'98 International Conference Digital Library & Knowledge
September 22, 1998, Seoul Korea

Presentation Transformation Scheme for Effective Multimedia Object Browsing

Jaehyuk Cha
Department of Educational Computing, Hanyang University
17, Haengdang-Dong, Seongdong-Gu, Seoul, 133-791, Korea
e-mail: jaehcha@email.hanyang.snu.ac.kr

Abstract

Users want to browse various groups of nested attribute values of an object. On the other hand, in case of the presentation of a multimedia object, the form-based presentation is superior to the graph-based presentation. Therefore we propose a form-based presentation transformation scheme that allows users to reorganize the presentation layout to fit the limited screen and to show the values of all the needed attributes. For the representation of the presentation scenario of an object a presentation information class and the presentation transformation operations are defined. We show how these operations transform the default presentation into the wanted presentation by navigating through a multimedia object with the COMIB (COMposite Icon Browser).

1 Introduction

A multimedia object is a composite object that refers to various kinds of media objects such as image, graphic, audio, and video [13]. Because of the limited screen size, any presentation layout such as the form and the table cannot fully show several multimedia object and their many referred objects altogether on the screen [8]. In order to see several nested attribute values of each object, users must repetitiously navigate from each object to its referred objects along the paths of those attributes.

For example, the form-based presentation cannot deal with several objects altogether and the table-based presentation cannot effectively represent the complex structure and the mono-media objects such as image and video because these media need a 2-dimensional space. And an icon-based presentation can show several multimedia object and their many referred objects [10, 4, 5]. An icon is the thumbnail of an object, for example, a miniature of the first page of an object or a smaller image instead of the full resolution image. However, it might not convey the meaningful information such as the complex structure

*This paper was supported in part by the 1998 Campus Research Fund of Hanyang Univ., Korea

of the object to a user. And Fisheye view [6, 11], Tree-Maps [12], Perspective Wall [7], and Cone Trees [9] are the presentation techniques for graphs. But, all of them cannot support the spatio-temporal presentation.

Thus existing presentation techniques cannot fully satisfy the user's request of browsing a large number of nested attribute values of each object altogether. Therefore we propose the form-based presentation transformation scheme that allows users to dynamically reorganize the presentation layout to show the values of all the needed attributes and fit the limited screen size

The rest of this paper is organized as follows. Section 2 explains the extension of the object-oriented data model and introduces the concept of a composite icon. Section 3 describes the object presentation layout of the COMIB (COMposite Icon Browser) [3] by an informal presentation of the entire retrieval session. Section 4 describes a presentation information object, that is the representation of the presentation scenario of a multimedia object used by the COMIB. Section 5 defines the presentation transformation operations on the presentation information, and shows how these operations transform the default presentation into the wanted presentation. Finally, we present conclusions in Section 6.

2 Preliminaries

2.1 Data Model and Its Extension

An object-oriented data model is an elegant basis for addressing all multimedia data modeling requirements such as type extensibility for new media and the support of methods for the intrinsic operations of mono-media [13]. This model is based on a number of basic concepts, namely object, identity, class, aggregation, inheritance [2].

Figure 1 illustrates the example classes of a university database schema that will be used in the remainder of the paper. An attribute of any class but the root class on an aggregation hierarchy is logically the attribute of the root class of the hierarchy, that is, the attribute is a **nested attribute** of the root class. In general, a nested attribute is denoted as a path expression, $a_1.a_2.\cdots.a_n$, that is a sequence of non-nested attribute names along the aggregation hierarchy of a class and $a_i(1 \leq i < n)$ should be the name of a tuple-valued attribute or a multi-valued attribute. For example, when the aggregation hierarchy of class *Student* contains classes *Department*, *Career*, *Professor*, and *Course*, the inherited attribute, 'picture' of *Professor* becomes the nested attribute of *Student*, denoted as 'advisor.picture'. And if the domain of a nested or non-nested attribute is one of primitive classes or mono-media classes, the attribute is a **presentable attribute**.

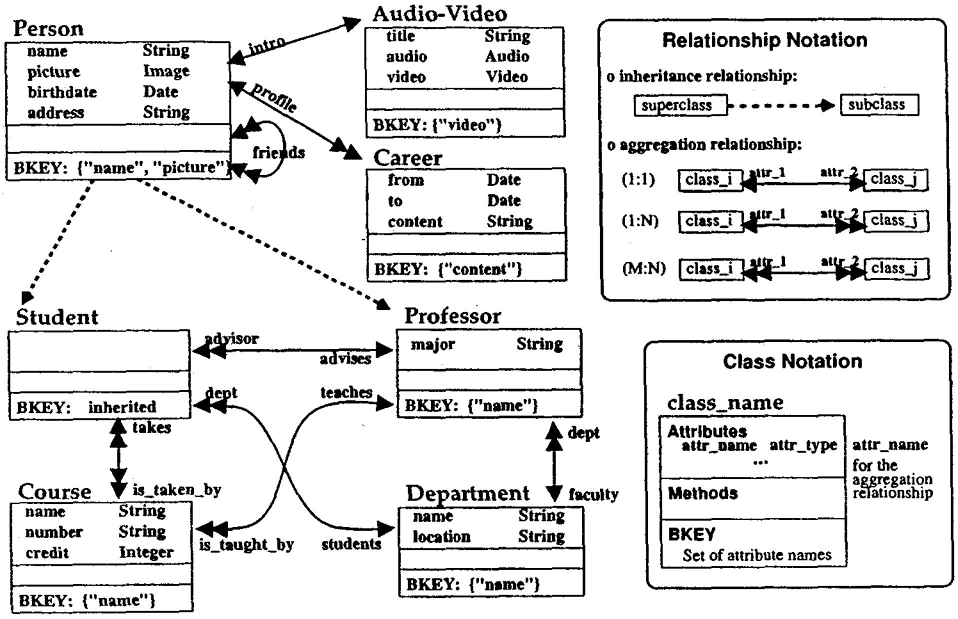


Figure 1: University Database Schema

In defining a new class, it must be explicitly specified whether each attribute has BKEY (Browsing KEY) characteristic or not. The BKEY attributes of a class are used for displaying initial composite icons. A class without any defined BKEY attribute inherits this characteristics from its superclass(es). Thus, in case of class *Student*, inherited attributes 'name' and 'picture' will have BKEY characteristic.

2.2 A Composite Icon

In general, an icon typically presents a symbol of the domain type, some non-nested attribute values, or the first page of an object. With these icons, various kinds of aspects of multimedia objects, specially complex structure, can't be effectively represented. For this reason, we extend a icon as follows. A composite icon is a miniaturized presentation of data projected from an object according to a set of presentable attributes. For associating composite icons to objects of a class, a mapping information based on the class is needed. This information is defined as follows. A composite icon path is a set of presentable attributes used for making up a composite icon from an object.

According to the above definitions, a nested attribute can also an element of a composite icon path, and a composite icon can represent various kinds of aspects of a multimedia object better. For instance, Figure 2 illustrates the relationship between the retrieved objects and the composite icons of class *Student* when the composite icon path consists of the attributes 'name', 'picture', and 'advisor.picture'.

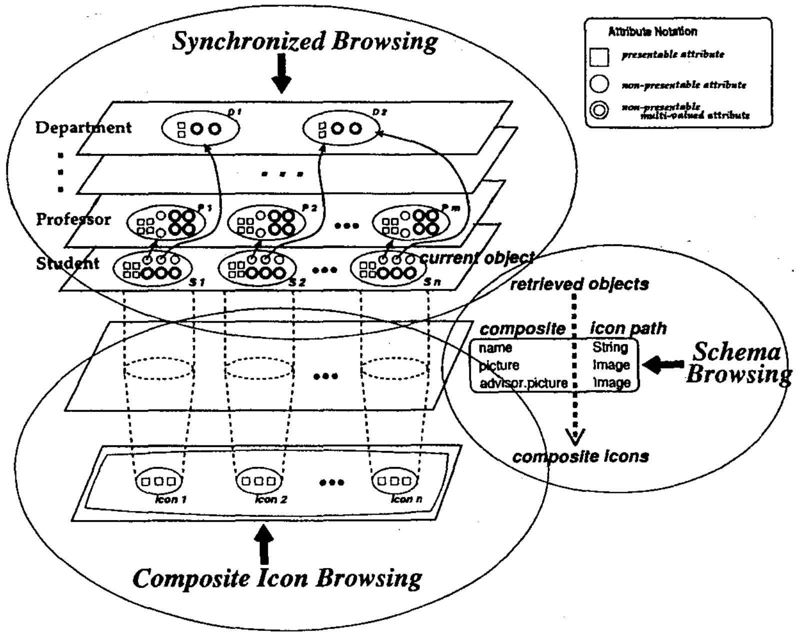


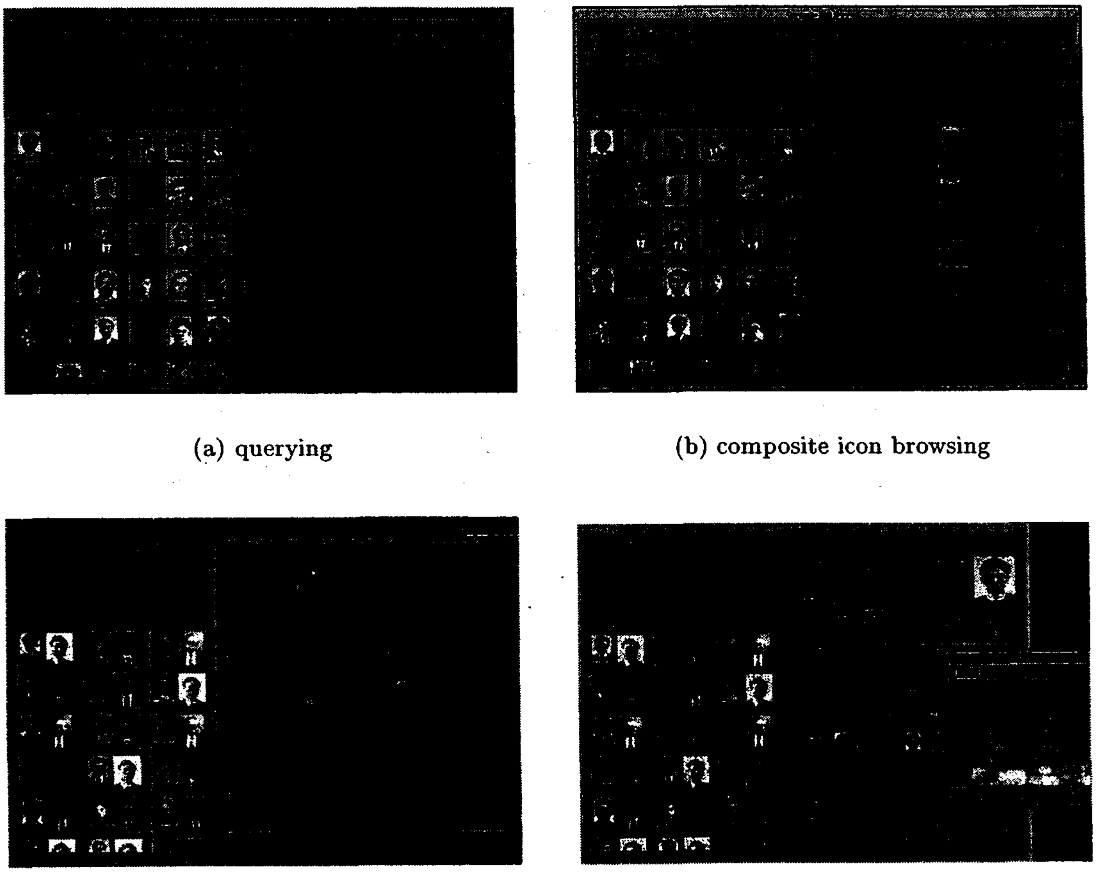
Figure 2: Browsing Techniques and Object of COMIB

For a class T, the default composite icon path comprises its BKEY attributes. For instance, as shown in Figure 1, the default composite icon path of class *Student* consists of two inherited BKEY attributes, 'name' and 'picture'. These attributes are used for displaying initial composite icons of *Student* objects.

3 Object Presentation Layout of COMIB

We will show the presentation layout of a multimedia object by performing the four retrieval techniques of COMIB [3] in an example retrieval session. Figure 3(b) shows the screen layout of COMIB that consists of three tiled subwindows and one pop-up subwindow, each of which supports one of the above four techniques respectively.

The top left subwindow is a query window that allows users to specify a query statement, and displays the status information and error messages in the message area. The bottom left subwindow is an icon window that displays the composite icons of the query result and allows users to select one of those icons. Then the corresponding object is presented fully in the right subwindow called an object window. The pop-up subwindow is a composite icon path window that visualizes a current composite icon path and an



(a) querying

(b) composite icon browsing

(c) composite icon browsing & schema browsing

(d) synchronized browsing

Figure 3: An Example Retrieving Student Objects

aggregation hierarchy, and allows users to directly specify a new composite icon path on the visualized aggregation hierarchy.

3.1 Querying

The COMIB interleaves navigation-oriented browsing with set-oriented querying using the query reformulation. In the query window, the user may formulate the query, and get a set of objects. If the query result is not satisfactory, the user can modify the query and resubmit it to the DBMS. For example, in order to search the students belonging to the department of computer engineering, the user may formulate the query as shown in the top left subwindow of Figure 3(a), "Select s From Student s Where s.dept.name='Computer Engineering'".

The COMIB supports two browsing techniques: composite icon browsing and synchronized browsing. These techniques allow users to browse and navigate the multimedia objects simultaneously or one by one respectively.

3.2 Composite Icon Browsing

When the user gets objects as the query result, his request of composite icon browsing invokes the presentation of composite icons, each of which consists of data projected from an object according to the current composite icon path. For example, the bottom left subwindow of Figure 3(b) shows the composite icons associated with the given objects of class *Student* according to the default composite icon path, {'name','picture'}.

In case of there being too many objects to scan, it is a tedious work for the user to repetitiously navigate through objects one by one, along the same path. Since a composite icon path associates composite icons with objects, it is possible for the user to easily inspect various kinds of composite icons from the given objects, by modifying the composite icon path. This gives users the effect of navigating through multimedia objects along the several paths of a class aggregation hierarchy. Consequently, it reduces the user's tedious work of repetitious short-sighted navigation.

For example, while looking at the composite icons of student objects as shown in Figure 3(b), suppose a user wants to navigate through the advisors' pictures of all the given students. Then, by adding the nested attribute, 'advisor.picture' to the current composite icon path, {'name','picture'}, he can see more expressive composite icons including the advisor's picture as shown in the bottom left subwindow of Figure 3(c).

3.3 Schema Browsing

For a novice user who knows little about the aggregation hierarchy, the COMIB helps him to directly manipulate the composite icon path on the visualized aggregation hierarchy of the query result. While browsing and zooming in the hierarchy, the user can toggle the leaf attribute node as highlighted or not. Therefore users can easily understand the conceptual structure of the query result using the iterative refinements approach [1, 14], and can directly specify the composite icon path using a mouse.

In Figure 3(b), the composite icon path window represents the aggregation hierarchy of class Student and the current composite icon path, {'name','picture'}. In order to add the attribute, 'advisor.picture' to the current composite icon path, first the user zooms in the leaf node, i.e. the attribute 'advisor' by additionally visualizing the aggregation hierarchy of the domain class, *Professor.* And he highlights the edge and the node representing the attribute, 'advisor.picture'. Thus the extended aggregation hierarchy with the modified composite icon path, {'name', 'picture', 'advisor.picture'} is shown in the composite icon path window of Figure 3(c).

3.4 Synchronized Browsing

To retrieve a multimedia object fully, the user need only select the corresponding icon. When selecting one composite icon in the bottom left subwindow, the multimedia object corresponding to the selected icon is retrieved. Objects are presented one at a time. For example, the right subwindow of Figure 3(d) presents the currently selected object of the class *Student.* This allows users to retrieve any object directly without scanning next or previous object repetitively.

When presenting a multimedia object, the COMIB can display its referred objects as the embedded form, as the default composite icon for a single-valued attribute, or as the set of default composite icons for a multi-valued attribute. For example, profile attribute values are presented as the table, and intro, friends, advisor, dept, and takes attributes are presented according to the default composite icon path of the domain classes *Audio-Video*, *Person*, *Professor*, *Department*, and *Course* respectively. This enhancement of the expressiveness for the object enables users to understand the query result better.

In addition, any referred object can be fully presented in the new pop-up window by selecting the composite icon corresponding to the object. For example, as shown in Figure 3(d), the object of *Audio-Video* class is presented by selecting the composite icon of intro attribute value.

4 Representation of Presentation Information

The COMIB provides two display modes for multimedia object presentation. In the icon-based display mode, objects are presented as the composite icons. In the form-based display mode, an object is presented in an object form with composite icons. To support the form-based and icon-based presentation of multimedia objects, each class has one or more presentation information objects, which represent the presentation scenario of the class. And each class has a default presentation information object. A multimedia object is presented using its class information and a presentation information object. These presentation information objects are the instances of *PresInfo* class of Table 1.

Table 1: Presentation Information (PresInfo) Class

Properties	**Semantics and Domain Types**
name	*The name of the Presentation Information* *String*
ciPath	*The composite icon path of the class supported by this object* *Set<String>*
direction	*The direction that places the attribute values in this presentation layout* *enum{ Horizontal, Vertical }*
gap	*The minimal interval between the attribute values in this presentation layout* *Integer*
label	*The flag to show the attribute names* *enum{ Show, Hide }*
content	*The name and the presentation information of all the attributes* *Set< Struct< name: String, pres: UnitPresInfo > >*

The name property represents the class name of a multimedia object and the serial number of this presentation information object in its corresponding class. If that class is the object class, the form-based presentation is used. If that class is the set class, the list-based presentation is used, showing all the element objects altogether. And the content property represents the name and the presentation information of all the non-nested attributes of the corresponding class. The presentation information of each non-nested attribute of this class is represented as the instance of the *UnitPresInfo* class of Table 2.

Table 2: UnitPresInfo Class

Properties	**Semantics and Domain Types**
presInfo	*The name of the presentation information object* *for the corresponding attribute* *String*
flag	*The flag to show the corresponding attribute value* *enum{ Show, Hide }*
type	*The presentation layout of this attribute values* *in the presentation of an object* *enum{ Embedded, Icon }*
labelPos	*The location of the attribute name relative to the attribute value* *enum{ None, Left, Right, Top, Bottom }*
area	*The spatial information of this attribute* *Struct< width: Integer, height: Integer >*

A unit of presentation is a window that displays the owned attribute values of an object and its referred objects as the embedded form or as composite icons. When the object of a class *A* refers to the object of the class *B* via an attribute, the presentation information object of the class A also has link to the presentation information object of the class B. If the type property for this attribute is "Embedded", the presentation of the object displays the referred objects as the nested windows respectively. If the type property for this attribute is "Icon", the presentation of the object displays the referred objects as the composite icons respectively. And after user's selecting this icon, the referred object is displayed in the independent pop-up window.

According to the aggregation hierarchy of class *Student*, the user can browse and navigate through *Student* database which contains a large number of objects of various classes such as *Student, Audio-Video, Career, Person, Professor, Department, Course.* To present the above objects, the set of default presentation information objects in Figure 4 are used.

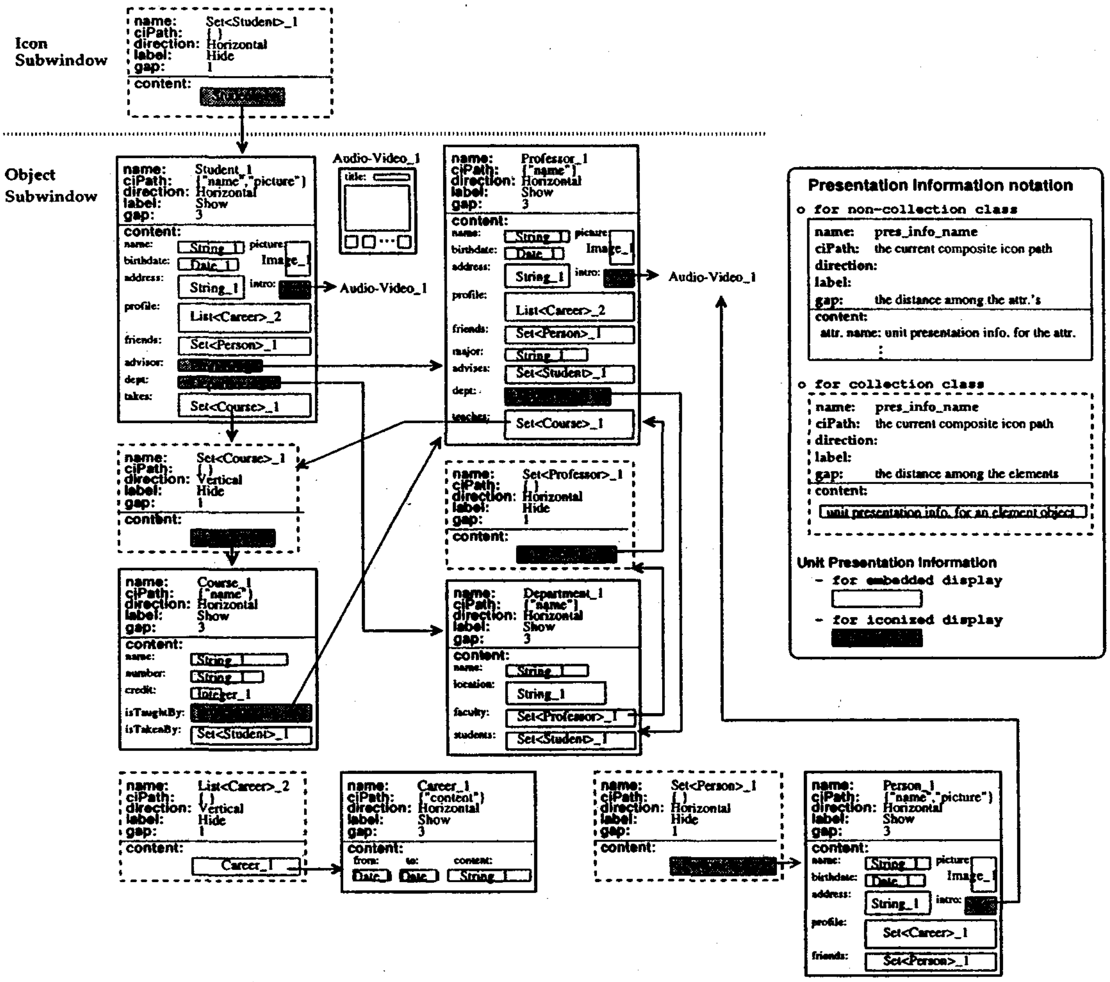


Figure 4: Default Presentation Information Objects for University Schema

Figure 4 shows the default presentation information object of *Student* class of which one object is shown in Figure 3(d). This presentation information object has various presentation informations for the attributes of the class *Student*. For example, the presentation informations of name, picture, birthdate, address attributes denote the system-supported presentation of the data according to their data type. And the presentation informations of intro, advisor, dept attributes denote the minimized presentation of the data via composite icons. And the presentation informations of friends, takes attributes denote the minimized presentation of the element data via composite icons according to the presentation information objects *t1, t2*, respectively. And the presentation information of profile attribute denotes the table-based presentation of data according to the 2-level embedded presentation information objects.

For users to browse the information of the course related to the student a number of windows of Figure 5 are needed as well as the default presentation information objects of Figure 4.

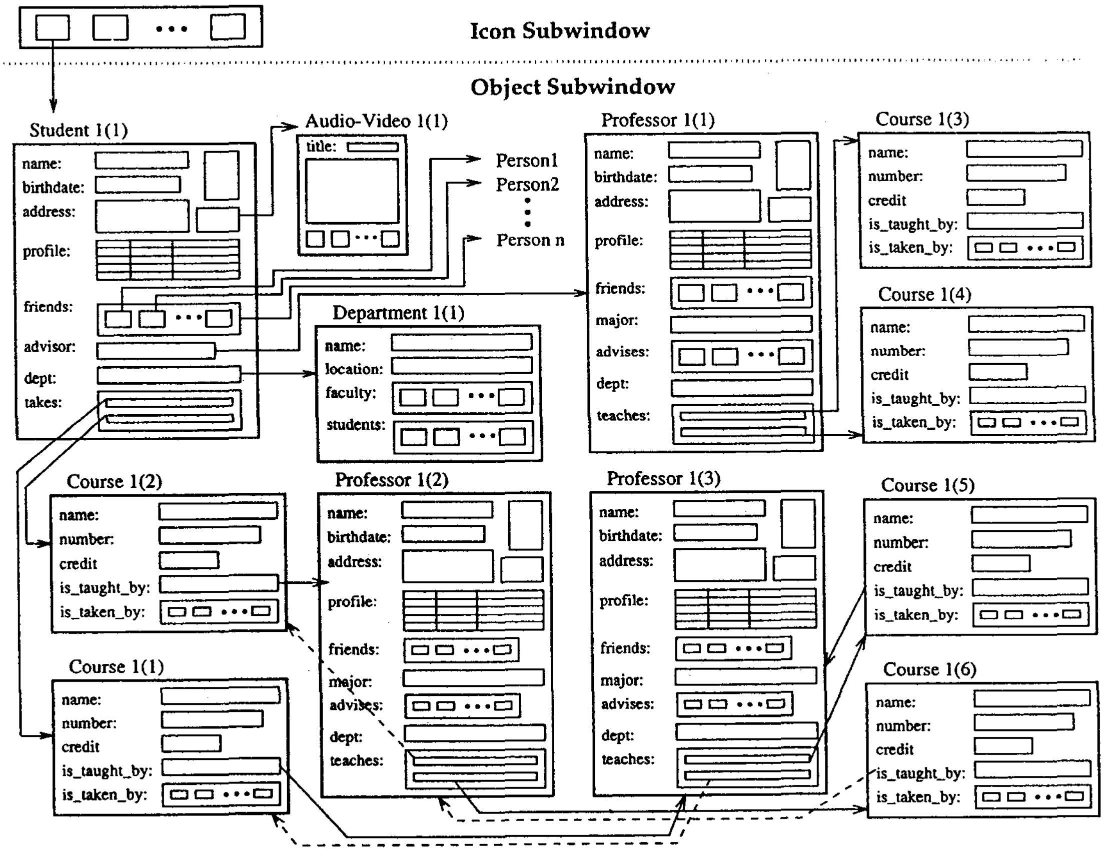


Figure 5: Browsing a Student object according to the default presentation information objects

But, because of the limited screen size, these windows cannot be shown simultaneously. That is, when navigating along the various long navigation paths, we cannot effectively perform navigation such as synchronized browsing. Therefore, in the following Sections, we propose the form-based presentation transformation scheme that allows users to reorganize the presentation layout to fit the limited screen size and to show the values of all the needed attributes.

5 Presentation Transformation Operations

In our presentation transformation scheme, the user can setup the synchronized navigation path and transform the default presentation layout into the wanted presentation layout, during browsing an example object. As a result, a new set of presentation information objects may be generated from the set of default presentation presentation information objects. And the user can perform synchronized browsing efficiently and effectively by applying the above presentation information objects to all the retrieved objects.

For example, in order to focus on the course information related to the student, the user may transform a set of default presentation information objects into a set of presentation information objects shown in Figure 6.

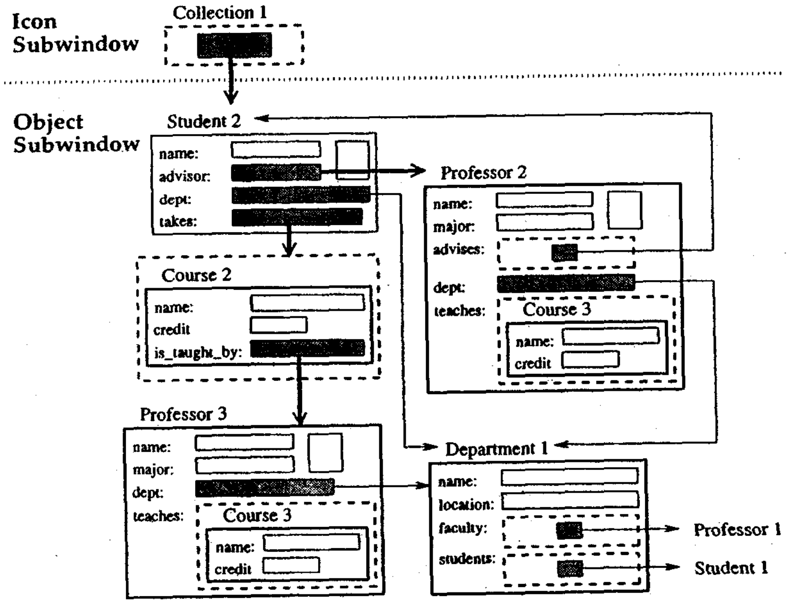


Figure 6: Transformed presentation information objects

And Figure 7 shows the screen layout of user's browsing a student object with the above presentation information objects.

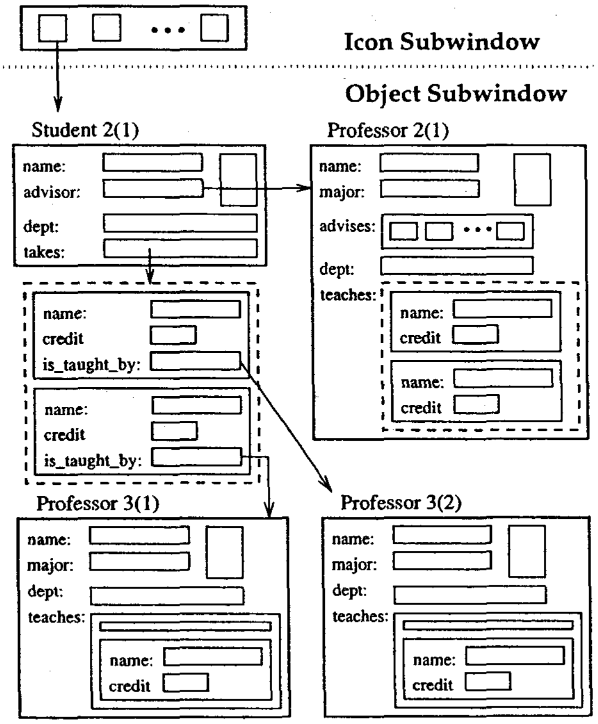


Figure 7: Browsing a Student object according to the transformed presentation information objects

5.1 Form separation/merging operations

Assuming that the domain of an attribute of a class A be the set class B, the presentation information objects of a class A initially contains the presentation information objects of a set class B. When the object of set class has a large number of element objects, it is better to present the object as a separated pop-up window rather than as an embedded subwindow. Thus, as shown in Figure 8, the *detach* operation provides the effect of extracting the nested subwindow from the source window by setting the type property for the corresponding attribute from 'Embedded' to 'Icon'. And the *attach* operation is the inverse of the *detach* operation. Thus the *attach* operation and the *detach* operation support zooming-in and zooming-out the selected portions of an object.

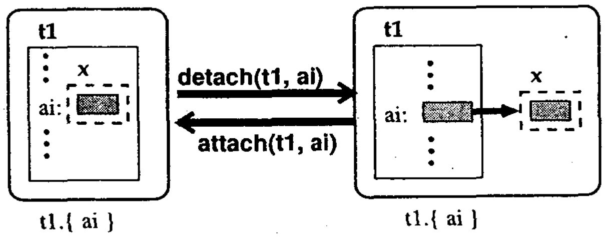


Figure 8: The example of Form operations

5.2 Attribute showing/hiding operations

Browsing multimedia objects of a class, the user focuses some attributes rather than all the attributes. Thus the user can trim off irrelevant attributes from the object presentation. The system supports two modes for doing this. As shown in Figure 9, the user can either specify the relevant attributes using *map* operation, or specify the irrelevant attributes using *hide* operation. And the *show* operation is the inverse of the *hide* operation.

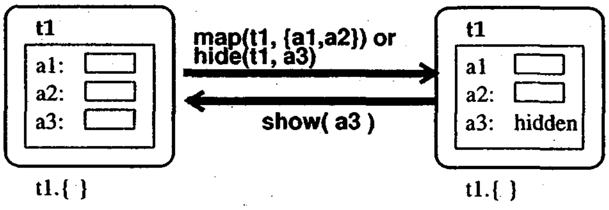


Figure 9: The example of attribute operations

6 Conclusion

This paper describes the form-based presentation transformation scheme. It allows users to reorganize the presentation layout to fit the limited screen size and to show the values of all the needed attributes. First, the presentation information object is defined for representing the schema-level presentation information of a multimedia object. are described using the example retrieval session of the COMIB. Second, the two presentation transformation operations on the presentation information object are defined. And the example-based presentation transformation mechanism is described, which transforms the default presentation information objects into the wanted presentation information objects using the above operations by navigating through an example multimedia object. We showed the proposed scheme can support the user's request of browsing a large number of nested attribute values of each object altogether.

References

[1] Carlo Batini, Tiziana Catarci, Maria F. Costabile and Stefano Levialdi. Visual strategies for querying databases. In *1991 IEEE Workshop on Visual Languages*, pages 183–189, Kobe, Japan, October 1991.

[2] Elisa Bertino, Mauro Negri, Giuseppe Pelagatti and Licia Sbattella. Object-oriented query languages: The notion and the issues. *IEEE Trans. on Knowledge and Data Eng.*, Volume 4, Number 3, pages 223–237, June 1992.

[3] Jaehyuk Cha and Sukho Lee. COMIB: Composite Icon Browser for Multimedia Databases. *Multimedia Tools and Applications*, Volume 3, Number 3, pages 203–224, November 1996.

[4] S. Christodoulakis, M. Theodoridou, F. Ho, M. Papa and A. Pathria. The multimedia document presentation, information extraction, and document formation in minos: A model and a system. *ACM TOOIS*, Volume 4, Number 4, pages 345–383, October 1986.

[5] P. Constantopoulos, E. Lutz, P. Savino and Y. Yeorgaroudakis. Multimedia document storage, classification and retrieval: Integrating the multos system. In *ESPRIT Conf. '89*, pages 819–836, 1989.

[6] G.W. Furnas. Generalized fisheye views. In *ACM CHI '86*, pages 16–23, Boston, MA, April 1991.

[7] J.D. Mackinlay, G.G. Robertson and S.K. Card. The perspective wall: Detail and context smoothly integrated. In *ACM CHI '91*, pages 173–179, April 1991.

[8] Emanuel G. Noik. A space of presentation emphasis techniques for visualizing graphs. In *Proc. of the Graphics Interface*, pages 225–234, Banff, Alberta, Canada, May 1994.

[9] G.G. Robertson, J.D. Mackinlay and S.K. Card. Cone trees: Animated 3d visualization of hierarchical information. In *ACM CHI '91*, pages 189–194, April 1991.

[10] Peter Rosengren, Ulf Wingstedt, Peter Kool and Marie Bern. A tools oriented visual interface for multimedia databases. In *Proc. of Int'l Symp. on Next Generation Database Systems and Their Applications*, pages 219–226, Fukuoka, Japan, September 1993.

[11] M. Sarkar and M.H. Brown. Graphical fisheye views of graphs. In *ACM CHI '91*, pages 83–91, Monterey, CA, May 1991.

[12] B. Shneiderman. Tree visualization with tree-maps: A 2-d space filling approach. *ACM Transactions on Graphics*, Volume 11, Number 1, pages 1–39, January. 1992.

[13] Darrell Woelk, Willis Luther and Won Kim. Multimedia Applications and Database Requirements. In *Proc. of IEEE Symp. on Office Automation*, April 1987.

[14] H. K. T. Wong and I. Kuo. GUIDE: Graphical user interface for database exploration. In *Proc. Int'l. Conf. on Very Large Data Bases*, page 22, Mexico City, Mexico, September 1982.