ELECTRONIC WORKSHOPS IN COMPUTING

Series edited by Professor C.J. van Rijsbergen

Jessie Kennedy and Peter Barclay (Eds)

Interfaces to Databases (IDS-3)

Proceedings of the 3rd International Workshop on Interfaces to
Databases, Napier University, Edinburgh, 8-10 July 1996

Paper:

A Graphical User Interface to the Object-Oriented Database System VODAK on the Basis of the Generic Visualisation Toolkit Lyberworld

Thomas Schweickert and Matthias Hemmje

Published in collaboration with the
British Computer Society

A Graphical User Interface to the Object-Oriented Database System VODAK on the Basis of the Generic Visualisation Toolkit Lyberworld

Thomas Schweickert, Matthias Hemmje

GMD-IPSI,

Dolivostr. 15, 64293 Darmstadt, Germany,
email: {schweike, hemmje}@darmstadt.gmd.de

Abstract

With the Lyberworld system a new approach offering the user a convenient interface to the data stored within an information retrieval system has been introduced. Accessing database contents by using the visualisation metaphors *search tree* and *relevance sphere* has achieved two main benefits: first, the user can construct and pose queries much easier than using a query language and second, query results are presented in a way which in opposition to, e.g., tabular output displays the relations between the different queries and query results. In our most recent research work we are aiming at integrating the visualisation concepts of Lyberworld with the object oriented database system VODAK, a powerful tool to implement database applications which require a complex database schema. Posing queries against such a database is a complex task which requires profound knowledge of the database schema and a query language. The approach presented within this paper integrates the Lyberworld concepts with the requirements for a visual interface to the VODAK database system. Furthermore, it develops Lyberworld towards a system which integrates and visualises data from various heterogeneous data sources.

1 Introduction

With the Lyberworld system an new approach offering the user a convenient interface to the data stored within an information retrieval system [8] has been introduced. Accessing database contents by using the visualisation metaphors *search tree* and *relevance sphere* has achieved two main benefits: first, the user can construct and pose queries much easier than using a query language and second, query results are presented in a way which in opposition to, e.g., tabular output displays the relations between the different queries and query results.

In our most recent research work we are aiming at integrating the visualisation concepts of Lyberworld with the object oriented database system VODAK [9], a powerful tool to implement database applications which require a complex database schema. Posing queries against such a database is a complex task which requires profound knowledge of the database schema and a query language [1].

The approach presented within this paper integrate the Lyberworld concepts with the requirements for a visual interface to the VODAK database system. Furthermore, it develops Lyberworld towards a system which integrates and visualises data from various heterogeneous data sources.

2 Object Oriented Database Systems

Whereas the relational database systems (RDBS) nowadays form the standard for database applications in the business and administrative area, the object oriented database systems (OODBS) have gained more and more importance in the last ten years. While the strength of the relational data model lies in its simplicity and mathematical foundation, the various object oriented data models offer concepts which allow direct mapping of real world aspects to database schemas. The main application areas of object oriented database systems are those which require a complex data model.

The first generation of object oriented database systems evolved as extensions of object oriented programming languages. These extensions provided persistency, which means objects endure program runtime and can be accessed by different users. A second generation of OODBS distinguish themselves from earlier approaches by

a unified data model independent from a certain programming language. Like RDBS, these systems have abilities like transaction management, data recovery and use dedicated languages for schema definition, data manipulation and query formulation.

VODAK [9] is such a second generation OODBS. It uses the language VML for schema definition and application development. VML contains descriptive elements for schema definition and procedural elements for defining object behaviour and accessing database contents by means of navigation. For query formulation the declarative language VQL is provided which can be seen as an object oriented extension to SQL. It allows set-oriented access to database contents.

3 3D Information Visualisation

In 3D information visualisation two main aspects are of particular importance: the presentation of information and the interaction between users and the computer.

The first aspect to be considered is that of presentation. In [10] a taxonomy of different approaches for database visualisation can be found. Figure 2 illustrates these approaches and shows, how the structure of the data to be visualised can be seen as a criterion for which approach to choose.

Figure 1: Taxonomy: Approaches for Information Presentation

In the “benediktine approach“, there is a direct mapping between spatial dimensions and object attributes. When using similarity relations, the spatial position of an object reflects its relation to other objects by means of a similarity measure; similar objects have a small spatial distance. Using hyperstructures enables the display of structural relationships between objects. Finally, in many cases one can think of a “human centered approach” where the presentation of objects reflects the way a person thinks and acts in a special working context. So called “Real world” metaphors play a major role in these approaches (e.g. Bookhouse [12]).

When visualising a database, the following aspects can be considered:

1. Database structure
2. Database contents
3. Manipulation of database contents
4. Queries to a database

An advance to present a unified graphical representation for all of these tasks can be found in [5]. It is not always necessary to visualise all these aspects. In [14] two main tasks for database visualisation are named: *graphical query formulation* and the *displaying of query results* (database contents). The restriction to these two tasks makes sense since non expert users of databases are concerned mainly with these tasks. Dealing with the database structure and manipulation of database contents is the domain of experts who are accustomed to using formal description and query languages.

There are many approaches for graphical query formulation. These differ in the degree the users can influence the query composition process. Sometimes, the users are given a graphical metaphor with an expressiveness similar to a formal query language [2]. Thus the user is able to formulate deliberate queries. The disadvantage is, that the task of query formulation despite of being easy still is very time consuming and still requires some knowledge about the schema of the underlying database. Examples are the "filter-flow"-principle for formulation of boolean queries to a set-oriented database presented in [14] or the schema-oriented approach for querying an OODBS as shown in [3]. The opposite approach to deliberate query formulation is providing the user with a fixed amount of queries. Flexibility is retained by parameterisation of these queries. This is the approach the Lyberworld-system [8] and the system presented in this paper are based on. A way in the middle is the "Query by Icons"-System as shown in [11]. This system derives possible queries from the database schema and presents them by means of iconic and natural language descriptions.

The way query results are displayed *in detail* should reflect the characteristics of the data displayed. Geographical data could be displayed in form of a map, for biomolecular data a viewer for the molecular structure can be useful. *Relations* between data items can be visualised using graphs. It is an advantage if there is no strict separation between query formulation and result display. When the same metaphor is used for these two tasks, the user is given an immediate feedback whether his query actions were successful.

3.1 Lyberworld

Lyberworld [8] is a threedimensional, interactive user-interface to an IR system. It meets the basic requirements to such an interface: posing queries and displaying the query results. Furthermore, it provides additional important features such as using visual metaphors, allowing query results to be the starting point for new queries, transparency of the query history, relevance judgement, filtering and clustering of results.

The strength of Lyberworld lies in its information model (figure 2) which goes beyond the data model of the underlying IR system. That means, data items are arranged in a way that allows the user to recognise coherences between data items not explicitly stored in the underlying database.

Figure 2: Information model of Lyberworld

Due to his current working task the user has a only vague defined set of what could be called "potentially interesting documents". He wants to approximate this set with documents extracted from the database. This approximation is called the context set which contains documents and terms and is represented by the conetree metaphor. The user can extend the context set by expanding the conetree. He can do this by means of two different queries, one of which retrieves documents relevant for a term and another which retrieves terms relevant for a document.

The context set grows with every expansion, which makes it necessary to provide the user with a tool to filter a manageable amount of information from this set. The relevance sphere is a metaphor which supports this filtering process by implementing visual clustering and filtering techniques as described in greater detail in [8].

3.2 A Toolbox for Information Visualisation

Some Application frameworks for two-dimensional user-interface-development (such as the IRIS View Kit [15]) provide the developer with high level components such as components for two-dimensional visualisation of graph structures. In the area of three-dimensional interface-development such development supports are still missing. As a consequence, in the Lyberworld project the metaphors were separated to form a Toolbox for information visualisation. The objectives were to give the redesign a more modular structure and to make the tools universally applicable. Up to now, the toolbox comprises of the *visual tree component* for the presentation of hierarchical structures and

the *relevance sphere component* for the presentation of weighted multidimensional dependencies.

The main requirements for these components were

- Optimisation of performance.
- Improved usability using results from usability studies done for the first Lyberworld version. Especially fully direct manipulation should be achieved.
- Independence from a certain database application, i.e., a generic implementation

4 Concept

By satisfying the above listed requirements, we have developed a visualisation tool which allows quick development of an application in the sense of the "rapid prototyping" concepts.

4.1 An Example Application

As an application, we have the receptor-ligand database ReLiBASE from the Docking-D project [7]. The main objectives of this project are the integration of heterogeneous data in the area of biomolecular research and algorithmic analysis of this data. Up to now, the database occupies about 500 MB disk space. The conceptual database schema can be seen in figure 3.

Figure 3: Conceptual Database Schema

4.2 User Groups

Within discussions with industrial project partners we have been able to determine two groups of potential users for the toolkit. The first group consists of experts experienced with the schema of a certain database application as well as with the query language VQL. People in the second group of users have profound knowledge in the domain of a certain database application, e.g. in the area of biomolecular research, but are quite unfamiliar with database systems. In our system design scenario, the first group configures the visualisation system to the needs of the second group.

4.3 Modelling an Information Network

Lyberworld already used a visualisation schema independent from the schema of the underlying information-retrieval-system. This resulted in the context set. But the semantic of this context set had some restrictions. The

context set only contained objects of the two classes "document" and "term". Only two kinds of similarity relations between these objects were provided.

Figure 4 displays the role of the information network. First, it fulfils the task of an information workspace as introduced in [13]. The user doesn't have to deal with the specific access protocol of a certain data source, he just has to deal with the access mechanisms of the information network. Second, it isn't necessary to tailor a user interface to the specific needs of each data source.

Figure 4: Information Model

Figure 5 shows an example network. The elementary information item in the network is the data object. Each object belongs to exactly one class. Coherencies between objects are expressed by relations. We currently differentiate three kinds of relations. In the example, the collection $R1$ of objects $\{p1, p2, p3\}$ builds a relation of type R_N . $R2$ relates the object $p1$ to the objects $\{l1, l2\}$, this relation is of type R_{1N} . Finally, $R3$ relates the object $p3$ to the objects $\{c1, c2, c3\}$; as this relation contains additional weights, it is of type R_{1Nw} . Each relation has a class which is derived from the classes of the objects which constitute the relation.

Classes:

Protein, Ligand, Complex

Objects:

$\{p1, p2, p3\} \subseteq \text{Protein}$

$\{l1, l2\} \subseteq \text{Ligand}$

$\{c1, c2, c3\} \subseteq \text{Complex}$

Relations:

$R1 \in R_N, R2 \in R_{1N}, R3 \in R_{1Nw}$

Relation classes:

class $(R1) = \langle \text{Protein} \rangle$

class $(R2) = \langle \text{Protein, Ligand} \rangle$

class $(R3) = \langle \text{Protein, Complex} \rangle$

Figure 5: Information Network

The information network forms a tree structure. It is constructed by a sequence of queries against a data source. Objects of the information network can become parameters for new queries. This means, that the underlying queries also form a tree structure. Figure 6 displays the query tree corresponding to the information network of figure 5.

Figure 6: Query Network

The result of a query is a new relation which contains the newly retrieved objects and which expresses the relation between the query parameter and these new objects. Hence, to each type of relation there exists a corresponding query type:

Query Type	Description
Select<C>	Selection query without a parameter. Result of the query is a Relation $R \in R_N$ with $\text{class}(R) = \langle C \rangle$
Select< C_1, C_2 >	Selection query with an object of class C_1 as parameter. Result of the query is a Relation $R \in R_{1N}$ with $\text{class}(R) = \langle C_1, C_2 \rangle$
Eval< C_1, C_2 >	Evaluation query with an object of class C_1 as parameter. Result of the query is a Relation $R \in R_{1Nw}$ with $\text{class}(R) = \langle C_1, C_2 \rangle$

4.4 Configuring the Information and Query Network

The above described information network is independent from a specific data source. If the OODBS VODAK is used as a data source, a mapping between VODAK and the information network has to be defined. This mapping is realised by means of a configuration language. Example 1 shows a part of such a configuration file. It names the visualisation classes `Protein_keyword` and `Protein` together with visual attributes, in this case the colours to be used when objects of these classes are to be displayed.

```

STRING %DB = "db1"

class Protein_keyword                                # Visualisation classes
  colour blue
class Protein
  colour green

query Protein_keyword                                # Query 1
  title "All protein keywords"
  selection
    access "protein.keyword"
    from   "protein IN %DB::Protein"

query Protein from Protein_keyword                    #Query 2
  title "All proteins for a given keyword"
  selection
    access "protein, protein.name"
    from   "protein IN %DB::Protein"
    where  "protein.keyword = '%NAME'"

viewer from Protein                                  #Viewer
  title "Show molecule structure"
  vieweragent "RasMol"
  path "/home/embl/ReLi/data"
  file "%.ent"
  
```

Example 1

The mapping is achieved by parameterisation of VQL-queries and interpretation of VQL-results. The first part of a query description describes type and class of the relation the query produces. In the second part, the VQL-query is described.

4.5 Mapping the Networks to the Visualisation Tools

The information network has to be mapped to the visualisation tools. This mapping has to implement the two aspects presentation and interaction.

The *search tree* plays a special role as it used to build information network and the query network. The presentation of the information network from figure 5 is shown in figure 7.

Figure 7: Mapping to the Tree

Queries and viewers can be seen as methods of visualisation classes and objects. Queries available for an object or a class are displayed by means of a context menu. The entries of the context menu are labelled with the titles of the queries and viewers as set in the configuration file. The user interacts by selecting an object and choosing a query or a viewer from the objects context menu. As a consequence, either a query is executed and the tree expanded or a viewer for the object is started.

In opposition to the tree, the *relevance sphere* displays only parts of the information network. It allows the presentation of multiple relations of type R_{INw} . These relations are accumulated as shown in figure 8. The resulting relation is of type R_{NMw} and can be presented using the visual sphere tool as it is shown in figure 9. Similar to the search tree, viewers can be started for the objects shown in the sphere.

Figure 8: Relation of Type R_{NMw}

There are three possible strategies which relations of type R_{INw} to combine to a relation of type R_{NMw} :

1. Combine deliberate relations
2. Combine relations of the same class

- Combine relations which are result of the same query

Figure 9: Representation of a Relation as a Relevance Sphere

5 An Example Session

The above described method has been implemented in a prototype under IRIX 5.3 on Silicon Graphics Workstations using the Open Inventor Toolkit [16] for three-dimensional graphics and C++ as implementation language.

5.1 Exploration with the Search Tree

The configuration file from example 1 allows the dialog as displayed in figure 10. By means of a context menu of the empty root label the user has selected the class `Protein_keyword` which results in marking the root label with this class.

Figure 10: Example Dialog with the Search Tree

Now, the context menu for the root label shows the titles of the matching queries, in this example the query "All protein keywords". If the user selects this query, the corresponding query to the database is executed and the resulting items are displayed in a cone tree. In the example, the user has picked one of the labels marked with the protein keyword "ALTERNATIVE SPLICING" and started the query with the title "All proteins for a given keyword". The resulting proteins are displayed in a new subtree. For a protein, a molecule viewer can be started.

5.2 Working with the Relevance Sphere

A second example displays how the relevance sphere can be used to show up similarity relations between objects. The excerpt from the configuration file in example 2 shows the description of a query which retrieves similarity relations between ligands.

```
query Ligand from Ligand
  title "Ligands similar to this ligand"
  relevance
    result (STRING,REAL)
    access "ligand.name,
           model.bonds->SubStructureMatch(thisModel.bonds)"
  from "ligand IN %DB::Ligand, model IN ligand.models,
       thisLigand IN %DB::Ligand,
       thisModel IN thisLigand.models"
  where "thisLigand.name == '%NAME'
        AND SubStructureMatch(thisModel.bonds) > 0.2"
```

Example 2

In the dialog displayed in figure 11 this query has been executed three times, with the result of every execution being displayed in a new subtree.

Figure 11: Retrieving Similarity Relations

Opening the sphere tool creates a dialog as displayed in figure 12. The reference ligands are displayed on the sphere surface, the similar ligands are situated in the sphere. Moving the reference spheres and changing their size gives further clues about similarity relations.

Figure 12: Display Similarity Relations with the Sphere

6 Conclusions and Outlook

The work presented within this paper extends the earlier published Lyberworld visualisation concepts and integrates them with the object oriented database system VODAK. In accordance with the requirements for a visualisation user interface an underlying information model is derived. Its implementation and use in an example application is demonstrated. On the basis of these results, one can state that Lyberworld has been enhanced significantly towards a system which is capable of integrating and visualising data from various heterogeneous sources. Although the integration of other data sources than VODAK databases has not been described in detail, the introduced information and query network layers enable developers to abstract from the formal restrictions and functionality of different data interfaces and the expressiveness of their modelling and query languages. One can easily add interfaces to other IR and database systems by following the introduced approach. In fact, interfaces to other database systems like, e.g., the relational database systems ADABAS-C and ADABAS-D of GMD-IPSI's industrial co-operation partner Software AG, are currently under development. Furthermore the "old" data interface to the INQUERY [4] document retrieval system is still maintained and implements already a second interface from the Lyberworld visualisation toolkit towards a state of the art IR system.

7 References

1. Aberer K, Fischer G. Object-Oriented Query Processing: The Impact of Methods on Language, Architecture and Optimization. GMD Technical Report No. 763, 1993
2. Batini C, Catarci T, Costabile M F, Levialdi S: Visual Strategies for Querying Databases. In: Proceedings of the IEEE Workshop on Visual Languages. Kobe, Japan, 1991
3. Boyle J, Fothergill, J., Gray, P.: Design of a 3D User Interface to a Database. In: Sawyer P (ed) Interfaces to Database Systems. Lancaster 1994
4. Callan J P, Croft W B, Harding S M: The INQUERY Retrieval System. In: Proceedings of the 3rd on Database and Expert Systems Applications. 1992.
5. Gemis M, Paredaens J, Thyssens I: A Visual Database Management Interface Based on GOOD. In: Interfaces to Database Systems. Glasgow 1992
6. Heilmann M, Hemmje M. Entwurf und Implementierung eines graphischen Visualisierungswerkzeuges für gerichtete Relationen auf der Basis der direktmanipulativen Relevanzkugel-Metapher. To appear in: GMD-Studien. St. Augustin, 1996
7. Hemm K, Aberer K, Hendlich M: Constituting a Receptor-Ligand Database from Quality-Enriched Data. In: Proceedings of the International Conference on Intelligent Systems in Molecular Biology. Cambridge, UK, 1995
8. Hemmje M, Kunkel C, Willet A. Lyberworld: A Visualisation User Interface Supporting Full Text Retrieval. In: Proceedings of ACM SIGIR '94. Dublin, 1994
9. Institut für integrierte Publikations- und Informationssysteme. VODAK V4.0 User Manual. GMD 1995
10. Mariani J, Benford S: Virtual Environments for Data Sharing and Visualisation - Populated Information Terrains. In: Sawyer P (ed) Interfaces to Database Systems. Lancaster, 1994
11. Massari A, Pavani S, Saladini S, Chrysanthis P.K: QBI Query By Icons. In: Proceedings of the International Conference ACM-SIGMOD. S.Jose, California, 1995
12. Pejtersen A M: A library system for information retrieval based on cognitive task analysis and supported by an icon based interface. In: Belkin N J, van Rijsbergen C J (eds) Proceedings of the 12th annual International Conference on Research and Development in Information Retrieval (SIGIR '89). Cambridge, Mass., 1989
13. Robertson G G, Card S K, Mackinlay J D. Information Visualisation using 3D Interactive Animation. Communications of the ACM 1993; 36(4): 56-71
14. Shneiderman B. Designing the User Interface, 2nd. Ed. Addison-Wesley, 1992
15. SGI Online Manuals. IRIS ViewKit Programmer's Guide.
16. Wernecke J. The Inventor Mentor, Programming Object-oriented 3D Graphics with Open Inventor, Release 2. Addison-Wesley, 1992