

Querying by Iconic User Interface on an Object-Oriented Database Desktop.

Armstrong Kadyamatimba and John Mariani

Computing Department, Lancaster University, Lancaster, LA1 4YR.

Tel: (01524) 65201 ext. 93783, Fax (01524) 493608 {arms,jam}@comp.lancs.ac.uk

1. Introduction

In (Kadyamatimba 1996) we described our basic concept of Desktop Objects for supporting browsing and direct manipulation schema evolution with the Oggetto OODB (Mariani 1992). Iconic User Interface (IUI) is an extension to provide a fully integrated querying service. Substantial work exists on graphical interfaces to databases: QBD* (Angelaccio 1990), OdeView (Agrawal 1990) and Moggetto (Sawyer 1995). Most of these interfaces support the browsing of schema of databases. However, OdeView extensions (Dar 1995) support browsing and querying but as separate operations. Querying operations should be integrated with browsing mechanism as in PESTO (Carey 1996). However, the difference with our work is that we exploit the desktop's direct manipulation to integrate the querying operations and results browsing.

The focus of IUI is not on the statement and formulation of the query, although this is certainly a major part of any future work on the system. Rather, once queries have been stated (potentially by expert users), our concern is that they are easily available and useful to even novice users, and that their appearance and presentation of results are potentially indistinguishable from the browsing process supported by our desktop. This is a major diversion from some of the systems mentioned above. The next sections describes the IUI.

2. Browsing the type lattice and instances

The “root” window shows all its subtypes and “instances” folder (figure 1). We can continue browsing the schema of the database by selecting the icons which represent types and open them to view their structure (figure 2). Cluttering of objects is controlled by displaying only two instances of a type and the excess instances are automatically put into the “instances” folder. Extra instances are displayed in multiples of ten (figure 3). The values of instances display (two and ten) are subject to “user profile” information and can be tailored by the users.

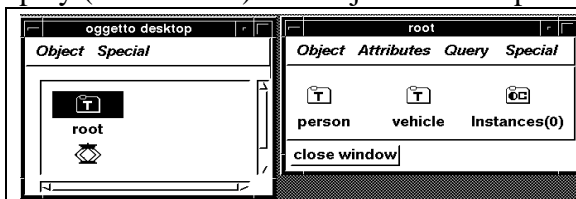


Figure 1 : The Initial Desktop and open “root” window

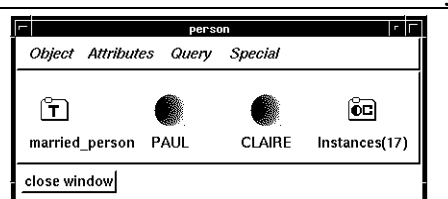


Figure 2 type “person” showing subtype and two instances plus instances

3. Querying the desktop

When a user selects a query folder, the associated query will be executed, and the folder opened into a window containing the results of the query. (see figure 4, 5 and 6 for querying mechanism illustration). By employing a specialist where the query statement is complex, end-users have a simple mechanism for calling up the results of a complex query. If we click on the query folder “old_age”, a window (figure 6) pops up containing all the object instances of person type in an iconic form which satisfy the specified query.

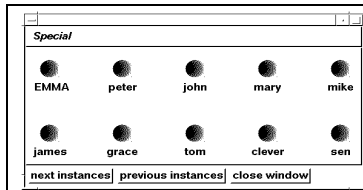


Figure 3: extra instances of type "person"

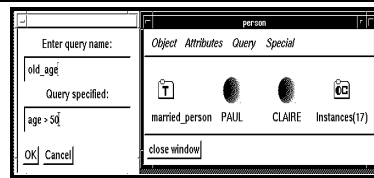


Figure 4 Create query prompt

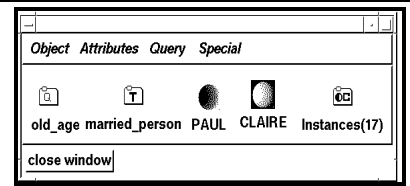


Figure 5 window "person" showing subtype, instances and query folder

The user can further manipulate the resulted object instances by opening them to view their attributes. Direct manipulation of the query folders allows a simple visual programming language to be developed which provides for the specification of union and intersection of query results. This allow users to build quite complex queries from seemingly simple visual building blocks (see figure 7).

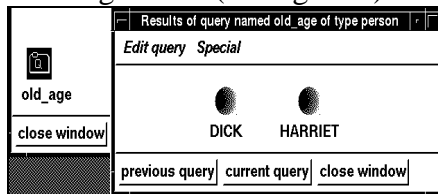


Figure 6: Query results of query "age>50"

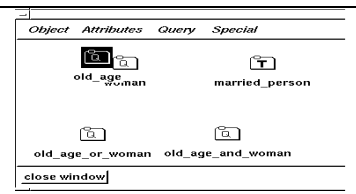
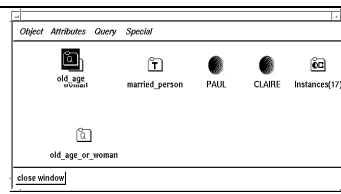


Figure 7: Creating new queries by direct manipulation

4 Conclusion

We have presented the UI, an iconic user interface which can be used to support dynamic query construction by creating query folders and view the query results iconically or textually. The query folders created provides an effective filtering mechanisms for selecting only the needed objects of interest and further browse them. We have described and shown how query folders can be used as building blocks for complex query construction. The interface cleanly integrates the browsing mechanism of the type and structural lattice with the querying operations.

References

- [Agrawal 90] Agrawal, R., Gehani,N., Srinisan,J. "OdeView: The Graphical Interface to Ode." *Acm sigmod International Conference on Mgt of Data*, Editor: H.Gracia-Molina and H. Jagadish, ACM Press, Pages: 34-43.
- [Angelaccio 90] Angelaccio, C., Catarci T, Snatch. "QBD*: A fully visual query system." *Journal of Visual Languages and Computing Vol.:* 1 Pages: 255-273.
- [Carey 96] Carey, M., Laura Haas, Vivek Maganty, John Williams. "PESTO: An integrated Query/Browser for Object Databases." *VLDB*, Bombay India, VLDB, Vol.: 22.
- [Dar 95] Dar, S., . N H Gehani, H V Jagadish, J Srinivasan. "Queries in an Object-Oriented Graphical Interface." *Visual Languages and Computing Vol.:* 6 No.: 1, Pages: 27-52.
- [Kadyamatimba 96] Kadyamatimba, A., John Mariani, Peter Sawyer. "Desktop Objects: Directly Manipulating Data and Meta Data." *User-Interfaces to Databases Systems*, Edinburgh Scotland, Editor: Jessie Kennedy and Peter Barclay.
- [Mariani 92] Mariani, J.A. "Oggetto : An Object Oriented Database Layered on a Triple Store." *The Computer Journal Vol.:* 35 No.: 2.
- [Sawyer 95] Sawyer, P., Sommerville, I., Colebourne, A., Mariani, J.. "Interactive Database Objects." *IFIP 2.6 Third Working Conference on Visual Database Systems*.