Describing and Characterising Visualisations

POSITION PAPER

Jessie B Kennedy, Kenneth J Mitchell, Peter J Barclay

Computer Studies Department, Napier University
Canal Court, 42 Craiglockhart Avenue, Edinburgh EH14 1LT, Scotland, UK
e-mail: <jessie,kenny,pete>@dcs.napier.ac.uk
phone: +44 0131-455 5340 ; fax: +44 0131-455 5394

1. Introduction

Although models exist for user-interfaces in general and are applicable to interfaces to database systems (IDS), it has been shown that there is a requirement for models which address the particular needs of databases. A generic framework for describing and specifying interfaces to databases has been proposed [1]. Currently this framework is being used as a model for the development of an environment for the construction of user interfaces to object oriented databases, DRIVE [2].

This paper reviews this framework and extends some of its components to permit a comprehensive categorisation of visualisations. It should be noted that interfaces to databases are more than visualisations and the framework deals with interfaces in general not simply visualisation. The paper will limit discussion to visualisation and the issues we consider important in the design of visualisations. The example visualisations categorised under the framework will be limited to 3D visualisations or non-immersive virtual reality.

Section 2 gives an overview of the IDS (user-interface to database) framework; section 3 extends particular components of the framework and discusses the categorisation of interfaces and section 4 highlights issues for discussion at the workshop.

2. IDS Framework

The framework that provides the foundation for IDSs is based on Abowd and Beale's interaction framework, which identifies four major components of an interactive system, i.e. *system, input, output* and *user* [3]. In applying this framework specifically to IDSs, modifications and extensions to these components have been made. An IDS has been defined to be the composition of *database, interaction, visualisation* and *user* components with the common features of each of these IDS components being further identified.

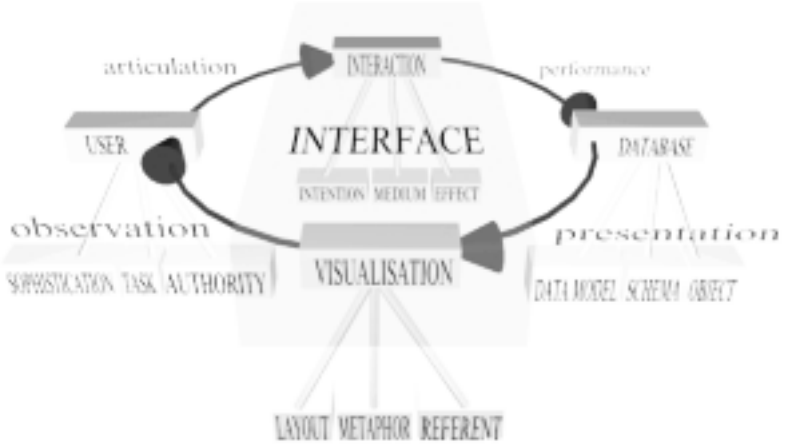

Figure 1 A framework for the characteristic components of user-interfaces to databases

2.1 Database

Depending on the purpose of the IDS, the database component may relate to the data model, the schema described under the data model, the objects instantiated from classes in the schema or some combination of these.

Data Model: The set of data modelling constructs present in any particular data model determine the flexibility, behaviour, expressiveness and level of integrity possible in the data described. Visualising the data model supported by the database can help users to understand precisely the semantics of the schema [4].

Schema: A database schema precisely defines the structure and meaning of the data in a database, in terms of the data model. Visualising the schema may be advantageous for new users to the database to use as a 'map' of the database or may be useful to a database administrator to visualise and update the schema.

Objects: Objects are the basic units of information in a database and it is important to provide appropriate interfaces to this information. Elementary objects might hold one atomic item of information, whereas complex objects typically have a variety of properties, operations, constraints and related objects. It may also be appropriate to visualise transformations or aggregations on the data.

2.2 User

The major characteristic features of a user affecting the IDS are his/her sophistication, task, and authority.

Sophistication: The user's sophistication is defined in terms of his/her knowledge of the components of the framework, i.e. the objects, schema or data model of the database, the style of interaction and the meaning of the visualisation.

Task: The tasks which users must perform are pertinent to interface design. In general terms these include querying , browsing, data entry and manipulation.

Authority: A user's authority is defined by the level of access to data provided by the interface, e.g. the permission to read, write or update this information. An end-user may only merit a limited view of the schema/data, whereas administrators normally require a full view.

2.3 Interface

In this framework, an interface is composed of visualisation and interaction components. A visualisation component is preferred to the more general output component in deference to the extensive body of research on data and database [5,6,7,8] visualisation.

2.3.1 Visualisation

A visualisation component refers to the output of graphical information particular to the current database application. The user's observation of and the database's presentation of this visualisation component is determined by the chosen layout and metaphor.

Referent: The purpose of a visualisation is simply to communicate to the user some component of the database user interface, i.e. the referent which the current visualisation is representing. Typically, this concerns the visualisation of the database's contents, from concrete visualisation of database objects [9] to abstract visualisation of schemata or queries [10], or even users[6].

Metaphor: A metaphor defines the symbol used to represent the component being visualised. A metaphor may range from the direct representation of the component, to an abstract symbol in some way related to the component in question. Formal definition of the mapping from the referent to the metaphor may be possible [11,4].

Layout: A layout is defined as the position of interface components relative to other components in a common environment. This is particularly important in communicating the structure of the data to the user. Clearly, the structure of the schema in the database component will frequently determine the designer's choice of visual layout.

2.3.2 Interaction

An interaction component refers to the input of information articulating the user's intention to the interface. This intention is communicated through the interaction component's medium and performed to achieve a specified effect.

Intention: The intention of an interaction component is the specific interface action which satisfies some goal. This concerns both the subject of the interaction component and its practical function. The subject will be some selection of database, user and interface components and its use may concern the entry, manipulation, browsing, or querying of data.

Medium: The medium through which a user's intention in an interaction is communicated concerns both physical and logical aspects of the interface. Physical aspects include the mouse or keyboard and logical aspects, the buttons, scroll bars, menus, text fields, dialog boxes and objects. The logical aspects of an interaction component's medium is directly associated with a visual component's referent in the interface.

Effect: The effect of an interaction component can change any aspect of any component in the IDS framework. For example, the effect of a button whose intention is to delete an object in the database will (obviously) remove an object from the database.

3. Categorisation of Visualisations under IDS Framework

The framework for IDSs described above covers not only the categorisation of the components of the interface which can be used for classifying interfaces, but also the relationships between these components in the interface. This information is necessary when building and modelling IDSs but is less relevant for simply classifying interfaces or visualisations. For example, some components play a limited part in categorising visualisations, e.g. the *effect*, which specifies the behaviour of an action in the interface. Other components can be usefully specified in more detail to aid in classifying the interface. The role of the components in classifying or categorising interfaces, with particular reference to visualisations, is expanded upon below, with example visualisations given to illustrate particular features of specific IDSs or information visualisation tools.

- < Database
  - < Data Model
  - < Schema
  - < Objects (raw data, transformed)
- < User
  - < Sophistication
  - < Task (update, browse, query)
  - < Authority
- < Interface
  - < Visualisation
    - < Referent
    - < Metaphor (mutable, algorithmic)
    - < Layout (2D/3D, mutable, animated)
  - < Interaction
    - < Intention
    - < Medium (direct manipulation, navigable)
    - < Effect

Figure 2 IDS framework with classifications of sub-components with respect to information visualisation

3.1 Database

The database component can be used in the classification to define the range of information a user can visualise with a given tool, e.g. the schema, the raw data, complex transformations on the data. Any database component may be the referent of the visualisation. OPOSSUM [4] is an example of a system supporting visualisation of schema information only, while FilmFinder [13] is a visualisation of mainly object information.

3.2 User

In the user component of the framework, the feature of particular relevance to the classification of visualisations is the task component. Database tasks can be classified into query, browse, update (add, delete, modify), transform. Figure 3 shows an interface with visualisation which the user can use for browsing the database, but is unsuitable for specifying complex queries or updating the data. Additionally the user may be the *referent* of the visualisation as in QPIT [6].

3.3 Interface

The interface component of the framework is of greatest relevance to the classification of visualisations. In the visualisation component of the interface, the *referent* is related to what the user can visualise. The *metaphor* can be further classified on properties such as 2D or 3D; whether it is fixed or mutable, i.e. can the metaphor be altered interactively by the user; and whether it is still or animated. Figure 4 shows an interface which supports both 2D and 3D metaphors.

Figure 3 Browsing on-line timetable information using an employee database interface

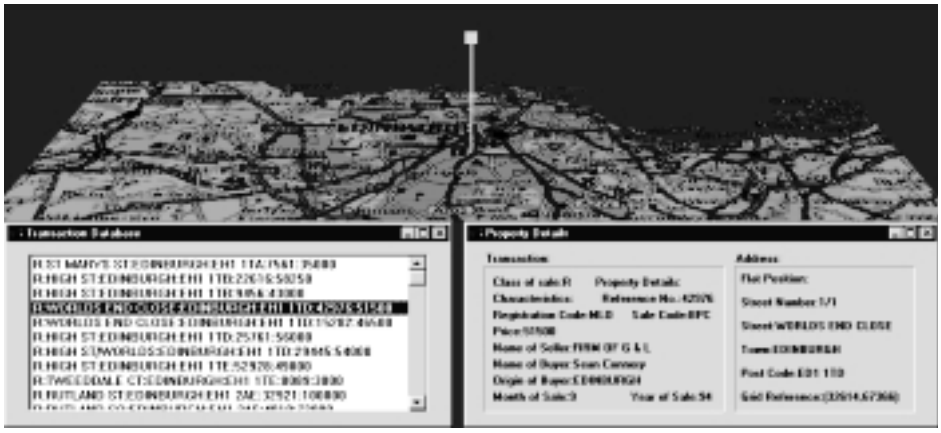

Figure 4 Combined 2D & 3D metaphors for a property transactions database

The *layout* for visualisations can be classified as being fixed, mutable or algorithmic. Figure 5 gives an example of an interface where the layout has been generated by a helical plotting algorithm. The algorithm plots sequences of data items in a helix, which may be fine tuned with parameters such as, the radius, period and vertical velocity. The layout generation mechanisms may give more or less control to the user, for example, the user may be able to choose a particular style of visualisation or may be able to specify the attributes of the information to be used in determining the layout. Types of visual layouts include linear, circular [5], form, grid, hierarchical [13], scatter plot and graph [8] structures.

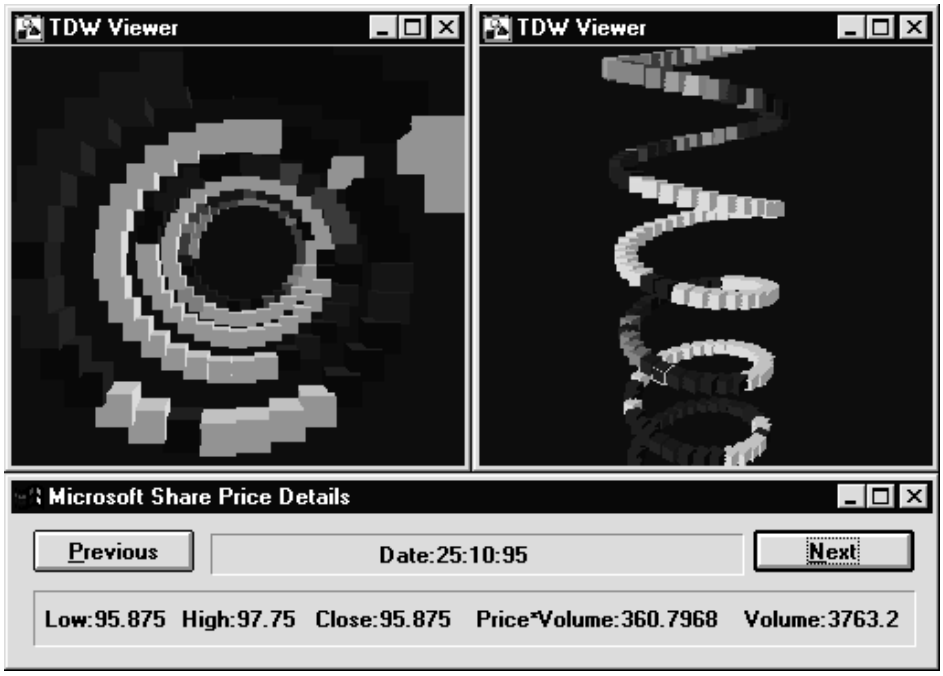

Figure 5 Helical layout generation for time series stock market data

The interaction component of the interface could also be used in the classification of IDSs, however only the *medium* sub-component is of specific relevance to the classification of visualisations. The medium is related to the metaphor in use and will clearly have an effect on the visualisation. For example, the use of buttons versus direct manipulation of objects will have a bearing on the visualisation. Navigation is of particular interest in 3D visualisations, where relative and absolute movements may be supported.

Table 1 summarises the classification of figures 3-5 based on a subset of the components of the framework. Even such a high level checklist can highlight differences in the three visualisations offered. More detailed checklists for each individual interface element will clearly result in greater understanding of the visualisation system in question and provide a sound basis for comparison with systems.

IDS Framework Sub-Component	Classification	Fig. 3	Fig. 4	Fig. 5
Data Object	Raw Data	✓	✓	✓
	Transformed	×	×	✓
User Task	Update	✓	×	×
	Browse	✓	✓	✓
	Query	(✓)	×	×
Visualisation Layout	Mutable	×	×	✓
	Algorithmic	×	×	✓
Visualisation Metaphor	2D/3D	Both	Both	Both
	Mutable	×	×	×
	Animated	×	×	×
Interaction Medium	Direct Manipulation	✓	✓	✓
	Navigable	✓	×	✓

Table 1 Classification of example IDSs

4. Discussion

We have shown how the IDS framework can be used as a basis for categorising information visualisation systems. During the workshop we would like to discuss the use of this framework as a mechanism for the classification of visualisations and consider the classification of the participants' 'favourite interface' according to the classification. Further we would like to discuss the use of the framework for the classification of interfaces to databases more generally. The detailed sub-components or features described could be explored and additional features suggested if thought appropriate or necessary.

5. References

1. K Mitchell, J Kennedy & P Barclay (1996) A Framework for User Interfaces to Databases, *Proceedings of AVI'96 International Workshop on Advanced Visual Interfaces*, Gubbio, Italy, May 1996

2. K Mitchell & J Kennedy (1996) DRIVE: An Environment for the Organised Construction of User-Interfaces to Databases, *Proceedings of IDS3, the 3rd International Workshop on Interfaces to Databases*, Napier University, Edinburgh, July 1996

3. G.D. Abowd & R. Beale (1991) Users, systems and interfaces: A unifying framework for interaction, *HCI'91: People and Computers*, 4, 73-87.

4. E.M. Haber, Y.E. Ioannidis & M. Livny (1994) Foundations of Visual Metaphors for Schema Display, *Journal of Intelligent Information Systems*, 3.

5. M.H. Rapley & J B Kennedy (1994) Three dimensional interface for an object oriented database, *2nd International Workshop on Interfaces to Databases*, 2:8, 133-158.

6. S. Benford & J. Mariani (1994) Populated Information Terrains, *2nd International Workshop on Interfaces to Databases*, 2:9, 159-169.

7. B.M. Collins (1992) Data Visualisation - Has it all been seen before? *International State-of-the-Art Conference on Animation and Scientific Visualisation,* 1-33.

8. J. Boyle, J.E. Fothergill & P.M.D. Gray (1994) Amaze: a three dimensional graphical user interface for an object oriented database, *2nd International Workshop on Interfaces to Databases*, 2:7,117-131.

9. K.J. Mitchell, J.B. Kennedy & P.J. Barclay (1995) Using a Conceptual Language to Describe a Database and its Interface, *British National Conference on Databases*, 13:7, 101-119.

10. M.M. Zloof (1975) Query by Example, *Proceedings of the National Computer Conference*, 431-437.

11. T. Catarci, M.F. Costabile, A Massari, & G. Santucci (1994) Database Interaction: 3D or not 3D, *Proceedings of FADIVA Workshop*, 1,8.

12. N Jog & B Shneiderman (1995) Starfield Information Visualisation with Interactive Smooth Zooming, *Proceedings of VDB-3 Visual Database Systems*, Lausanne, Switzerland, March 27-29 1995, Ecole Polytechnique Federale Lausanne.

13. F. Steinfath, K. Bohm, B. Lange (1994) Evaluation of Complex Information Processing Systems in 3D-Space, *FADIVA Workshop*, 1:2.