SURF: A semantic update and retrieval facility

by FRED MARYANSKI and DARRELL STOCK
University of Connecticut
Storrs, Connecticut

ABSTRACT

The definition and design of a query language based on a semantic data model and targeted for personal workstations with color graphics is presented. The language, SURF, contains a browsing facility which permits the user to learn the structure of the database schema by exploring diagrams used in the database design phase. By using the browser, the user can create forms upon which the database operations are expressed. SURF queries are formulated by traversing diagrams and filling forms thus minimizing the keyboard input required of the user. Color is also utilized in the composition of complex Boolean expressions. The integration of a graphical interface with a semantic data model is intended to simplify database access for the nonprogramming workstation user.
INTRODUCTION

Motivation and Problem Solution Overview

The nature of interfaces to database systems is fundamentally dependent upon the functionality of the computing system and the conceptual model offered by the database package. Many of the current generation of database query languages provided keyboard directed interfaces to a tabular relational model. The shortcoming of conceptual models such as the relational model is that they present a limited set of descriptive options to the end user. The user is forced to map a mental image of the problem space into a collection of flat tables. Recently, a number of researchers have embarked on studies of a new generation of semantically rich conceptual data models which would permit the description of data in a manner closer to the user's perception. This class of data models is generally known as semantic data models.1, 2

Concurrent with the evolution of semantic data models is the growth of personal workstations from purely ASCII character oriented devices to systems offering a variety of pointing devices, powerful graphical capabilities, and bit map color displays. It seems clear that the next generation of database query facilities must exploit the functionality of modern personal workstations and semantic data models to provide the end user with a rich, yet simple, interface to databases.3, 4, 5

The above observations have led to the definition and implementation of the SURF query facility.6 SURF is based upon the semantic data model defined by Peckham7 as part of the Data Model Compiler project at the University of Connecticut.8 In order for a personal workstation to support SURF, it must provide bit map graphics with color and a mouse. The initial implementation of SURF utilized a Unix workstation which offers extensive graphical facilities beyond those required by the query facility. The software for the DMC project of which SURF is a component is presently being moved to a less powerful, Unix-based, personal workstation. While SURF is implemented in C under Unix, the choice of both the language and the operating system are independent of the basic principles of the query facility.

Design Objectives

The design of SURF was driven by the following parameters in order to produce a conceptually pleasing database access facility.

1. The ability to tailor the conceptual data model to the mind set of the end user:

SURF is designed upon a semantic data model so that the end user may interact with the database using the concepts of his/her own discipline.

2. The provision of a graphical interface for query formulation:

Positional information and screen context are exploited in order to simplify the task of the end user.

3. The ability to determine the structure of the database in a straightforward manner:

A browsing facility supports the user's need to examine high and low-level details of the schema.

4. Automation of the details of integrity and consistency maintenance:

Semantic integrity constraints are enforced to avoid inadvertent creation of inconsistent results.

QUERY LANGUAGE CLASSIFICATION

In the classification scheme of Lochovsky and Tsichritzis,9 query languages are categorized as keyword, by-example, natural language, graphic, or multimedia. SURF combines features of by-example and graphic languages and is most closely related to OBE/OBE10 and LID.11

THE SEMANTIC DATA MODEL

The data model employed by SURF is an extended entity-relationship model with a subtype/supertype inheritance structure for entities which are defined in terms of their properties, operations, and constraints.7 A graphical application design tool, DBDT,12 assists the designer in the specification of the entities, relationships, operations, and constraints of a given application domain. SURF's role is to map queries expressed using a combination of tables and DBDT diagrams into transactions against the semantic database.

Relationships

In general, semantic data models2 can be distinguished by their built-in relationships. The data model employed by SURF7 directly supports four types of relationships: IS-A, reference, nest, and association. The IS-A relationship is utilized to express generalization/specialization among entity types. The form of IS-A employed here is a template-oriented inheritance mechanism which is very strict in terms of inheritance of properties but does provide for overriding defaults and refining constraints at the subtypes.

The latter three fundamental relationship types are the basis for all user-defined relationships. They have the func-
1. Reference—a mapping from one entity to another. A reference is realized as an attribute of the referencing entity.

   **Example**—a STUDENT entity references a PROFESSOR entity through the attribute ADVISOR which is of type PROFESSOR.

2. Nest—a mapping from one entity, the Nest Owner, to a set of entities, the Nest Members. A nest is a set-valued attribute of the owner.

   **Example**—a STUDENT entity contains a nest of the COURSE entities as its COURSE_REQUEST attribute.

3. Association—a many-to-many mapping between two or more entity types. Associations are represented as independent components of the model with their own properties, operations, and constraints.

   **Example**—TAKE is an association between the STUDENT and COURSE entities that has GRADE as an attribute.

**Semantic Integrity**

Rules are included to preserve the semantic integrity and consistency of the database. Integrity constraints must be evaluated each time an operation is executed. This involves determining which (if any) relationships an entity participates in and then evaluating the appropriate rules for those relationships over the specified operation. The following constraints are automatically enforced by SURF.

1. If an object is to be inserted as a nest member, that member must exist as an entity object.
2. An object instance cannot be deleted if it is being used as a nest member.
3. The deletion of a nest owner instance implies the deletion of all nest objects owned by it.
4. If an object instance is to be inserted as a reference that reference instance must exist as an entity object.
5. An object instance cannot be deleted if it is being used as a reference.
6. The deletion of a referencing instance implies the deletion of all reference objects owned by it.
7. An object instance cannot be deleted if it is being used as an association participant.
8. All association participant instances must exist as entity objects before an association instance can be inserted.
9. Insertion of a subtype instance implies the insertion of one instance of each related supertype.
10. Deletion of a supertype instance implies the deletion of the corresponding subtype instances related to the supertype instance.

**THE SURF QUERY FACILITY**

A high-level view of the overall semantic database system is illustrated in Figure 1. A database designer (database administrator) applies the DBDT tool to specify the structure of the semantic database. SURF is driven by the metadata generated by DBDT. The major elements of SURF are:
1. A browser program which permits database structure analysis
2. A retrieval/update processor which supplies the resources to interact with the semantic database and the SURF user

Browsing a Semantic Database

When the design phase is complete, the nonprogramming user can activate the Browser which provides a visual representation of the database structure. A SURF browsing session consists of selection, navigation, and tailoring.

Selection

In the SURF environment, a semantic database consists of a collection of diagrams. When a session is initiated, the user is presented with lists of the diagrams of the database as in Figure 2. Note that isolated entities are those which have not been included in any diagrams. The mouse icon at the bottom of the screen provides the user with a reminder of the actions associated with each button. Figure 3 illustrates the contents of the screen following the selection of an IS-A diagram.

Navigation

The navigation facility permits the user to examine the detail of database objects while retaining an overall perspective of the schema. When the user selects an object, a window describing one level of detail appears. Successive levels can be obtained by selecting complex attributes of the selected entities as in Figure 4. In the navigation facility, the system indicates the relationship among entities and attributes by lining up the top portion of a window with the entity from which it is generated. Shading is employed to indicate the most recently activated object.

Tailoring

Tailoring involves selecting the attributes of the entities to appear on the query forms. If no tailoring is performed, all attributes of the selected entities will be available at query formulation time. Tailoring is mouse-selectable and is visualized by changing the color (highlighting) of database objects and attributes from gray to orange.

The Retrieval/Update Processor

Forms are used to represent the entities and relationships of the semantic data model. Query transactions are expressed by form filling and mouse maneuvering. Figure 5 presents an example of the screen immediately after browsing and before transaction formulation. The steps in the specification of a query are:

1. Select the operation (RETRIEVE, INSERT, DELETE, MODIFY)
2. Specify the selection criteria (or fill in data on insertion) on the forms
3. Select the attributes for output (default is output all)

Figure 6 expresses the query “Retrieve the names of all students with GPA greater than 3.8.” The output of a SURF retrieval appears in tabular form, listing values for mouse-selected attributes.

One of the more difficult problems facing developers of query languages is the expression of disjunctive and conjunctive conditions. This is greatly simplified in SURF, and you can experiment with several different forms of queries to see how the system operates.
tive queries. The typical end user who has not been schooled in Boolean or Aristotelian logic may tend to use “AND” and “OR” interchangeably in the verbal expression of queries. SURF attempts to solve this dilemma through the use of colors. Using the function keys, terms of the selection expression can be entered in any of four colors. A boolean AND is undertaken on entries appearing in the same color. A boolean OR is undertaken on entries appearing in different colors. The use of colors provides the user with a straightforward visual representation of complex queries.

Insertions involve filling in attribute values on the forms identified in the browsing process. Key attributes require values while non-key attributes will receive a null value if the user does not elect to supply information at that time. Null values may be later assigned values by using the Modify operation. Figure 7 illustrates the insertion of the ID, NAME, AGE, and ADVISOR for the student James. Null values will be stored in the GPA and CREDITS attributes.

SURF deletions are expressed by placing selection criteria next to the appropriate attributes. The same rules as for retrievals are followed. For example, Figure 8 presents the query which deletes all students whose advisor is Brown. The highlighting of the STUDENT entity indicated that students are to be deleted.

A modification operation entails first identifying the objects to be altered and then specifying the replacement values for the appropriate attributes. Attributes to receive new values are identified by selecting the attribute name which results in the name being shaded. If an attribute is to be used for both identification and value replacement, then entries are made on multiple lines within the field for that attribute. SURF permits multiple lines per attribute for the expression of this and more advanced operations. The rule for interpreting modification operations is that the value on the last line of a shaded attribute is used as the replacement value. This feature is illustrated in Figure 9 in which all students with grade point averages greater than 4.0 and with less than 60 credits, have their grade point average set to 4.0.

A design goal for SURF was to provide straightforward expression for the most common operations. As demonstrated above, the context provided by the diagrams and forms permits the statement of queries expected of the typical end user in a rather simple fashion. For a discussion of the handling of more complex operations, see the work by Stock.

### The Experienced SURF User

One of the more serious problems facing the designer of database query facility is that of user maturity. Features designed to guide the first time user through a query session can become annoying to the person who has worked with the language for six months. SURF addresses this issue in two ways:

1. **Macros**—The experienced user who has repetitive tasks to perform may define a sequence of screens as a macro and call this operation when required.
2. **Defaults**—The user who is familiar with the structure and contents of the database can shortcut the browsing process by merely selecting the entities from a diagram and then moving to the transaction menu without selecting specific attributes. As a default, the system will generate a form containing the attributes of the entity. At worst, the user will obtain a superset of the attributes required as the response to the query. However, it is a simple matter for an individual familiar with the data to visually select the attributes of interest from the screen.

### CONCLUSION

**Summary**

SURF is a database query facility for the new generation of personal workstations and semantic data models. By exploiting current workstation and database technology it offers end users straightforward access to data which is represented in a clear conceptual model. The semantic richness of the data model allows the representation of information in a manner that is close to that of the end user’s perception of the environment. Thus, the end user is not forced to learn the terminology and internal organization of the database system. Through the use of diagrams, forms, color, and a pointing device, a context is established for each operation. Therefore, the user enters simple selection expressions and data values in order to specify database operations. The end user does not use the keyboard to input keywords or any other form of descriptive text.
Status and Future Plans

The initial implementation of SURF has been completed recently. Plans are to exercise and enhance SURF as the Data Model Compiler project continues to investigate the production of semantically rich database management systems. All components of the overall system are being moved to a smaller, Unix-based personal workstation.

As personal workstations continue to grow in functionality and the quality of their interfaces increases, database query facilities will take advantage of these features to offer the end user more options for the expression of database operations. With similar advancements in the data modeling area, the possibility of productive database manipulation by the true casual user could be realized.

ACKNOWLEDGEMENT

The research was supported in part by a grant from the National Science Foundation, ECS-8401487.

REFERENCES
