

SOPView+: An Object Browser which Supports Navigating Database by Changing Base Object

Sung-Woo Chang

Hyoung-Joo Kim

Department of Computer Engineering
Seoul National University
Seoul, Korea 151-742
E-mail: {swchang,hjk}@oopsla.snu.ac.kr

Abstract

In this paper, we present the design and implementation of SOPView+, an object browser that supports navigating a large database by changing base object. Base object is an object which is a basis for navigation; forward navigation is provided for the reference paths ahead of the base object and backward navigation for the ones behind it. SOPView+ allows users to change the base object along the reference hierarchy among a number of database objects; this makes it possible for them to explore a large database until they find objects of their interest on the limited screen space, solving the screen real estate problem. SOPView+ also supports extended synchronized browsing, which synchronizes displaying objects along both forward and backward navigational paths.

1 Introduction

In object-oriented databases(OODBs), the unit of storing and retrieving information is an object[1]. The OODB thus is a collection of a number of objects. The object usually contains plenty of reference information; it represents inter-relationship between objects, and plays an important role to express rich semantics. Therefore, in order to get easy and correct understanding of information on object-oriented database, a browsing mechanism is needed which clearly visualizes objects and their relationships with other ones and allows users to navigate freely a large reference hierarchy of numerous objects.

The SOP project at the Seoul National University is developing an object-oriented database management system called SOP(SNU QODBMS Platform) which supports ODMG(Object Database Management Group) specification[2] and its associated tools.

The prototype¹ has developed in 1995 and is on its way to stabilization and commercialization. SOPView[10] is a graphical user interface on SOP OODBMS which supports visual query and object browsing(including visualization structure, synchronized browsing, and intermediate reference set query). SOPView+ provides more advanced browsing facilities than SOPView.

1.1 Motivation

According to Batini *et al.*, “*browsing is essentially a viewing technique, aimed at gaining knowledge about the database, and the main hypothesis is that the user has only a minor knowledge about the database; furthermore he accesses the system without any predefined goal.*”[3] Within the hypothesis, users’ browsing activities can be specialized into the following two phases: (1) *database exploration to locate objects of their interest* and (2) *information retrieval to get information which they want to*. In the first phase, users usually start a browsing session by designating an object as a starting object or posing a query and getting result objects; they then step forward to browse other related objects which are referenced by the former one(s) through relationships. This process proceeds repeatedly until they find ones of their interest. In this phase, users are inclined to wander here and there, repeating short-sighted navigation[3], and could navigate long distance from the starting object before locating ones they want. On the contrary, once they found them, in the second phase, they stop iterative exploration and scrutinize the objects and their neighborhood in detail to get information.

Traditionally, how to best make use of the avail-

¹It consists of object manager, schema manager, OQL query processor, visual query and object browser, multimedia library, and RDB schema gateway, etc.

able screen space has been a fundamental issue in user interface design, commonly referred to as the *screen real estate problem*. During the first phase, the screen space could be wasted when navigating database. For example, in Figure 1, suppose that the user started

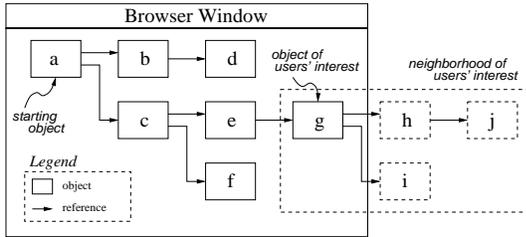


Figure 1: The problem of not displaying an object and its neighborhood of users' interest

a browsing session with an object *a* and found an object *g* of his/her interest. Now, he/she wants to look up information which is contained in the object and its neighborhood, but, it is difficult or inconvenient to browse them because not much screen space is left around the object. The reason is that the objects which has already been navigated and are of no his/her interest remain on the screen, and thus consume the screen space of limited size. Therefore, a browsing technique which hides objects of no interest and highlights only ones of concern on the screen should be supported to save the limited screen space.

1.2 SOPView+'s Solution

Currently, a number of OODB browsers have been reported in the literature[4][5][6][7][8][9], which will be reviewed in Section 5, but none of them has ever proposed a clever solution for the problem stated above. In order to solve the problem, we introduce the notion of *base object* to use it as a tool to explore the sea of database objects. In SOPView+, base object is an object which is a basis for navigation; *forward navigation* is provided for the reference paths ahead of the base object and *backward navigation* for the ones behind it. After designating an object as a base object, users can navigate its neighborhood using forward and backward navigation. Furthermore, SOPView+ allows users to change the base object along the reference hierarchy among a number of database objects; this makes it possible for them to explore a large database on the screen while changing their target until they locate objects of their interest. This browsing technique thus can help navigating database on the limited screen space, solving the screen real estate problem.

In order to support repetitious short-sighted navigation, SOPView+ also provides *extended synchronized browsing*; it synchronizes displaying objects for both forward and backward navigation, which has improved prior synchronized one supported by many object browsers such as KIVIEW[5], OdeView[6], and PESTO[7].

SOPView+ has the features as follows :

- **Database exploration by changing base object** : It allows users to designate an object as a base object for navigation and change it following the reference hierarchy. This browsing technique can help exploring a large database on the limited screen space, solving the screen real estate problem.
- **Forward and backward navigation** : It permits forward navigation for the reference paths ahead of the base object, and backward navigation for the ones behind it. Backward navigation is supported exploiting the inverse relationship between objects.
- **Extended synchronized browsing** : It supports extended synchronized browsing which synchronizes displaying objects for both forward and backward navigation.

1.3 Paper's Organization

The rest of this paper describes the browsing facilities and demonstration of SOPView+. The remainder of this paper is organized as follows. Section 2 describes the basic browsing facilities of SOPView+, supporting database exploration by changing base object; SOPView+'s support for forward and backward navigation and extended synchronized browsing is also described. Section 3 then demonstrates a browsing session on SOPView+, showing the screen figures of actual browsing. Section 4 briefly highlights a few details of SOPView+'s current implementation and discusses limitations and future plans for SOPView+. Section 5 reviews related work on database user interfaces and explains how SOPView+ differs from existing database browsers. Finally, Section 6 summarizes the paper.

2 Browsing Facilities

In this section we describe SOPView+'s visual elements including notion of *base object* and its basic support for *database exploration by changing base object*, *forward and backward navigation*, and *extended synchronized browsing*.

2.1 Visual Elements

In SOPView+ the basic visual elements for representing objects consist of two object nodes; single object node and collection node as in Figure 2. *Single*

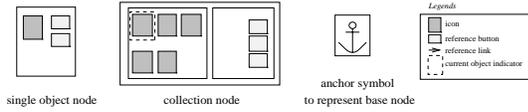


Figure 2: Basic visual elements for representing objects

object node shows a referenced object in database and consists of an icon² to represent it and its reference attributes. *Collection node* displays a collection of referenced objects and consists of a collection of icons and their common reference attributes. With these visual elements the cardinality of reference relationship can be clearly visualized and the reference hierarchy among database objects can be represented by the connection of them via reference links. [10] has details about the visualization structure.

Base object is an object which is a basis for navigation and it is designated by putting an anchor symbol which represents the base object on the object node for it. *Base node* is an object (single object or collection) node on which the anchor symbol is placed. Especially in collection node, current object becomes the base object.

2.2 Database Exploration via Changing Base Object

2.2.1 Long-Distance Navigation

SOPView+ supports exploring database by changing base object; it allows users to drag the anchor symbol and drop it on an object of their interest. Then the object becomes the base object for navigation, and its position is changed to the top-left corner of the screen in case that users does not choose backward navigation discussed earlier. The relationship between object database which has complex and huge reference hierarchy and the visualization structure of SOPView+ is as in Figure 3. In Figure 3 (a) shows the visualization structure of a front part of large reference hierarchy when an object a_1 is designated as the base object; on the contrary (b) displays that when an object e_1 is designated as the base object to browse other rear part of the reference hierarchy. By designating an object as

²Icon is a pictorial attribute which represents an object. Before browsing objects, the user can select an attribute which would be displayed as an icon. If there is no such selection specified, key attribute will be displayed textually by default.

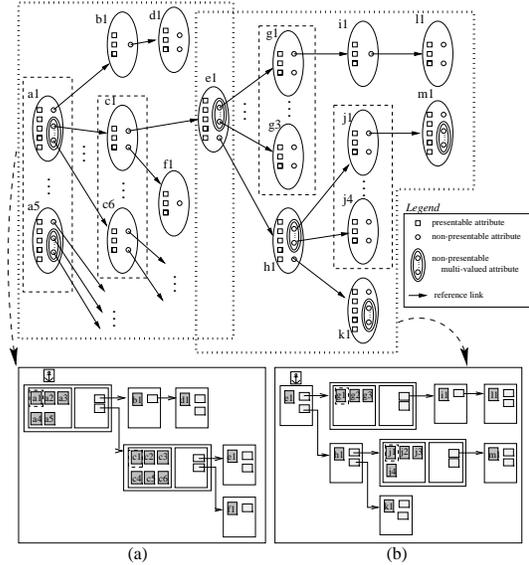


Figure 3: The relationship between object database of complex reference hierarchy and visualization structure of SOPView+

the base of navigation, as you can see in Figure 3, the user can freely navigate an arbitrary part of the reference hierarchy to locate objects of his/her interest, not being restricted by the limited screen space.

2.2.2 Moving-Back via Landmark

On the way of browsing the user can install a *landmark* where he/she wants to come back; the landmark keeps the record of an object node and its current object. After changing the base object, the contents of object nodes in the middle of previous reference path behind the base node are usually changed due to the effect of backward navigation; thus it is hard to recover an object node to the original status only by moving the anchor symbol back. If the user, however, has set a landmark on an object node, it is easy to move back and recover the original status. Landmark can be set in the object node's pop-up menu.

2.3 Forward and Backward Navigation

SOPView+ provides *forward and backward navigation* with the base object between; forward navigation for the reference paths ahead of it and backward navigation for the ones behind it. Suppose that there are three collections of objects A , B , and C , a reference ra_1 of cardinality m from an object a_1 in collection A to those in collection B , and another reference rb_1 of cardinality m from an object b_1 in collection B to

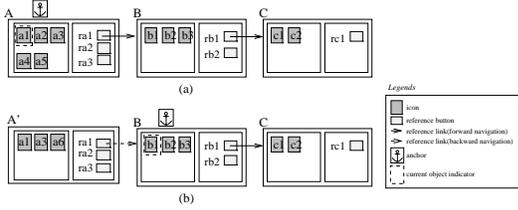


Figure 4: The visualization of reference structure before and after changing base object (in case of cardinality m)

those in collection C . Suppose that the collection node A is the base node and the user is navigating following a reference attribute ra_1 . Then another collection node B which contains objects referenced by the object a_1 will be brought up. Again, if current object in collection node B is b_1 and the user is choosing a reference attribute rb_1 in collection node B , other collection node C of objects referenced by the object b_1 will also be shown up as in (a) of Figure 4. We call this type of browsing as *forward navigation*. In this way, the user can keep navigating following forward reference paths starting from the base node. In (b) of Figure 4 the anchor symbol is moved to collection node B which has objects b_1, b_2 , and b_3 . Now collection node B becomes the base node, and current object b_1 in it the base object. As soon as the user changes the base node, collection node A , previous base node, will be switched to other collection of objects which refers to the base object. We call this type of browsing as *backward navigation*. As a result of changing the base object, forward navigation is executed following forward reference paths from the base node, and also backward navigation is committed following previously navigated reference paths.

In order to support backward navigation, SOPView+ exploits *inverse relationship*[1][2]. Inverse relationship is provided by most OODBMSs to support referential integrity; it is maintained for relationships through the maintenance of inverse-attribute pairs[1]. To provide backward navigation as (b) in Figure 4, SOPView+ displays a collection of objects which are referenced by b_1 through a reference attribute ra'_1 , which is the inverse relationship of ra_1 . If inverse relationship does not exist, backward navigation is no longer supported; previous reference path to base node disappears, and base node is placed and displayed at the top-left corner of the window. Even though inverse relationship exists, backward navigation can be nullified and invisible if the user does not want to display it. If backward

navigation is activated and the user tries to browse reference attribute which is the inverse relationship of backward reference path, SOPView+ shows a warning message to notify that the same reference object can be displayed twice in both direction.

2.4 Extended Synchronized Browsing

SOPView+ provides *extended synchronized browsing* based on forward and backward navigation stated above. Existing synchronized browsing only displays related objects synchronously following forward navigational paths; on the contrary our extended synchronized browsing do the same following both forward and backward ones. In (b) of Figure 4 let us change the base object in the base node B from b_1 to b_2 . Then the objects in collection node A' will be switched to new ones which refers to the base object b_2 . And also the objects in collection node C will be changed to what are referenced by b_2 . It is the extended synchronized browsing technique which displays related objects synchronously following both forward and backward navigation.

3 Browsing Session

In this section we demonstrate SOPView+'s example browsing session. We begin by presenting the schema for a simple university database that will be used in our examples.

3.1 Example Schema

Given in Figure 5 is an object schema for a university database. The schema includes 'Professor', 'Stu-

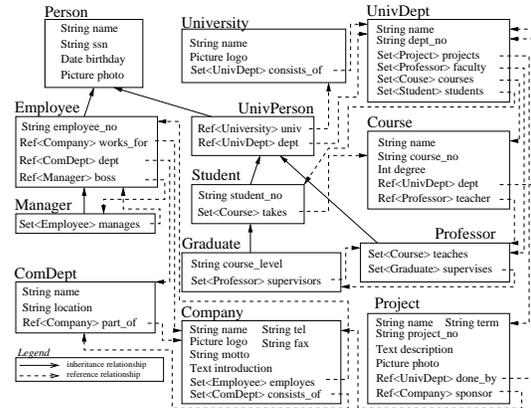


Figure 5: An object schema for a university database

dent', and 'Graduate' classes for modeling members in university, and also 'Employee' and 'Manager' for

company members. ‘Person’ class is a parent class for all of them. Other classes are provided to model other informations such as membership, enrollment, or project activities. Note that the schema contains a number of relationships among classes.

3.2 Phase 1: Database Exploration

3.2.1 Designation of Starting Object

To begin a SOPView+ browse session, the user poses a query and gets a collection of result objects, or chooses an object in the list of names of objects, from which to begin navigation. Let us begin the session by executing an example query as follows : **“Show all the professors who belong to department of computer engineering”**. [10] has details about how to specify the query with visual manner and execute it. Figure 6 shows SOPView+’s reference hierarchy window, which contains a collection node in which a collection of professor objects resides; in Figure 6 the

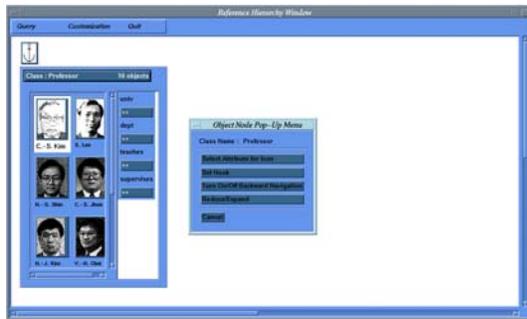


Figure 6: The demonstration of displaying a collection of result objects after query execution

user has selected *photo* attribute as an icon, so the photos of professors have been displayed in the collection node. The collection node is designated as a base node by default.

3.2.2 Forward Navigation

With the query result, the user can navigate along the reference attribute to find objects of his/her interest. In Figure 6 suppose the user has changed current object from ‘C.-S. Kim’ to ‘H.-J. Kim’ and chosen *supervises* reference attribute to see graduates whom he supervises; a collection of Graduate objects then has shown up as in Figure 7. With the collection of Graduate objects, suppose that the user has navigated more to browse department object to which the graduates belong and course objects which they take. A department object to which current graduate ‘S.-W. Chang’ belongs and a collection of course objects

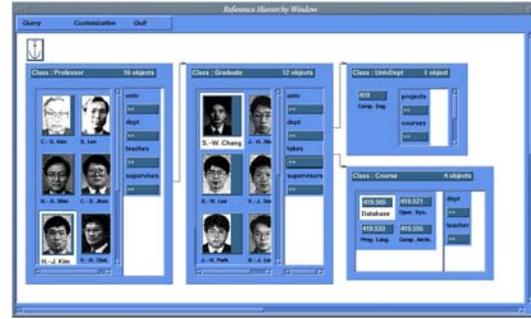


Figure 7: The demonstration of object browsing via forward navigation

which he takes have brought up after the user clicked on the buttons for *dept* and *takes* reference attributes respectively as in Figure 7.

3.2.3 Backward Navigation after Changing Base Object

In Figure 7 suppose that the user wants to know who supervises the Graduate ‘S.-W. Chang’. As discussed earlier the user has to drag the anchor symbol which is on the Professor collection node and drop it onto that for Graduate objects. Current Graduate object ‘S.-W. Chang’ then becomes the base object and backward navigation occurs between the Professor and Graduate collection. The Professor objects in the Professor collection node has switched to those who supervise the Graduate ‘S.-W. Chang’ as in Figure 8. However, nothing happens for the UnivDept single object node and the Course collection node as the current object of Graduate collection is not changed.

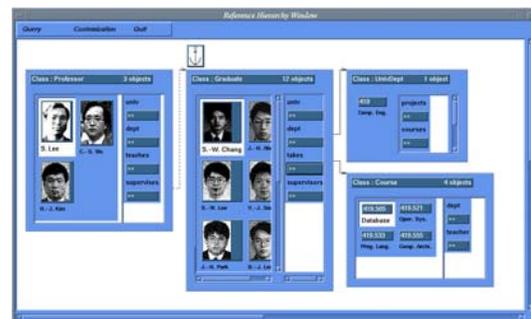


Figure 8: The demonstration of backward navigation after moving the anchor

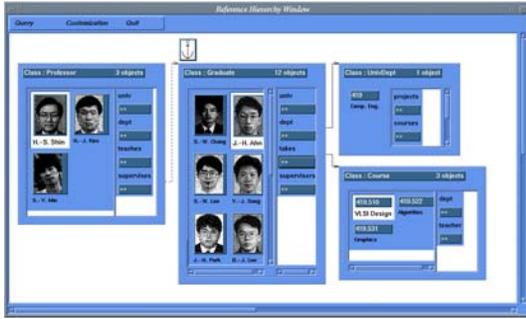


Figure 9: The demonstration of extended synchronized browsing

3.2.4 Extended Synchronized Browsing

In Figure 8 let us step forward from ‘S.-W. Chang’ to ‘J.-H. Ahn’ in the Graduate collection node. The UnivDept object and the Course objects will be switched to one to which the graduate ‘J.-H. Ahn’ belongs and those which he takes; in addition, the Professor objects also will be shuffled to those who supervise him as in Figure 9. It shows the execution of *extended synchronized browsing* which synchronizes displaying objects for both forward and backward navigation.

3.2.5 Long-Distance Navigation

In Figure 8 suppose that the user wishes to designate the UnivDept object as the base for further navigation from it. First the user has to move the anchor symbol to the UnivDept single object node, designating it as the base node. After changing the base node, in order to browse a rear part of reference hierarchy starting from UnivDept node, the user also has to make all the backward reference paths invisible by clicking on ‘TURN OFF BACKWARD NAVIGATION’ button in pop-up menu, placing UnivDept node at the top-left corner of the window. The user then can browse following any reference paths to find objects of his/her interest. Figure 10 shows the demonstration of browsing objects via forward navigation with the UnivDept node as the base node.

3.3 Phase 2: Information Retrieval

Suppose that the user wants to look at the information of Project object and Company object which is its sponsor in Figure 10. If the user double-clicks each object he/she wants to see, the object information window shows up, displaying each object’s information as in Figure 11 In Figure 11 the user has chosen *description* and *photo* attributes to see their information in detail. SOPView+ can display text,

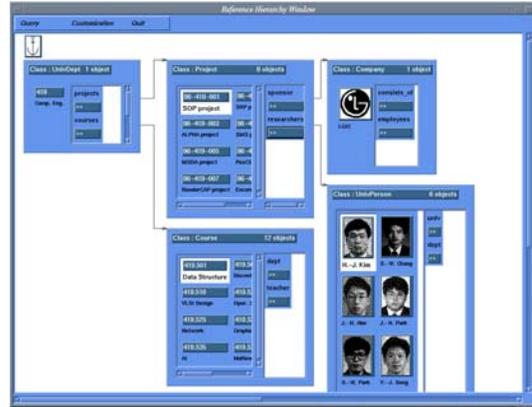


Figure 10: The demonstration of long-distance navigation via changing the base object



Figure 11: The demonstration of browsing information

pictures of diverse formats, and video data as a support for multimedia information.

4 Implementation Details

4.1 Current Status

SOPView+ prototype has been developed in a UNIX environment on SUN workstations. It consists of about 27,000 lines of C++[11] code. Motif[12] widget has been used to facilitate the implementation of the graphical interface. SOPView+ has developed on top of SOP OODBMS as a tool for browsing database objects. Currently, we are trying to port SOPView+ to other OODBMSs such as *O₂* and ObjectStore.

4.2 Limitations and Future Plans

SOPView+ is quite useful in terms of the browsing facilities which allows users to explore the ocean of objects in both forward and backward direction without being restricted by the limited screen space. Of

course, while it provides many helpful browsing facilities, it also has some limitations as follows. First, we assume that objects which are returned after query execution belong to a certain class in the schema, not supporting new objects which are created by relational join, union, and projection. Second, it does not provide any solution to prevent duplicate representation of the same objects. Currently, if the user navigate through two reference attributes which belong to different objects but refer to the same object, the referenced object will be displayed twice in different object nodes.

Our future plans are as follows. First, we are going to supplement the limitations discussed above. Next, we are thinking of porting SOPView+ into Java environment; it would include not only browsing facilities but visual query interface which had presented in [10]. We finally will add some additional features such as customizability by end-users and portability to other OODBMSs which also support ODMG specification[2].

5 Related Work

Many database interface programs for browsing information on DBMS have been proposed. A survey and taxonomy on the programs can be found in [4].

KIVIEW[5], OdeView[6], and PESTO[7] provide *synchronized browsing* of related objects. KIVIEW first proposed the notion to avoid repetitive navigation through reference path. Its internal model is characterized as a semantic network, its browsing session interleaves navigation and manipulation activities. However, it allows users to display only the immediate attributes. OdeView is a system that mostly influenced SOPView. OdeView is a graphical interface for Ode OODBMS, and it provides facilities for examining schema, querying visually, browsing the database, and synchronized browsing. Especially, OdeView overcomes the limited representation of referenced objects in KIVIEW, and represents referenced objects using separate window. However, it does not support the representation of set of objects and browsing facility for long-distance navigation. PESTO is an integrated query/browser which supports both browsing objects and querying them in place. It provides navigational capability, synchronous browsing, and formulation of complex query. However, it says nothing about how to navigate long-distance following reference path and query the objects in place which are far-away from the collection of root(starting) objects.

Visual browsers which displays objects using icon

include IconicBrowser[8] and COMIB[9]. IconicBrowser represents classes and objects which consist of database as icons, and provides facilities for querying visually and browsing result of query using these icons. It, however, does not provide facilities for navigating referenced objects and synchronized browsing. COMIB proposes a browsing method via composite icon. It displays result object together with other objects which are located far-away through composite-icon, which are made by the composition of several icons of objects, providing the overview of object database. With this simultaneous presentation, users can see and compare the attributes of result objects and referenced objects at the same time. However, it also does not provide navigational facilities and synchronized browsing.

In addition to these interfaces, a number of interfaces provide graphical facilities for browsing database schemas and visual query; they include GUIDE, ISIS, SKI, Pasta-3, and SMARTIE. They support in common query specification by graphical editing of the schema and presentation of result objects. [4] has survey and references to these and other systems.

6 Conclusions

We have described SOPView+, a browser for object-oriented databases which supports database navigation by changing the base object, forward and backward navigation, and extended synchronized browsing. We have also demonstrated an example browsing session on SOPView+. In SOPView+ the user can freely navigate an arbitrary part of the reference hierarchy to locate objects of his/her interest, without being restricted by the limited size of screen space just by moving the anchor symbol and designating an object as the base for navigation. Like other browser, SOPView+ allows users to navigate forward a database by following the reference relationships among its objects; in addition SOPView+ supports backward navigation by only moving anchor symbol onto the object which they want to set as a base of navigation. Based on the bi-directional navigation, it has extended existing synchronized browsing technique so as to synchronize objects in both directions. We think that SOPView+ browsing facilities provide intuitive yet useful interface to users who want to browse large database which contains huge and complex reference hierarchy.

Acknowledgements

We would like to acknowledge the comments of the reviewers which improved the presentation. First au-

thor especially acknowledges S.-W. Lee for a number of discussions.

[12] Open Software Foundation, *OSF/Motif Programmer's Guide*, Prentice Hall, 1993

References

- [1] R. G. G. Cattell, *Object Data Management*, p.94, Addison-Wesley, 1992
- [2] R. G. G. Cattell, *The Object Database Standard: ODMG-93, Release 1.1*, Morgan Kaufman Publishers, San Mateo, California, 1994
- [3] C. Batini *et al.*, "Visual Strategies for Querying Database," *Proc. of 1991 IEEE Workshop on Visual Languages*, pp.183-189, 1991
- [4] C. Batini *et al.*, *Visual Query Systems*, Technical Report N. 04.91, Dipartimento di Informatica e Sistemistica, Universita di Roma "La Sapienza", 1991
- [5] A. Motro, A. D'Atri, and L. Tarantino, "The Design of KIVIEW: an Object-Oriented Browser," *Proc. of 2nd Int'l Conf. on Expert Database Systems*, pp. 17-31, 1988
- [6] R. Agrawal, N.H. Gehani, and J. Srinivasan, "OdeView: The Graphical Interface to Ode," *Proc. of Int'l Conf. on Management of Data*, San Diego, California, 1990
- [7] M. Carey, L. Haas, V. Maganty, and J. Williams, "PESTO: An Integrated Query/Browser for Object Databases," *Proc. of Very Large Database 96*, Bombay, India, Sep. 1996
- [8] K. Tsuda, M. Hirakawa, and T. Ichikawa, "IconicBrowser: an Iconic Retrieval System for Object-Oriented Databases," *Journal of Visual Languages and Computing*, Vol.1, No.1, pp.59-76, 1990
- [9] J. Cha and S. Lee, "COMIB: Composite Icon Browser for Multimedia Databases," *Multimedia Tools and Applications*, Vol. 3, No. 3, pp.203-223, 1996
- [10] S.-W. Chang, S. Lee, and H.-J. Kim, "SOPView: A Visual Query and Object Browsing Environment for SOP OODBMS," *Proc. of 20th Int'l Computer Software and Application Conf.*, pp.354-360, 1996
- [11] Bjarne Stroustrup, *The C++ Programming Language*, 2nd Ed., Addison Wesley, 1992