

COSPEX: A System for Constructing Private Digital Libraries

Masanori Sugimoto*

Norio Katayama

Atsuhiko Takasu

Research and Development Department

National Center for Science Information Systems

3-29-1, Otsuka, Bunkyo-ku, Tokyo, 112, Japan

Abstract

In this paper we describe a system called COSPEX (COnceptual SPace EXplorer) which aims at assisting users in constructing their own private digital libraries. Information servers on the Internet have their own purposes. Each of the servers is constructed distributedly and managed autonomously, and its semantics is different from each other. On the other hand, users' information needs are different from each other. Now that almost infinite information resources are accessible through the Internet, a new method for promoting the mediation between the semantics of information servers and users' information needs is inevitable for users to fully utilize these information resources. COSPEX is a system which can enhance such mediation. When users of COSPEX access a certain information server, its database schema is loaded and visualized on the user interface. Users can interactively construct their own queries by viewing the visualized database schema. COSPEX can also assist users in their information discovery. By visualizing the relations among objects and attributes in a database, it assists users in finding their necessary information effectively.

Users of COSPEX can create their own collection from large information resources, that is, a private digital library, which is useful for their intelligent activities.

1 Introduction

In this paper we describe a system called COSPEX (COnceptual SPace EXplorer) which aims at assisting users in constructing their own private digital libraries.

Information servers on the Internet, for instance digital libraries [ACM, 1996; Schatz and Chen, 1996] have their own purposes. Each of the servers is constructed distributedly and managed autonomously, and its

semantics is different from each other. On the other hand, users' information needs are different from each other. Now that almost infinite information resources are accessible through the Internet, a new method for promoting the mediation between the semantics of information servers and users' information needs is inevitable for users to fully utilize these information resources. COSPEX is a system which can enhance such mediation.

COSPEX is composed of three modules, Visual Query Editor (VQE), Conceptual Space Visualizer (CSV) and Private Digital Library Manager (PDLM). VQE assists users in their query formulation. It first loads and visualizes a schema of a database accessed by a user. By viewing it, he/she can easily understand the semantics of the database and how to construct his/her query. CSV assists users in their information discovery. It can configure a metric space to visualize the relations among objects and attributes in a database. By looking at it, users can know the semantic relations among objects and find necessary information effectively. PDLM is a manager of a user's private digital library. It stores and manages object sets which a user has retrieved through VQE and CSV. Users of COSPEX can organize and evolve their own collection, that is, a private digital library from large information resources.

In Section 2 the configuration of COSPEX is described. In Section 3 experiments and evaluations are shown. Section 4 describes related works. Section 5 gives conclusions and future works.

2 Configuration of COSPEX

2.1 Overview

Figure 1 shows an overview of COSPEX. COSPEX is programmed by Java language and is executable on a WWW (World Wide Web) browser like Netscape Navigator. In order to realize the accessibility to multiple information servers and visualize their database schema and object-attribute relations, each information server is augmented with the following information and components for COSPEX:

- Schema information

*Currently a visiting researcher in Department of Computer Science, University of Colorado at Boulder, CO, USA.

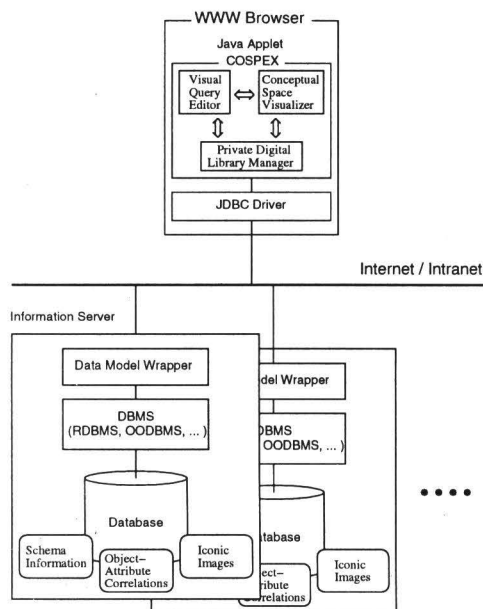


Figure 1: Overview of COSPEX

Each information server provides its own schema information, i.e., the description of its database schema, which is expressed based on the entity-relationship model and consists of the list of classes, predicates, and domain constraints.

- Object-attribute correlations

Each information server provides object-attribute correlations to visualize relations among database contents. Object-attribute correlations are provided in the form of ternary relation, (o, a, c) , where o is an object, a is an attribute, and c is the correlation between o and a . Typical examples of object-attribute correlations are term frequency of documents, color histograms of images, etc. These correlations are either statically stored in databases or dynamically computed on demand.

- Iconic images

Each information server has iconic images to visualize its database schema and to be used for a visual query.

- JDBC drivers

Each information server has its own driver of Java Database Connectivity (JDBC) [Sun Microsystems, 1996]. JDBC is one realization of the concept of Open Database Connectivity (ODBC) and allows Java applets to contact with remote database servers.

- Data model wrappers

COSPEX employs the entity-relationship model as the global data model, i.e., requests to and replies from databases are expressed based on the entity relationship model. Therefore, if an information server uses a DBMS (database management system) of

other data models, e.g. RDBMS (relational DBMS) or OODBMS (object-oriented DBMS), a data model wrapper is introduced to translate the global model to the local model and vice versa.

In this paper COSPEX is used as a query interface for a digital library. However, it is an universal query interface because it is applicable to every information server which satisfies these conditions.

2.2 Visual Query Editor (VQE)

The role of VQE [Katayama *et al.*, 1996] is to assist users in their query formulation. Figure 2 shows an example of a query expression on VQE. It consists of two parts: a palette and a canvas. The palette is located above and contains icons of classes. The contents of the palette is determined by the schema information supplied by each server. The canvas below is a workbench of query expressions. When users access an information server, VQE first loads and visualizes a schema of its database on its palette. They construct their query expressions in the form of a network structure. Retrieved object sets by VQE are stored and managed by PDLM.

For example, the query expression in Figure 2 is formulated as follows:

- (1) A user selects what he/she wants to retrieve or what he/she is interested in by dragging and dropping a class icon from the palette to the canvas. In Figure 2, an article icon is dropped on the canvas.
- (2) By double clicking an icon on the canvas, a user can see a list of predicates applicable to the class. The list is constructed from the loaded schema information and reflects the domain constraints of predicates. By selecting a predicate from the list, a template of a link is laid on the canvas.
- (3) By selecting the class and attached it to the link end, a link between two classes is accomplished. A user repeats this sequence of operations until the expression represents what he/she wants to retrieve.
- (4) A user can assign a constant value to an icon via a dialog box. The dialog box pops up by double clicking an icon with pressing a shift key. In Figure 2 a character string, "John Smith," is assigned to the name of an author.
- (5) Finally, a user specifies what to retrieve. By clicking an icon, he/she can select the query target. The selected icon is marked with a round rectangle. For example, an article icon is selected for the query target in Figure 2. This expression is ready to be issued and means that there exists an article, it is written by an author, and the author has a name, "John Smith."

A formulated expression is converted into a logic-based query expression and issued to the information server. For example, the expression in Figure 2 is converted into the following expression:

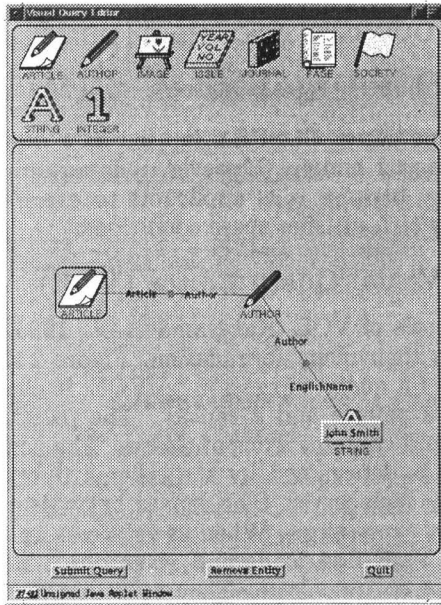


Figure 2: Visual Query Editor

```
[ Article=$article, Author=$author ],
[ Author=$author, Name="John Smith" ],
[ Class=/class/article, Instance=$article ],
[ Class=/class/author, Instance=$author ]
```

The first predicate corresponds to the link between an article and its author. The second corresponds to the link between the author and the author's name. The third and fourth specifies domain constraints for articles and authors. The correspondence between the entity relationship model and predicate logic is very simple. A class icon with no assignment corresponds to a variable, an icon with an assignment to a constant, and a link to an atom.

The major feature of VQE is that it can guide users to formulate their query expressions interactively by visualizing a database schema information loaded from the server. By viewing it, users can interpret its semantics and know how to construct syntactically valid query expressions which retrieve what they need.

2.3 Conceptual Space Visualizer (CSV)

The role of CSV is to help users to select and discover objects by visualizing object-attribute relations [Sugimoto *et al.*, 1995a]. CSV starts with receiving an object set stored in PDL or retrieved by VQE. Figure 3 shows an example of CSV which shows the relations among objects (in this case, articles) and their attributes (in this case, terms) in a two-dimensional metric space. By viewing it, users can know their semantic relations. Through the interaction between users and CSV, they can find new objects or new relations among objects, and reflect them to their query expressions on VQE. Retrieved objects by using CSV are stored and managed by PDL.

CSV is composed of two submodules, an automatic in-

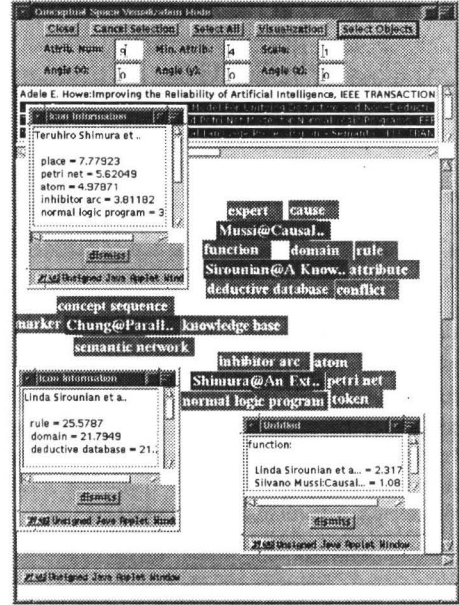


Figure 3: Conceptual Space Visualizer

dexing submodule and a statistical analysis submodule. By loading the corresponding object-attribute correlations from the server, CSV starts with an automatic indexing submodule. It configures an attribute vector for each object by using an automatic indexing algorithm developed by Salton [Salton, 1975] with our own extension. The weight of *attribute j* in *object i* among *n* objects (w_{ij}) is calculated as follows:

- (1) Calculate the densities of *attribute j* in *object i* (d_{ij}) and in *n* objects ($\bar{d}_j$)

$$d_{ij} = \frac{\text{frequency of attribute } j \text{ in object } i}{\text{frequency of all the attributes in object } i}$$

$$\bar{d}_j = \frac{\text{frequency of attribute } j \text{ in } n \text{ objects}}{\text{frequency of all the attributes in } n \text{ objects}}$$

- (2) Calculate the distribution of *attribute j* (v_j)

$$v_j = \frac{\sum_{i=1}^n (d_{ij} - \bar{d}_j)^2}{n-1}$$

- (3) Calculate the weight of *attribute j* in *object i* (w_{ij})

$$w_{ij} = \frac{d_{ij} \times v_j}{\bar{d}_j^2}$$

CSV successively starts a statistical analysis submodule. This submodule implements the dual scaling method (See [Nishisato, 1980] for details) and analyzes attribute vectors in order to visualize the relations among objects. Most of statistical analyses clarify the relations only among objects (in this case, articles) in data. The main feature of the dual scaling method is that it clarifies the relations not only among objects but also among attributes (in this case, terms) and among objects and attributes by giving them scores.

Through these calculations, the submodule creates a score table in which the scores of the objects and their

attributes are described. CSV then configures a space which visualizes object-attribute relations. Figure 3 shows the relations among five articles. From this space users can interpret the semantic relations, for example, the articles 'Mussi@Causal..' and 'Siounian@A Know..' have a relation with each other by the term 'function,' and that the terms 'semantic network' and 'knowledge base' are key words for 'Chung@Parall..'

By looking at the space, users can find articles which they would like to read. When users cannot make their information needs clear at the beginning of their search processes [Taylor, 1968; Belkin *et al.*, 1982], they can articulate their needs by discovering new terms.

CSV enables users their interactive operations. When they discover interesting objects or attributes on a space which CSV configured, they can select a region around them. Then a different space is configured where new attributes and detailed relations among the selected objects are shown. Such interactions between users and CSV can enhance their information retrieval process because relevance feedback [Salton, 1989] is effectively realized.

CSV can visualize the ternary relation in the form of (o, a, c) where o is an object, a is an attribute, and c is the correlation between o and a . In Figure 3 CSV applies its automatic indexing submodule and statistical analysis submodule to the ternary relation (o, a, c) where o is an article, a is a term, and c is the correlation between an article and a term. The choice of object-attribute relations and definition of their correlations, i.e., what attributes to be used and what algorithm to be used for calculating correlations, are unrestricted and at the disposal of database administrators. For example, CSV can also visualize journal-term relations (based on the term frequency of each journal), academic society-author relations (based on the number of papers of each author carried in the publications of each society), and so on.

2.4 Private Digital Library Manager (PDLM)

PDLM is a manager of a user's own private digital library. Figure 4 shows an example of PDLM. When a user selects one of the accessible information servers, its contents are shown above and the contents of his/her private digital library are shown below on PDLM. Then a user can start VQE in order to construct query expressions or start CSV in order to visualize object-attribute relations. A workspace shown in the middle of PDLM is used to store objects retrieved through VQE and CSV.

PDLM has two major functions as a manager of a private digital library. The first function is the management of object sets retrieved by users using VQE and CSV. Object sets can be freely added to or removed from their private digital libraries to compose their useful collection. This function captures the fundamental function of the classical, i.e., non-digital, private libraries.

In order to make a desirable object set, a user can also carry out operations such as intersection, union, and difference on object sets which are retrieved from the

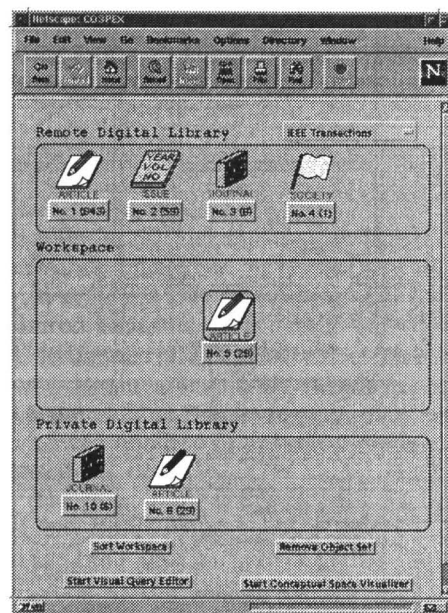


Figure 4: Private Digital Library Manager

same server. The second function which is currently in the planning and implementing stage is that it can autonomously collect object sets by accessing information servers on the Internet. PDLM stores not only object sets but also rules which has been used to retrieve them. Rules stored in PDLM are query expressions constructed by a user on VQE and classification rules generated by a decision tree algorithm [Quinlan, 1993] on CSV. By using these features PDLM can assist users in organizing and evolving their own private digital libraries.

2.5 Usage of COSPEX

Figure 5 illustrates an example of an usage of COSPEX. Users interact with VQE, CSV and PDLM alternately. In this example, a user first selects one of the accessible information servers from his/her URL (Universal Resource Locator) list.

(a) Query formulation.

Suppose that the user thinks of a keyword "agent." Then, he/she starts VQE and formulates a query to retrieve articles with a term "agent."

(b) Target alteration.

If the query result is too large to be displayed or visualized, COSPEX allows users to alter the query target for seeking more suitable search conditions. In this example, the target is altered from articles to journals to obtain a smaller result. This operation corresponds to altering the abstraction degree of the query target.

(c) Attribute discovery.

Once the query result is small enough to be visualized, CSV enables users to visualize relations among

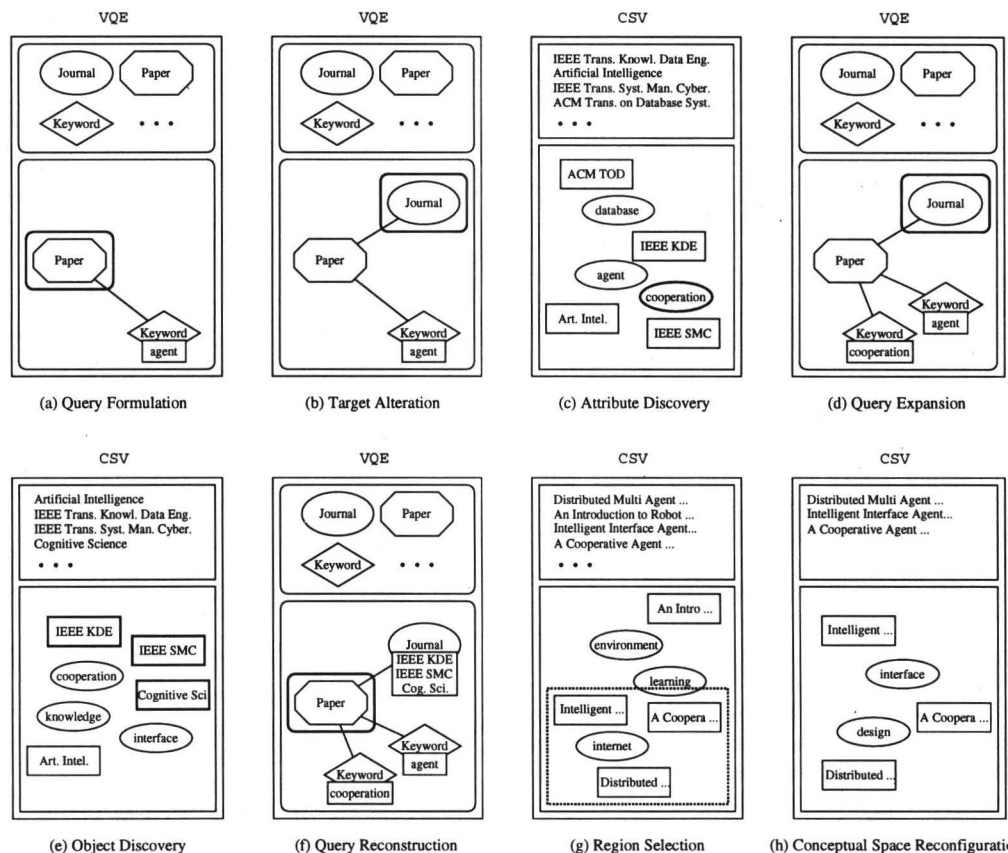


Figure 5: An example of an usage of COSPEX

objects and their attributes. In this example, relations among journals and their key terms are visualized. In this example, rectangles and ovals represent journals and terms, respectively.

One of the distinctive features of CSV is the discovery of attribute values. Because CSV places not only objects but also attributes within a single conceptual space, users can discover new attribute values relative to the objects. Discovered values can be used for query expansion in the next step.

(d) Query expansion.

Suppose that the user discovered a new term "cooperation" in step (c). Then, he/she can expand the query by adding the new term to the expression. Now, the expression means to retrieve journals which include articles with terms, "agent" and "cooperation."

(e) Object discovery.

In this step, the user visualizes relations among journals which are obtained by the query in step (d). The configured conceptual space is different from the one in step (c) because retrieved journals are reduced by the query expansion in step (d).

This step illustrates another feature of CSV, i.e., the discovery of objects. CSV enables users to discover

objects which attract their interest. In this example, the user discovers that the journals, "IEEE KDE," "IEEE SMC," and "Cognitive Sci.," are relevant to his/her interest and that the others are irrelevant. Then, he/she can pick up relevant objects and feed them to the query expression of VQE.

(f) Query reconstruction.

Objects selected in CSV are fed into the query expression and disjunctive search condition is formulated in VQE. This expression means to retrieve articles which have terms, "agent" and "cooperation," and which are included in "IEEE KDE," "IEEE SMC," or "Cognitive Sci."

(g) Region Selection.

In this step, CSV visualizes articles obtained in the previous step. This step illustrates the interactive operation to a space configured by CSV. When a user feels interested in specific terms or articles, he/she can select a region around them. Then a different space is configured where new terms and detailed relations among selected objects are shown.

(h) Conceptual Space Reconfiguration.

This step illustrates the reconfigured conceptual space. It shows newly weighted key terms and detailed relations. This interactive operation to CSV

enables users to reduce the result set by analyzing relations among them.

Users finally find their necessary objects and store them to PDL as new components of their own private digital libraries. As described above, COSPEX assists users in their information retrieval by enhancing the mediation between the semantics of information servers and their information needs.

3 Experiments and Evaluations

Many experiments with our system have been carried out so far [Sugimoto *et al.*, 1995b]. Here we describe the experiments about the visualization effects of CSV. Nine subjects consisting of graduate students and staff of Tokyo University used CSV in the following three different modes of visualization capability:

- (V0): No visualization (Boolean search only)
- (V1): Visualization of object relations
- (V2): Visualization of object-attribute relations

They rated each mode (very bad = 1; bad = 2; average = 3; good = 4; very good = 5) by considering the following question: could you discover objects or attributes effectively?

The hypothesis test¹ shows that users can discover objects and attributes more effectively in (V1) than (V0) at the 0.005 level of significance and in (V2) than (V1) at the 0.025 level of significance. This shows that (V2) is more effective than (V0) and (V1) in helping users discover objects and attributes. From the post-experiment interviews users told that the visualization of object-attribute relations is useful for the articulation of their information needs.

Table 1 shows the retrieval effectiveness on (V0) and (V2). Subjects are asked to search articles which they need. The fact that (V2) is higher than (V0) in recall and precision ratio explains that CSV can assist users in finding their necessary articles effectively.

4 Related Works

A large number of visual interfaces have been proposed in the database literature [Goldman *et al.*, 1985; Czejdo *et al.*, 1990; Vadaparty *et al.*, 1993; Mohan and Kashyap, 1993; Miura and Moriya, 1992; Miura, 1994; Merz and King, 1994; Balkir *et al.*, 1996]. Compared with these predecessors, the current implementation of VQE has limited expressive power in terms of query formulation. We need to extend its capability to permit more complex queries. However, VQE is superior to them in terms of database schema visualization because its interactive visualization technique is unique to VQE and helps users to capture database schemata and guides them to formulate syntactically valid query expressions.

¹We carried out one-sided and paired t-tests, because the samples are paired and small.

	recall	precision
(V0) mode of CSV	50%	63%
(V2) mode of CSV	58%	78%

Table 1: A summary of the retrieval effectiveness

Many techniques for visualization and interactive operations have been proposed in the studies of information retrieval systems. I³R [Thompson and Croft, 1989] is a distributed expert system based on a blackboard model. It configures a space by calculating the similarity among documents. BEAD [Chalmers and Chitson, 1992] also configures a document space by calculation based on the analogy of potential fields in physics. These two systems don't visualize object-attribute relations (e.g. document-term relations) simultaneously but object relations. InfoCrystal [Spoerri, 1993] assists users by visualizing their vector-space searches and Boolean searches. It differs from CSV because it only shows the search terms which users input and doesn't show the other weighted terms and relations among them. Scatter/Gather [Cutting *et al.*, 1992], which classifies a large collection of documents into a small number of group, differs from CSV because it doesn't visualize users' search processes. In LyberWorld [Hemmje *et al.*, 1994] users navigate along the links between documents and terms in a space during their search processes. CSV, on the other hand, configures a space where all the objects are visualized and users can select a specific region from it.

In the current configuration of COSPEX, we employ a wrapper mechanism to achieve the interoperability of multiple digital libraries. However, more sophisticated mechanisms are being studied in the digital library literatures [Atkins *et al.*, 1996; Paepcke *et al.*, 1996]. For example, the UMDL (University of Michigan Digital Library) project proposes a distributed agent architecture which permits multiple specialized agents to be dynamically coordinated and work together in order to accomplish a given task [Atkins *et al.*, 1996]. This architecture promotes the modularity, flexibility and incrementality of digital library services. This kind of architecture is not incompatible with the architecture of COSPEX. It can be applicable to COSPEX and seems to be effective for promoting the modularity of its implementation.

5 Conclusions and Future Works

In this paper we proposed a system called COSPEX. By enhancing the mediation between the semantics of information servers and users' information needs, COSPEX assists users in constructing their own collection, that is, a private digital library from multiple information servers on the Internet.

We have several plans to extend the functions of COSPEX. One of them is to apply it for a collaboration testbed. When object sets and search rules are stored and shared by several users, COSPEX can assist users in their knowledge sharing and facilitate their col-

laboration. Class identification and object identification are also critical in managing a private digital library. A user collects objects from multiple information servers whose semantics are different from each other. In order to integrate the object sets from multiple information resources, a new class may need to be defined by merging existing classes through the interaction with a user. We will implement these extensions to make COSPEX more useful.

References

- [ACM, 1996] ACM. *First ACM International Conference on Digital Libraries*, 1996.
- [Atkins et al., 1996] D. Atkins, W. Birmingham, E. Durfee, et al. Toward Inquiry-Based Education Through Interacting Software Agents. *IEEE Computer*, 29(5):69 – 76, 1996.
- [Balkir et al., 1996] N. H. Balkir, E. Sukan, G. Ozsoyoglu, et al. Visual: A graphical icon-based query language. In *Proc. of 12th Int. Conf. on Data Eng.*, pages 524 – 533, Feb 1996.
- [Belkin et al., 1982] N. J. Belkin, R. N. Oddy, and H. M. Brooks. Ask for information retrieval: part I. background and theory. *Journal of Documentation*, 38(2):61 – 71, June 1982.
- [Chalmers and Chitson, 1992] M. Chalmers and P. Chitson. BEAD: Explorations in Information Visualization. In *Proc. of SIGIR'92*, pages 160 – 169, 1992.
- [Cutting et al., 1992] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proc. of SIGIR'92*, pages 318 – 329, 1992.
- [Czejdo et al., 1990] B. Czejdo, R. Elmasri, M. Rusinkiewicz, et al. A graphical data manipulation language for an extended entity-relationship model. *IEEE Computer*, 23(3):26 – 36, Mar 1990.
- [Goldman et al., 1985] K. J. Goldman, S. A. Goldman, P. C. Kanellakis, et al. Isis: Interface for a semantic information system. In *Proc. of SIGMOD'85*, pages 328 – 342, May 1985.
- [Hemmje et al., 1994] M. Hemmje, C. Kunkel, and A. Willett. LyberWorld - A Visualization User Interface Supporting Fulltext Retrieval. In *Proc. of SIGIR'94*, pages 249 – 259, 1994.
- [Katayama et al., 1996] N. Katayama, M. Sugimoto, and J. Adachi. A universal query interface for heterogeneous distributed digital libraries. In *Proc. of the 7th Int. Workshop on Database and Expert System Applications (DEXA'96)*, pages 332 – 339, Sep 1996.
- [Merz and King, 1994] U. Merz and R. King. Direct: A query facility for multiple databases. *ACM Trans. Inf. Syst.*, 12(4):339 – 359, Oct 1994.
- [Miura and Moriya, 1992] T. Miura and K. Moriya. On the completeness of visual operations for a semantic data model. *Data & Knowl. Eng.*, 9(1):19 – 44, 1992.
- [Miura, 1994] T. Miura. Nesting quantification in a visual data manipulation language. *Data & Knowl. Eng.*, 12(2):167 – 196, Mar 1994.
- [Mohan and Kashyap, 1993] L. Mohan and R. L. Kashyap. A visual query language for graphical interaction with schema-intensive databases. *IEEE Trans. Knowl. and Data Eng.*, 5(5):843 – 858, Oct 1993.
- [Nishisato, 1980] S. Nishisato. *Analysis of Categorical Data: Dual Scaling and Its Application*. Univ. of Toronto Press, 1980.
- [Paepcke et al., 1996] A. Paepcke, S. B. Cousins, H. Garcia-Molina, et al. Using Distributed Objects for Digital Library Interoperability. *IEEE Computer*, 29(5):61 – 68, 1996.
- [Quinlan, 1993] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan-Kaufmann, 1993.
- [Salton, 1975] G. Salton. *Dynamic Information and Library Processing*. Prentice-Hall, 1975.
- [Salton, 1989] G. Salton. *Automatic Text Processing - The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Publishing Co., 1989.
- [Schatz and Chen, 1996] B. Schatz and H. Chen. Digital Library Initiatives. *IEEE Computer*, 29(5), 1996.
- [Spoerri, 1993] A. Spoerri. InfoCrystal: A visual tool for information retrieval & management. In *Proc. of the 2nd Int. Conf. on Information Knowledge Management*, pages 11 – 20, 1993.
- [Sugimoto et al., 1995a] M. Sugimoto, K. Hori, and S. Ohsuga. A Document Retrieval System for Assisting Creative Research. In *Proc of the 3rd Int. Conf. on Document Analysis and Recognition (ICDAR'95)*, pages 167 – 170, Aug 1995.
- [Sugimoto et al., 1995b] M. Sugimoto, K. Hori, and S. Ohsuga. A System for Assisting Creative Research Activity. In *Proc. of 6th Int. Conf. on Human-Computer Interaction*, pages 685 – 690, Jul 1995.
- [Sun Microsystems, 1996] Sun Microsystems. *JDBC: A Java SQL API*, 1996. <http://splash.javasoft.com/jdbc/index.html>.
- [Taylor, 1968] R.S. Taylor. Question-Negotiation and Information Seeking in Libraries. *Colleges and Research Libraries*, 29:178 – 194, 1968.
- [Thompson and Croft, 1989] R. H. Thompson and W. B. Croft. Support for browsing in an intelligent text retrieval system. *Int. Journal of Man-Machine Studies*, 30:639 – 668, 1989.
- [Vadaparty et al., 1993] K. Vadaparty, Y. A. Aslandogan, and G. Ozsoyoglu. Towards a unified visual database access. *SIGMOD Record*, 22(2):357 – 366, Jun 1993.