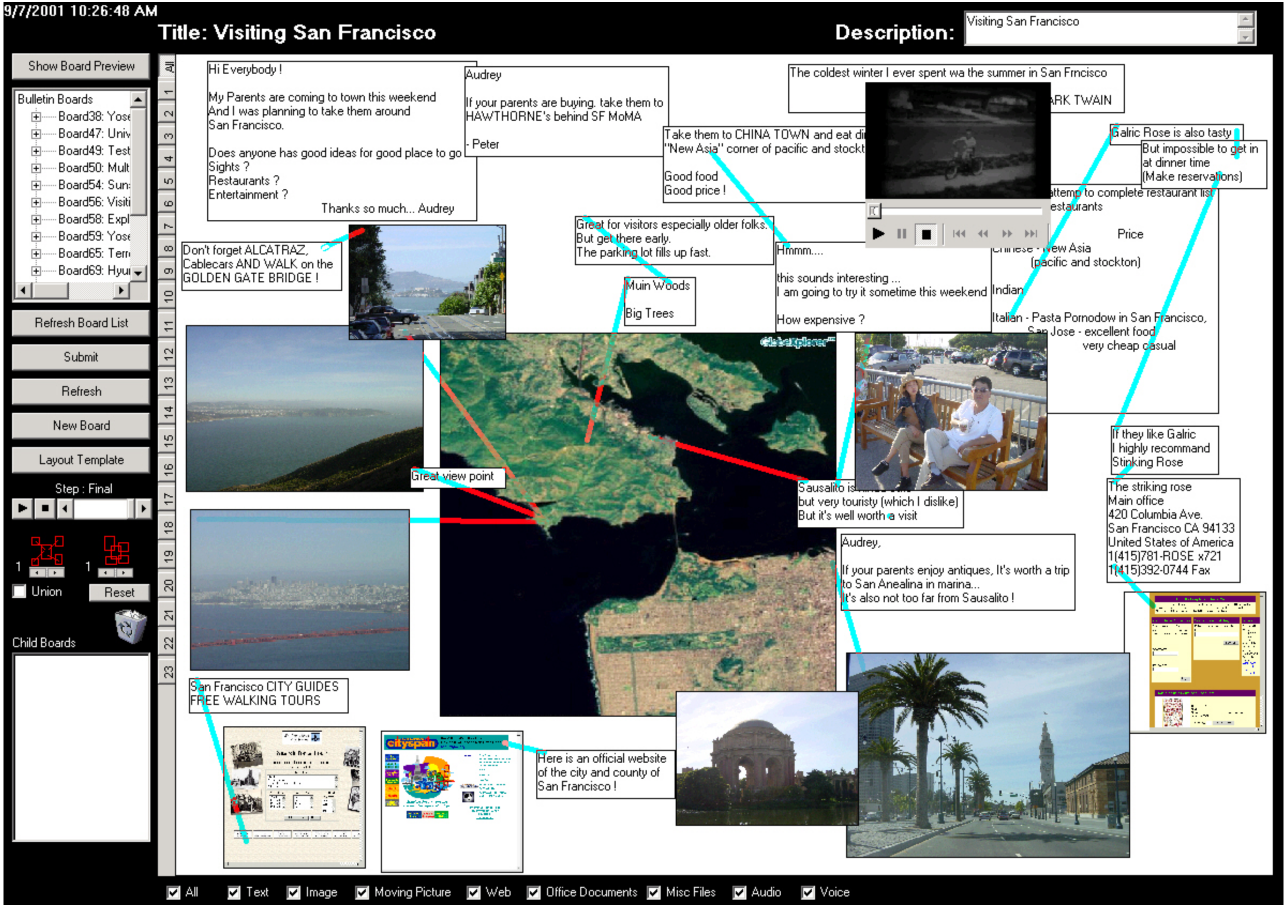
While many systems support the layout management of their contents to a certain degree, the demands of dynamic layout management in the *MBB* are worthy of special study. We believe that the *MBB* is a special case because users deal with heterogeneous multimedia messages and their initial layout, which are authored by multiple users.

3. Multimedia Bulletin Board

As mentioned in the previous section, the *MBB* was designed and implemented to enrich asynchronous communication and collaboration among people using multimedia objects. The multimedia objects supported by *MBB* can be classified into three categories; visual object, audio object, and file/URL object. The visual object category contains text objects, image objects, and moving picture objects. Users can type in text strings on the board to make annotations of other objects or just for displaying the text messages. Users can also select and drag the text strings from other applications (e.g. word processors, web browsers, email clients, etc.) onto the board for posting the information. Similarly, image objects and moving picture objects can be dragged onto the board from the file browsers or other image browser/moving picture applications. Users can freely move and resize the visual objects on the board.

The audio objects include the sound and voice objects. Voice objects record the users' voice and generate an audio file that can be used for annotating other objects or for posting voice messages. Sound objects are used for posting built in audio files on the board. Users can play the voice and sound objects without any audio player applications installed on their machines.

The File/URL category is used for transferring files or accessing the URLs through *MBB*. This category contains web objects, office document objects, and miscellaneous file objects. When users drag a URL from a web browser onto the board, the *MBB* client automatically generates the thumbnail of the web page and places it on the board as a web object. This thumbnail approach gives users more visible information about the posted URL [12]. The URL and title of a web page are shown as a tooltip text. The web object is handled just like an image object, however the designated action on this object is to launch a web browser and load the URL specified in it. The *MBB* can also be used as a storage archive system with the miscellaneous file objects. Users can upload any type of files by dragging them from a file browser onto the board.



Visiting San Francisco

Hi Everybody !

My Parents are coming to town this weekend
And I was planning to take them around
San Francisco.

Does anyone has good ideas for good place to go
Sights ?
Restaurants ?
Entertainment ?
Thanks so much... Audrey

Audrey

If your parents are buying. take them to
HAWTHORNE's behind SF MoMA

- Peter

The coldest winter I ever spent wa the summer in San Frncisco
ARK TWAIN

Take them to CHINA TOWN and eat di
"New Asia" corner of pacific and stockt

Good food
Good price !

Galric Rose is also tasty

But impossible to get in
at dinner time
(Make reservations)

attemp to complete restaurant list
estaurants
Price
Chinese - New Asia
(pacific and stockton)
Indian
Italian - Pasta Pomodow in San Francisco,
San Jose - excellent food
very cheap casual

Don't forget ALCATRAZ,
Cablecars AND WALK on the
GOLDEN GATE BRIDGE !

Great for visitors especially older folks.
But get there early.
The parking lot fills up fast.

Muin Woods

Big Trees

Hmmm....

this sounds interesting ...
I am going to try it sometime this weekend

How expensive ?

Great view point

Sausalito is
but very touristy (which I dislike)
But it's well worth a visit

If they like Galric
I highly recommand
Stinking Rose

The striking rose
Main office
420 Columbia Ave.
San Francisco CA 94133
United States of America
1(415)781-ROSE x721
1(415)392-0744 Fax

Audrey,

If your parents enjoy antiques, It's worth a trip
to San Anealina in marina...
It's also not too far from Sausalito !

San Francisco CITY GUIDES
FREE WALKING TOURS

Here is an official website
of the city and county of
San Francisco !

Figure 2. Multimedia bulletin board message on the topic of "Visiting San Francisco"

In addition, users can download a file from the board by double clicking on a miscellaneous file object. The office document object is treated a little bit differently from the miscellaneous file object. This is because users are using more and more office documents in everyday life and the visual displays of the office documents help users identify them more easily. Like a web object, the *MBB* client automatically generates the thumbnail image of the first page of the office document when users drag an office document onto the board.

Simple drag-and-drop interaction is used as a basic metaphor for composing messages such as add, delete, move, resize, link, and so on. This was intended to minimize users' burden of managing and posting the objects on the board. In addition, the *MBB* was designed to perform a proper action based on the object type such as "launch the associated applications" or "download the multimedia files" for the users' double-click interaction on the multimedia objects. Like other asynchronous communication systems, all the changes occurred on the board are updated and sent to the server only after users press the submit button. When the button is pressed, a new transparency layer is added on top of the current board.

Since a bulletin board only shows its final (current) state, it is difficult for users to identify the previous state of the board or to observe the evolving sequence of the board. This information may be useful for recognizing the user dynamics on the specific story thread or the discussion topic. The history keeping mechanism implemented in *MBB* enables users to go back to any previous state of the board. It also makes it possible for users to play back a specific range of the board states.

It is not unusual that several different story threads coexist within a board. If the board is crowded with other messages, it is difficult for users to follow the messages along the specific story thread or to recognize only the messages they are interested in. The *MBB* supports several filtering functions to resolve the problems. From the observation on the usage of an actual bulletin board we found two interesting facts, which influenced the design of the filtering functions. The first fact is that users have a tendency to make an explicit link (line or arrow) between the objects to make their messages belong to the specific story thread. The second one is that users tend to keep their messages as close as possible to the related message in case they don't make an explicit link. Since users depend on the spatial locality among the related messages, we define it as an implicit link. Users can filter out the objects on the board by object type, author, layer, and creation time. Users can also filter out the objects by specifying the depth of explicit or implicit link path from the selected object. The filtered sets of the messages can be combined conjunctively and the history keeping mechanism can be applied to the filtered objects as well.

Because our initial *MBB* prototype was designed to have a fixed size display (1024*768 pixels) without any navigation mechanism, users need some way to expand board space without losing the context of the existing board. To resolve this problem, we allow a hierarchical relationship among the boards. If there is not enough space to add a new multimedia object on the board, users can shrink the current board into a new image object and add it to a new board. The newly created board is inserted as a child of the current board and users can easily go back to the parent board by double clicking on the board image object. Users can preview the children of the current board as thumbnails in the *MBB* control bar, and they can also browse the whole hierarchy of the boards with the treeview list. The operations of creating a child board and going back to the parent board are performed with animation to minimize users' confusion and frustration that may come from jumping to another board in the hierarchy.

These novel features in the *MBB* prototype enable users to have richer communication and collaboration, but there still remain many challenges. In this paper, we focus on two of them:

(1) Most features mentioned above were designed and implemented mainly for helping users browse, filter, search, and follow the threads of messages on the board as they were initially created. However, in many cases, users need to perform more sophisticated tasks that cannot be easily achieved by keeping the existing structure of the messages.

(2) The partial or entire occlusion of messages often prevents users from identifying and recognizing the information on the board. Therefore, users need a way to manipulate the message content and display it in a clear way.

4. Dynamic Layout Template

To cope with these challenges we developed the concept of dynamic layout template (*DLT*) to dynamically restructure the layout of the multimedia messages on the board.

In *MBB*, all the multimedia objects and their relations on the board are represented as a graph structure. Each of the multimedia objects represents a node and a relation between the multimedia objects represents a link. The flow algorithm defined in the *MBB* projects the internal data representation (graph) onto the 2D space (board). *DLT* enables users to restructure any part of a board layout dynamically by applying a new flow algorithm to the designated part of the internal data representation.

Figure 3(a) shows the sample personal photo collection board created by three users. Image objects with captions (text objects) on the board are clustered into a few groups according to the related events by end-users. Figure 3(b) shows the transformed board layout after two *DLTs* (fence-grid and circular *DLTs*) have been applied to each type (image and text) of the multimedia objects on the board. Fence grid *DLT* is the layout template that places the grid cells along the four sides (top, right, bottom, and left) of the window. Users can specify the number of the horizontal and vertical cells in it. Circular *DLT* places the objects along the circumference of the circles. Users can select the number of concentric circles (they can be spaced evenly or relatively to the radius of circles) and the number of points where the objects will be placed. In Figure 3(b), image objects are sorted by creation time and arranged clockwise around the board with same size. Similarly, text objects have been arranged along the circumference of the circular *DLT* that is placed inside the fence-grid *DLT*. With this newly formed layout, users can see the image objects more clearly in chronological order without any occlusions. Users can assign a different size to each of the multimedia objects according to any of the attributes stored in the *MBB* database. For example, users can resize each of the selected image objects relative to the initial image size or in chronological order through the *DLT*. Figure 4(a) shows the sample web bookmarks board created by a single user. The user dragged some of the interesting URLs from the history of a web browser onto the board to save them as his favorites or bookmarks. The automatically generated web objects are placed on the board (2D space not 3D) according to the user's grouping mechanism just as in Microsoft's Data Mountain [7]. Figure 4(b) shows the newly generated layout of the board by applying two calendar *DLTs* (Jan 2002 and Feb 2002). All the web objects are arranged inside the calendar *DLTs*. If more than one web object is placed within a grid cell, users can choose either a tile (no occlusion but different size) or cascade (allow occlusions but equal size) arrangement. In this example, the upper left calendar *DLT* uses the cascade arrangement and lower right *DLT* uses the tile arrangement. It also shows a good example of applying the different ontologies to the same data set through *DLT*.

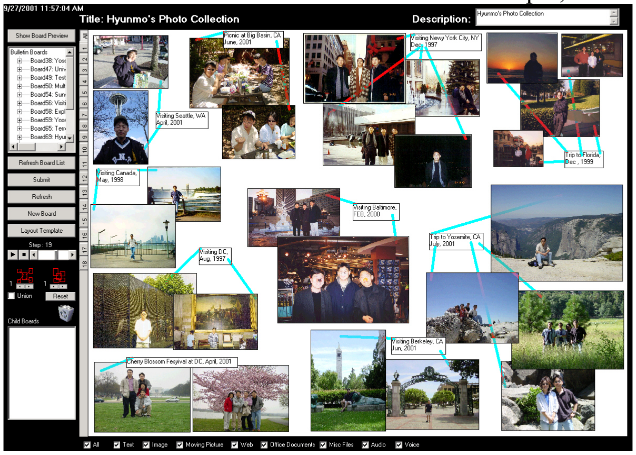

9/27/2001 11:57:04 AM
Title: Hyunmo's Photo Collection
Description:
Hyunmo's Photo Collection
Show Board Preview
Refresh Board List
Submit
Refresh
New Board
Layout Template
Union
Reset
Child Boards
Picnic at Big Basin, CA June, 2001
Visiting New York City, NY Dec, 1997
Visiting Seattle, WA April, 2001
Trip to Florida Dec, 1999
Visiting Canada, May, 1998
Visiting Baltimore, FEB, 2000
Trip to Yosemite, CA July, 2001
Visiting DC, Aug, 1997
Visiting Berkeley, CA Jun, 2001
Cherry Blossom Fesyival at DC, April, 2001
All
Text
Image
Moving Picture
Web
Office Documents
Misc Files
Audio
Voice

Figure 3. (a) Photo collection board with the original layout created by three users

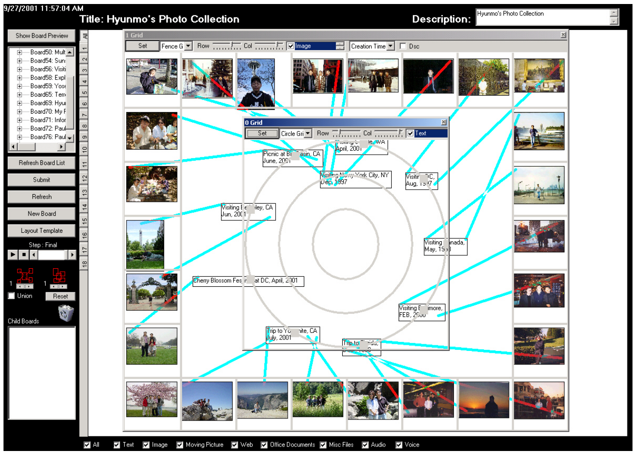


Figure 3. (b) Image objects are sorted and arranged by fence-grid *DLT,* and text objects by circular *DLT*.

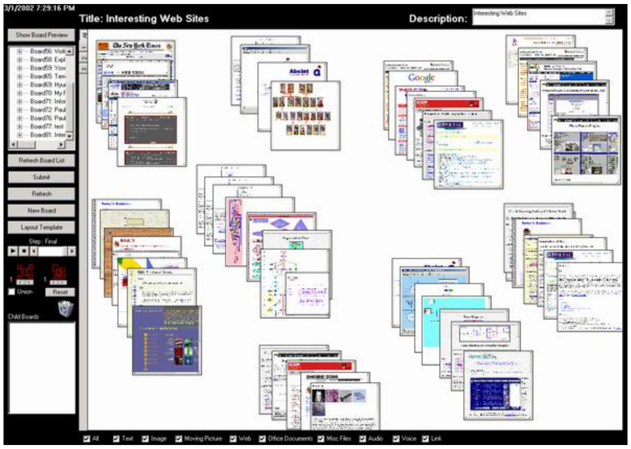


Figure 4. (a) Personal web bookmarks with the original layout created by a single user. Similar to Data Mountain arrangement with occlusions.

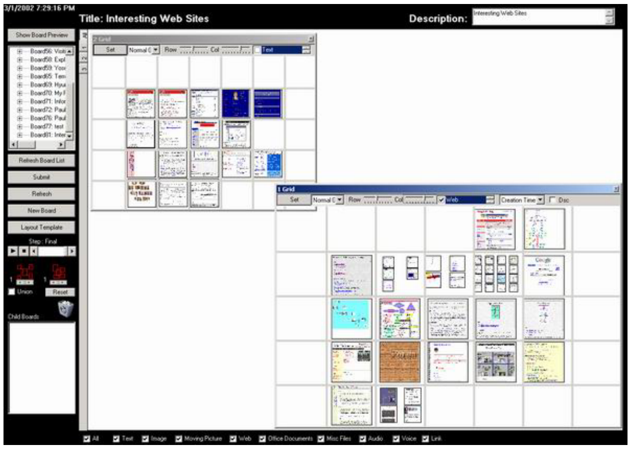
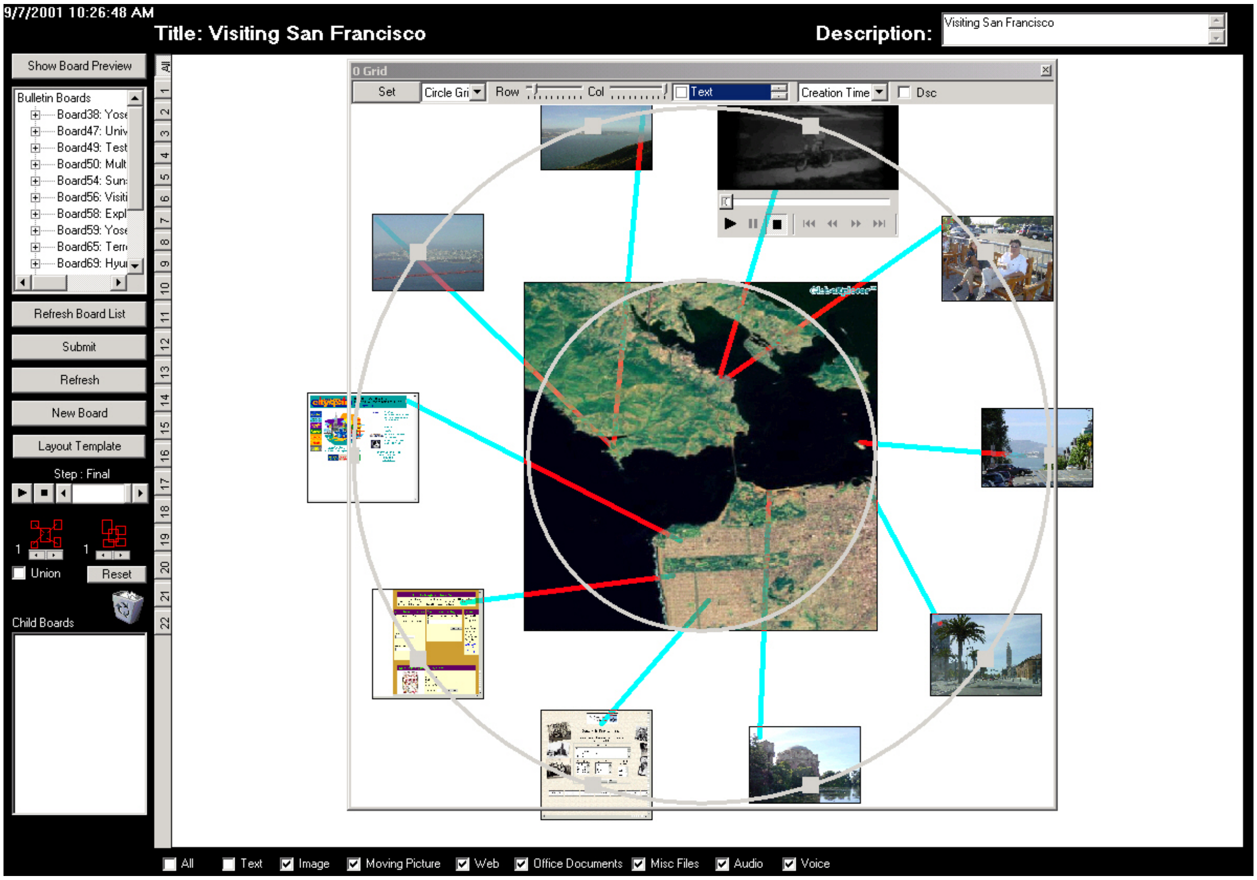


Figure 4. (b) Web objects are sorted and rearranged by two different size calendar *DLTs.* Bottom right *DLT* uses the tile arrangement to avoid occlusions.



9/7/2001 10:26:48 AM

Title: Visiting San Francisco

Description: Visiting San Francisco

Figure 5. Discussion board on the topic of "Visiting San Francisco" revisited. From the initial layout (Figure 2), text objects are filtered out, and the rest of the objects are arranged by Spring model (a model to minimize the crossing of explicit links and overlapping) *DLT* with the center of a map image object.

Figure 5 shows the transformed layout of the board from the sample discussion board on the topic of "Visiting San Francisco" (Figure 2) created by multiple users. In Figure 5, the text objects were filtered out from the board and the spring model *DLT* was applied to the rest of the objects. The spring model *DLT* is a layout template that spreads out the objects along the circle in a way that minimizes the crossing of explicit links and the overlapping among the objects. Figure 5 shows that an image object (map) was chosen to be the center of the *DLT* and the other objects were arranged around it without any crossing of the explicit links and with no overlapping.

MBB users can choose any number of *DLT*s simultaneously and apply them to the designated subset of the messages on the board. Users can select the subset of the message objects by object types, authors, time period, etc. and they can also select the objects by specifying a rectangular region on the board. *DLT*s were designed and implemented as resizable see-through windows to maximize the visibility of the objects and the flexibility of object management. In order to minimize users' confusion and frustration that may come from the sudden layout change, all the layout transformations performed by *DLT*s are animated. All the layout transformations are reversible and recorded in the history on the client machine.

In our initial prototype, we implemented five sample *DLTs*: grid (2D row by column), fence-grid (2D row by column but using only outside rows and columns), calendar (shows a month organized by weeks), circular (objects are evenly spaced in a circle that fits the board), and spring model (circular with a central image and no overlapping links). Our current research is identifying other *DLTs* that are useful for *MBB* systems through user studies. Future research will include the development of a *DLT* toolkit that enables users to define their own *DLTs*.

5. Design and Implementation of MBB

The *MBB* was built on a client/server architecture and the *MBB* client was implemented as an ActiveX component with Microsoft Visual Basic 6.0. The MBB client ActiveX control is placed on a web page using the OBJECT tag as shown below. The CLASSID field represents the unique control identifier (CLSID) that is created when the control is compiled.

```
<HTML>
<HEAD>
<TITLE>Multimedia Bulletin Board</TITLE>
</HEAD>
<BODY>
<OBJECT ID="MBB"
  CLASSID="CLSID:F7DA6D98-EA19-43CD-BB9B-
  448BB437CB34"
  CODEBASE="MBB.CAB#version=1,0,0,46">
</OBJECT>
</BODY>
</HTML>
```

When users load the above web page with a web browser, the MBB client ActiveX control is automatically activated and connected to the web server, FTP server, and database server that are running on the MBB server machine. Internet Information Service (IIS 5.0) web server is used for Hyper Text Transfer Protocol (HTTP). The FTP server is used for transferring the multimedia object files between the server and the client, and a Microsoft SQL server 2000 database server is used for retrieving and storing the information.

The MBB operates using a database (Microsoft MS SQL server 2000), which contains seven linked tables (Figure 6). The basic concept is that the MBB is composed of thousands of boards, each board contains dozens of layers, and each layer contains several multimedia objects created by users. A link can be built between any two objects regardless of the layers they belong to as long as they are contained in the same board.

In the MBB database schema, the Board table represents the collections of layers with attributes such as board title, board description, creation time, last update time, creator, and the path where the multimedia object files are stored on the server side. The Layer table represents the collection of the multimedia objects with attributes such as boardid (to represent which board the layer is contained in), layernumber (the order of the layer within a board), creation time, update time, and creator. A 1-to-many relationship between the Board table and the Layer table has been set so that a board can contain multiple layers but not vice versa. Similarly, 1-to-many relationship has been set between the Layer table and the Object table so that a layer can contain multiple objects but the same object is not allowed to be contained in more than one layer in this prototype. The Object table represents the multimedia objects in a board. It stores all the information about the object that appears on the board. In our initial prototype, attributes include the type of multimedia objects, position, size, and path as well as creationtime and updatetime. The deleted flag is also defined in the table to indicate whether this object has been deleted from the board just for history keeping. The Link table is used to specify the implicit or explicit link established between the multimedia objects. The attributes include creator, thickness of the link, and deleted flag too.

In addition to the above four tables, the MBB contains three more tables: BoardHierarchy, History, and User. The User table specifies which board is created by whom. The History table keeps the records of the modifications in the board such as add, delete, move, resize, and so on. Since all the modifications are time-stamped and recorded in the History table, MBB users can restore and regenerate any previous state of the board. Users can also play back a specific range of board states to see the evolving sequence of the board. The BoardHierarchy table is used to represent the hierarchical relationships among the boards. Since the board has a fixed size (1024*768 pixels) in our prototype, the board expands its space by creating child boards. The information stored in the BoardHierarchy table help users navigate the board hierarchies in MBB.

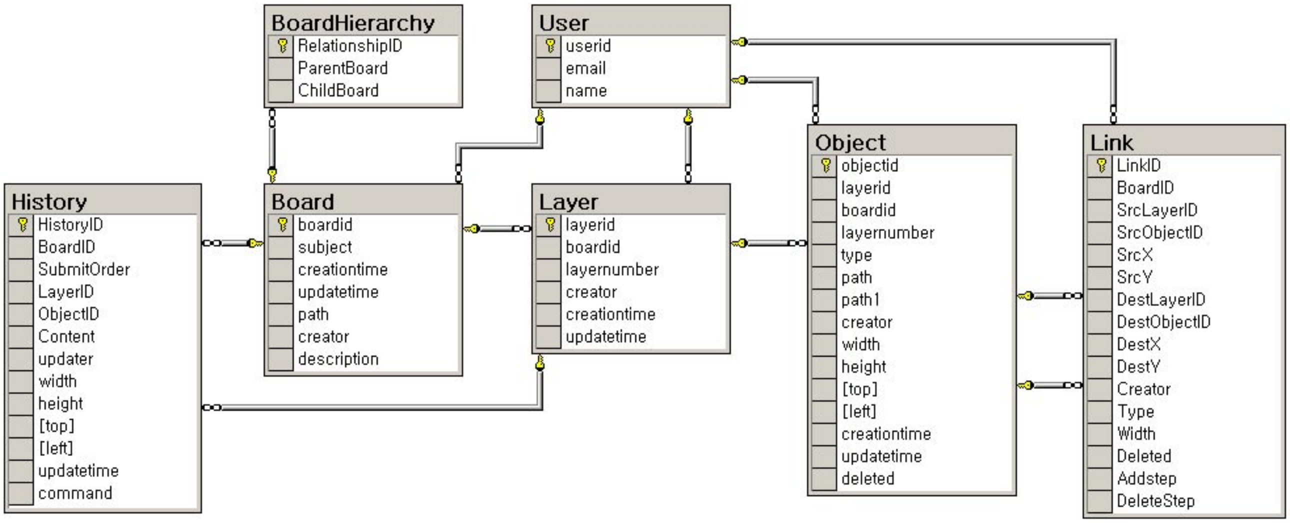


Figure 6. The schema of Multimedia Bulletin Board database

6. Conclusions and Future Directions

The *Dynamic Layout Templates* developed in the *Multimedia Bulletin Board* enable users to manage and reorganize large numbers of multimedia objects. The *DLT* makes it possible for users to transform the layout of the multimedia messages in a way that would be most appropriate for their sophisticated tasks. In addition, the *DLT* enables users to see the content of the multimedia objects in meaningful contexts, such as calendars. *DLTs* can be used to construct new information through the 2D spatial layouts, and also to apply appropriate ontologies.

Many new research directions seem apparent:

(1) user studies would evaluate the effectiveness of the current prototype and suggest design improvements

(2) formal metrics to measure the effectiveness of the system would guide future work.

(3) refinements to the automatic layout mechanism based on the characteristics of the multimedia messages would automate many tasks.

Future research also includes generalizing the concept of *DLT* so that it can be applied to other application areas such as desktop, document, and window management systems.

Acknowledgement: We appreciate the support of Ricoh Innovations, Inc., and the contributions of other members of the Human-Computer Interaction Laboratory at the University of Maryland.

7. References

[1] Kuchinsky, A., Pering, C., Creech, M. L., Freeze, D., Serra, B., and Gwizdka, J., "FotoFile: A Consumer Multimedia Organization and Retrieval System", *Proc. ACM CHI99 Conference on Human Factors in Computing Systems*, 496-503, May, 1999.

[2] Beaudouin-Lafon, M., "Novel Interaction Techniques for Overlapping Windows", *Proc. UIST 2001,* 153-154, 2001.

[3] Bell, B., Feiner, S., "Dynamic space management for user interfaces", *Proc. UIST 2000,* CHI Letters 2(2):239-248, 2000.

[4] Good, L., Bederson, B., "CounterPoint: Creating Jazzy Interactive Presentations", *Technical Report HCIL-2001-03, CS-TR-4225, UMIACS-TR-2001-14*, March 2001.

[5] Kandogan, E., Shneiderman, B., "Elastic Windows: Evaluation of Multi-Window Operations", *Proc. ACM CHI'97*, 250-257, 1997.

[6] Kullberg, R.L., "Dynamic Timelines: Visualizing Historical Information in Three Dimensions", *M.S. Thesis at the MIT Media Laboratory*, September 1995.

[7] Robertson, G., Czerwinski, M., Larson, K., Robbins, D. C., Thiel, D., van Dantzich, M., "Data Mountain: Using Spatial Memory for Document Management", *Proc. ACM UIST98*, 153-162, November 1998.

[8] Rose, D.E., Borenstein, J.J., Tiene, K., "MessageWorld: A new Approach to Facilitating Asynchronous Group Communication", *Proc. CIKM'95*, 1995.

[9] Uchihashi, S., Foote, J., Girgensohn, A., and Boreczky, J., "Video Manga: generating semantically meaningful video summaries", *Proc.7th ACM International Conference on Multimedia,* 1999.

[10] Shneiderman, B., Kang, H., "Direct Annotation: A Drag-and-Drop Strategy for Labeling Photos", *Proceedings of International Conference Information Visualisation (IV2000)*, 88-95, July 2000.

[11] Siviter D., Petre, M., Klein, B., "Harnessing technology for effective inter- and intra-institutional collaboration", *Report of the ITiCSE'97 Working Group on Supporting Inter- and Institutional Collaboration*, 1997.

[12] Woodruff, A., Faulring, A., Rosenholtz, R., Morrison, J., and Pirolli, P., "Using Thumbnails to Search the Web", *Proc. CHI 2001,* 198–205, April 2001.

[13] http://www.acdsystems.com/

[14] http://www.mgisoft.com/

[15] http://www.ofoto.com/Welcome.jsp