Capturing Chemical Structure Information in a Relational Database System: The Chemical Software Component Approach

T. R. Hagadone* and M. W. Schulz

The Upjohn Company, Kalamazoo, Michigan 49001

Received May 1, 1995®

This paper proposes an approach to the design of chemical structure databases in which a chemical software component is integrated into an extensible general-purpose relational database system. It is argued that this approach can be used to create chemical databases that encompass the complete range of structure storage, retrieval, and analysis capabilities normally expected in a traditional chemical database system while taking advantage of the superior general-purpose database features provided by relational database systems. The design and implementation of a full-featured chemical structure software component and its integration into the Microsoft Access relational database system are discussed. User experiences with this approach at Upjohn are reviewed. The chemical structure component described in the paper is included as downloadable supporting information.

INTRODUCTION

Over the past two decades, chemical database management systems have evolved to meet the special storage and retrieval needs of chemical structures and reactions and have become important software tools for the chemical and pharmaceutical industries. Over the same period, relational database management systems have evolved to meet the requirements of a broad range of business, engineering, and scientific data management applications and are now employed in nearly all enterprises, chemistry-related and nonchemistry-related. In spite of their separate heritages, modern chemical database systems and relational database systems offer a similar set of features which typically include the following: (1) a core database engine that defines a data model, data definition language, and data manipulation language; (2) a graphical user interface for designing and using database tables, data entry and retrieval forms, queries, and printed reports; and (3) a programming language and integrated development environment for the creation of custom applications to automate business and scientific tasks. Each type of database system has individual strengths. Chemical database systems' main advantage lies in their unique ability to effectively manage chemical structure information. Relational systems, on the other hand, because of their large user communities, competitive market environment, and substantial development organizations provide a superior set of general database management capabilities at a relatively low cost.

In view of the considerable similarity between chemical and relational database systems the question naturally arises as to how a relational system might be adapted to include chemical structure information and associated retrieval capabilities. In a previous paper[1] that explored this question, it was found that while the relational database systems of the time were able to support chemical structure storage and retrieval, the user interface components of the systems did not provide an adequate environment for the entry, display, and searching of structural information. Recently however, the technology of extensible software systems and reusable software components has advanced significantly and software "objects" created by one application program are now routinely embedded in the "documents" of a second program in a fashion that maintains a link to the originating application. One example of this capability is the way in which structure diagrams created with structure drawing programs are now routinely inserted into word processor documents and later reactivated for editing by simply double-clicking on the diagram as it appears in the word processing document. Chemical add-ins for spreadsheet programs make use of this extensibility technology as well.[2] In a similar fashion, some relational database systems now provide support for database fields whose content and semantics are defined by separate software components.

This paper will attempt to demonstrate that the extensibility features available in today's relational systems now provide an adequate foundation for the design of packaged chemical structure software components that can be used to fully integrate chemical structure information into a relational database. We will show that the combination of a relational system and an add-in chemical component can be effectively employed by users to design and implement powerful and flexible custom chemical database applications.

To demonstrate the feasibility of this approach, a chemical structure software component derived from Upjohn's Cousin[1] chemical information system was developed and integrated into the Microsoft Access relational database system for Windows. Microsoft Access (referred to as Access for the remainder of the paper) is a popular and full-featured database system that supports databases ranging from small and simple to large and complex, along with their associated queries, forms, reports and programmed applications. The chemical software component, called the Cousin add-in for Microsoft Access, augments Access with the structure storage and retrieval functions normally expected in a chemical database system. The Cousin add-in for Access is currently being used within Upjohn for the creation and management of personal and project-oriented chemical databases. Figures 1 and 2 show sample database forms typical of the type that are being created by users at Upjohn.

The paper begins with a review of the special chemical-structure-related capabilities provided in a chemical database

® Abstract published in *Advance ACS Abstracts,* August 15, 1995.

0095-2338/95/1635-0879$09.00/0

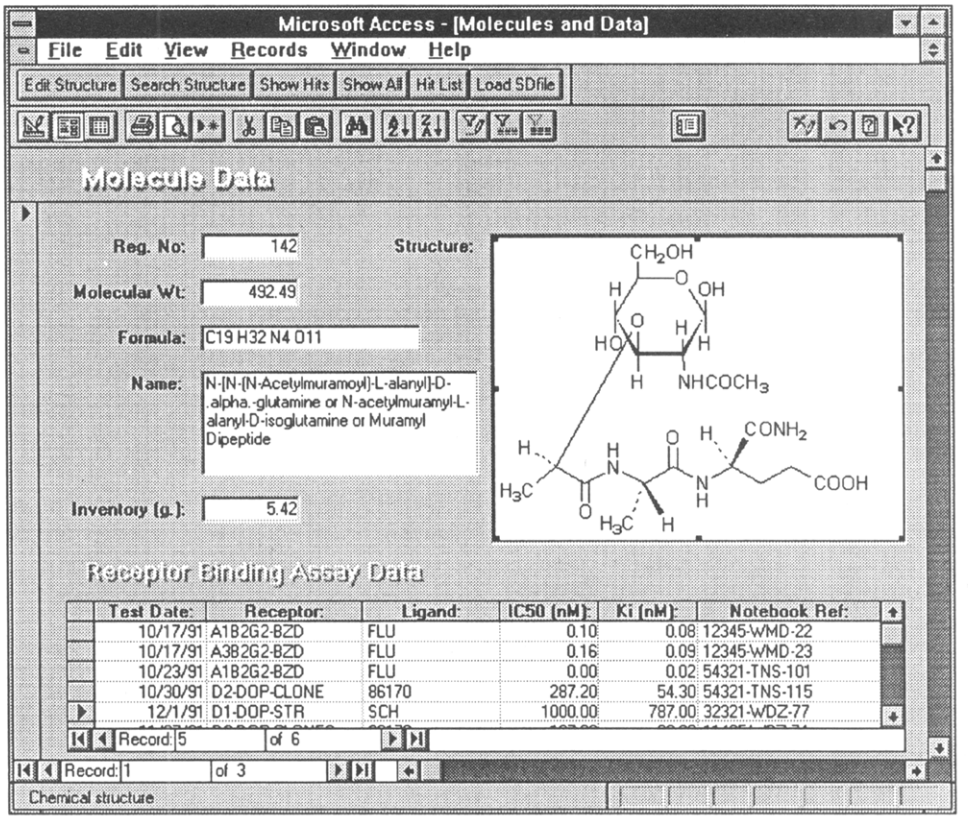

Figure 1. Sample Microsoft Access form displaying structural information and associated biological assay data from a second table.

system and then describes the design of the Cousin add-in and how it uses the extensibility features of the Access database system to integrate these capabilities. We then describe how the combination of Access and the Cousin add-in can be used to create a variety of types of chemical applications and conclude with a discussion of performance limitations and their possible solutions.

Although in this paper we use Cousin and Access to demonstrate the feasibility of the chemical software component approach, it is our position that the technique is general in nature and that other chemical software components can be developed for use with Access and other suitably extensible relational database systems. The full add-in described here is being included as downloadable supporting information to the paper to allow interested readers with a PC and Microsoft Access to evaluate for themselves the effectiveness of the chemical software component approach.

CHEMICAL SOFTWARE COMPONENT DESIGN

As discussed in the introduction, chemical and relational database systems share a common set of general database management features. In addition to these common features, a chemical database system is expected to provide special capabilities for storing and retrieving structural information as outlined below.

·Structure Data Model. The data model defines the elements of the chemical structures or reactions to be stored in the database and the way that the elements interrelate. The model is typically based on a connection table paradigm and includes atoms and bonds and their attributes as well as other chemical building blocks and a method of specifying the roles of, and relationships between, the various constituent parts.

·Structure Database Field Data Type. A structure field data type is provided as a means to define structure-containing fields in a database. It is an addition to the usual database field types such as number and text.

·Structure Form Control. Forms, such as those shown in Figures 1 and 2, are used to define the way that data will be presented graphically. Controls (boxes) on forms display the values of individual database fields and can be sized and positioned on the form at form design time by the user. A structure control displays structural information from an underlying structure database field and is an addition to the usual set of controls such as number and text.

·Structure Drawing. A method must be furnished for entering and editing the chemical structures and reactions stored in the database.

·Structure Searching. A chemical database system provides a method for the user to define structure search queries including full structure, substructure, and similar structure queries, a search engine to execute the searches, and a way of displaying the results.

·Miscellaneous Functions. These functions can include a variety of structure-related capabilities such as the following: a method for automatically updating molecular weight and formula fields in a database; a way of loading and unloading structural data from files; techniques for managing "hit lists" resulting from structure searches; and programmatic methods for manipulating the individual atoms, bonds, and other elements of a structure.

In the approach described here, each of the above items is provided either by Access or by the Cousin add-in to supply the full functionality normally expected in a chemical database system. Access supplies a method for defining

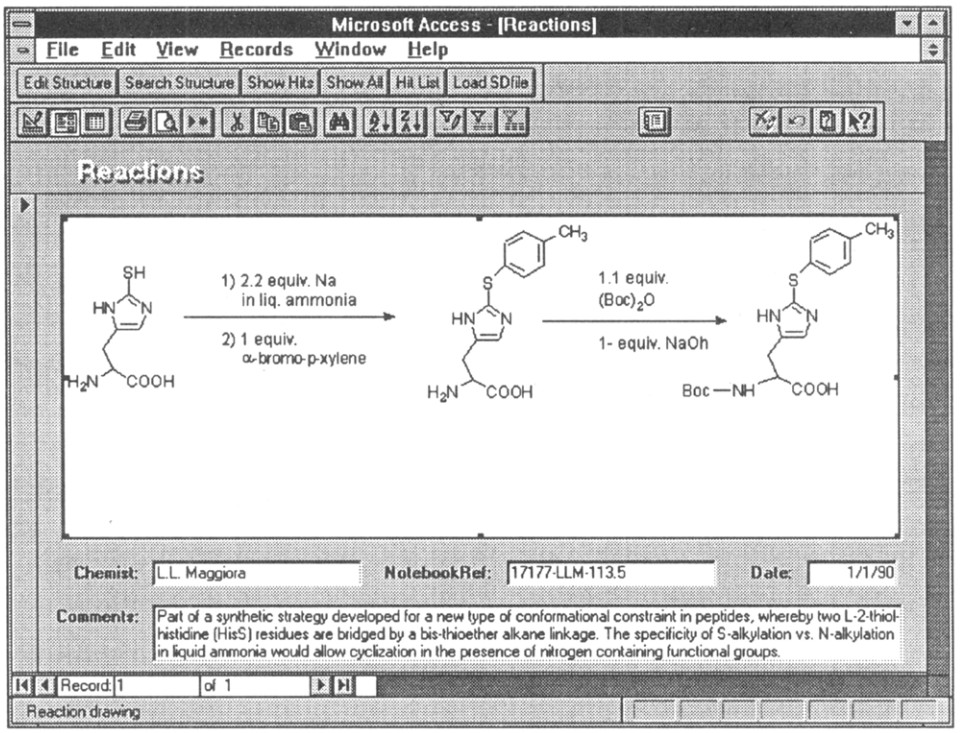

Figure 2. Sample Microsoft Access form for a reaction database.

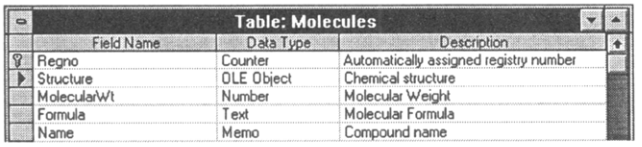

Table: Molecules

Field Name	Data Type	Description
Regno	Counter	Automatically assigned registry number
Structure	OLE Object	Chemical structure
MolecularWt	Number	Molecular Weight
Formula	Text	Molecular Formula
Name	Memo	Compound name

Figure 3. Adding a structure field to a Microsoft Access table.

structure fields in a database table and a complementary form control for structure display. The other items are furnished by the Cousin add-in with Access providing the "hooks" that allow these functions to be cleanly integrated. Below, we discuss the capabilities supplied by Access, those provided by the Cousin add-in, and the means by which the two are integrated.

Access and OLE. The fundamental software technology that allows structures to be included in an Access database and displayed to users via forms is known as Object Linking and Embedding (OLE). OLE, which is one of several competing software component architectures,[3] is a standard feature of the Microsoft Windows environment. The OLE standard defines a set of software protocols that allow software "objects" created by one application program to be inserted into the "documents" of a second program while maintaining their original identity and content. In this case, an Access database is the container document for Cousin chemical structure objects.

Construction of a chemical database application begins with the creation of database tables. In addition to the usual field types, Access provides an "OLE object" data type that is used to define structure fields. A sample table design screen for a table containing a structure field and other common fields is shown in Figure 3. Following table design, the user creates one or more forms, such as those shown in Figures 1 and 2, that provide a view onto the underlying table(s). Structures are displayed in a rectangular form control box called a "bound object frame" that is associated with the underlying database table's structure-containing OLE object field. The bound object frame is sized and positioned on the form at design time by the user.

In addition to a bound object frame's size and position, Access maintains a list of other properties that can be modified by the user to affect the appearance and behavior of the structure control box. For example, the "Size Mode" property determines how a structure is scaled for display, and the "On Updated" property allows a programmed action to take place whenever a structure is updated. Properties can be exploited to provide intelligent chemical behavior; for example, the "On Updated" property can be set to call a Cousin add-in function that will automatically update optional molecular formula and weight fields in a record whenever the associated structure is modified.

The Add-In's Responsibilities. The Cousin add-in is responsible for providing the remainder of the special capabilities of a chemical database system including the specification of a structure data model, structure searching, editing methods, and miscellaneous functions. The add-in furnishes most of these capabilities to the user through the structure toolbar shown in Figure 4. Clicking on a toolbar button performs the associated operation. The first button, Edit Structure, invokes the molecule editor for the structure in the current record and provides a full set of structure drawing tools. Upon exit from the editor the structure in the current record is updated. The Search Structure button invokes the query definition dialog box shown in Figure 4. Full structure, substructure, and similar structure searches are supported. Queries can include R-groups and other special structural constraints as described previously.[4] Following query definition, the search is executed and the records containing matching structures are displayed using the current form. Structure searching speed is discussed later in the performance section of the paper. The Show Hits and Show All buttons are used to toggle the set of records displayed in the current form between the matches found in the most recent search and the full set of records in the

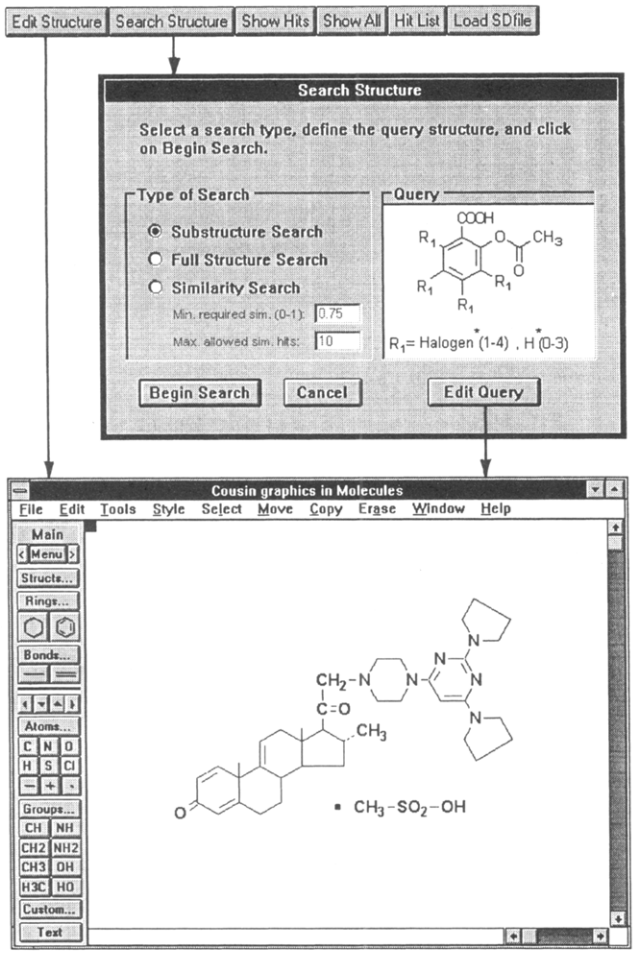

Figure 4. The structure toolbar and associated user interface dialogs.

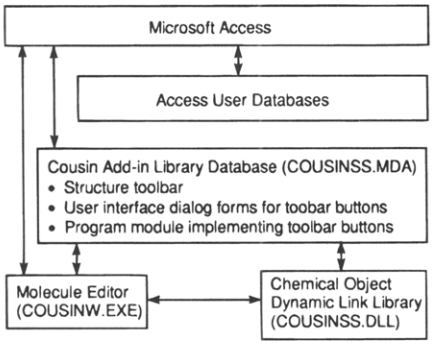

Figure 5. Control and information flow between Access and the Cousin add-in subcomponents.

underlying database table. The Hit List button provides dialog boxes that allow search result lists to be edited, saved to, and restored from files and logically combined together. The Load SDfile button displays a dialog box that allows structure-containing files in MDL Information Systems' SDfile format[5] to be loaded into an Access database.

Add-In Internal Design. The Cousin add-in is designed so that once it has been installed, it appears to be an integral part of Access and can be used with any Access database. The only change in Access from the user's point of view is the appearance of the structure toolbar. Below the surface of the interface, however, the add-in consists of three interrelated software subcomponents as is shown in Figure 5: an Access library database, the molecule editor, and a chemical object dynamic link library (DLL). An Access library database is a collection of toolbars, program modules, forms, and other database objects that act as extensions to Access. The Cousin add-in Access library database contains the structure toolbar, which is displayed automatically when Access starts, associated dialog box forms, and an Access Basic code module that implements the user's requests. The code module is responsible for orchestrating the interaction between the user and the other subcomponents of the add-in. Functions in the code module display and process dialog box forms, activate the molecule editor, and call chemical object DLL routines as they are needed to implement the structure toolbar functions. The molecule editor is invoked when the user wants to draw a database structure, a reaction, or a search query and can also be used as a stand-alone drawing program. The chemical object DLL routines are called to support a variety of lower-level program-accessible operations on chemical structure objects such as aromaticity and tautomerism perception, bit screen generation and matching, and atom by atom structure matching.

APPLICATIONS

Chemical structure databases can be classified along several dimensions. One dimension is defined by the point in a drug or chemical's life cycle where the database is built and used. It spans a time range beginning with the lead finding phase and continuing through development, production, and marketing. Another dimension is described by the size of the user community for a database and is usually approximately proportional to the number of structures in the database. This dimension ranges from relatively small personal databases to larger project, departmental, divisional, enterprise-wide, and commercial databases. A third dimension involves the type of structural information and associated data stored in a database. Structural information can be in the form of two dimensional structures, 3-D structures, reaction diagrams, or other structural representations. Associated data can include simple numeric and textual information, graphical data such as spectra, and multiple observation data stored in separate database tables such as biological assay results and physicochemical measurements.

The Cousin add-in for Access was initially created in response to requests from users with a need to design and manage personal chemistry databases. Since its introduction, the add-in has been employed in areas ranging from discovery research to chemical production for the management of personal and shared project and departmental 2-D chemical structure and reaction databases. The forms shown in Figures 1 and 2 are examples typical of the types of databases that users have created with the add-in. The form in Figure 1 includes a subform that contains biological data from a second database table. More complex forms containing multiple subforms, to display data for multiple biological assays, for example, and nested subforms have been created as well.

While the set of databases created at Upjohn to date using the add-in represents examples of several of the possible types of chemical structure databases, application of the add-in approach to other database types can be envisioned as well. For example, in addition to supporting personal databases on the user's local machine and shared databases stored on network servers, Access supports shared databases with tables attached from other database management

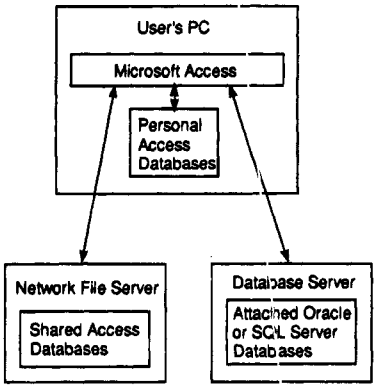

Figure 6. Personal, shared and attached Access databases.

systems such as Oracle[6] and SQL Server[6] as is shown in Figure 6. Since many organizations store their corporate chemical and biological data in systems such as Oracle and SQL Server, exploiting the capability to attach tables from these systems allows Access applications to be developed that manage corporate level data. Going a step further, it is possible to use the add-in to build corporate structure databases in attached Oracle or SQL Server tables. However, when used to search structures stored in large server databases, the present add-in encounters some performance limits which are discussed later in the paper.

One further dimension of chemical database systems is their programmability. User organizations regularly have a need to create custom chemical applications to meet internal scientific and business needs. Examples include applications to support the registry of new molecules and reactions into a database, systems to track compound samples and maintain accurate inventories, and custom interfaces for the retrieval and analysis of structural data and associated information. Advanced relational database systems address this need by providing a set of comprehensive application development tools. Access, for example, contains a complete application development environment that allows a user interface to be readily designed and implemented for an application and then linked to supporting code modules written in a dialect of the Basic programming language. These powerful, yet relatively easy-to-learn and use development environments simplify the application development process and bring the possibility of application development to a broader group of users than has been previously possible. A chemical software component, such as the add-in described here, completes the set of tools needed to create chemical database applications by providing a library of functions that can be called from user-written database code modules to perform chemistry-specific operations on structures and reactions. The set of operations required of a chemical component ranges from low level functions for the creation and manipulation of individual atoms and bonds to higher level functions that act on complete molecules and reactions to furnish services such as molecular weight calculation and structure matching. The combination of a powerful database application development environment and an integrated programmable chemical software component provides the full set of tools required by in-house and third party developers to create custom chemical applications tailored to meet the needs of individual organizations.

PERFORMANCE

The most critical aspect of performance for the present Access add-in is its structure searching speed. Structure searching, particularly substructure searching, has received significant attention over the last two decades[7] and performance has improved considerably. The structure searching method employed in the Cousin add-in uses a traditional two-step approach with an initial screening step followed by an atom by atom matching step and is based on software described previously.[8] A straightforward search technique is used in which each record in the Access database is read sequentially. For a full structure or substructure search, the structure bit screen, which is stored within the structure database field, is checked first to see if a match is possible. Structures that pass the screen are subjected to a detailed atom by atom comparison to determine if they actually match. In the case of a similarity search, a Tanimoto coefficient[9] is calculated using the query and structure screens with the resulting value being used as the similarity score.

Of the three types of structure searching supported by the Cousin add-in, substructure searching is the most computationally demanding. To evaluate the substructure searching performance of the add-in, a test database of 7500 structures was created containing a random selection of compounds having an average of 22 atoms and 22 bonds each. A suite of 16 sample substructure queries ranging from simple to complex and returning from 1 to 7500 hits was then executed against the test database using a Gateway 2000 P5-90 90-Mhz Intel Pentium machine. The shortest search time recorded was 10 s and the longest was 54 s, with a median time of 12 s and a mean time of 16 s, resulting in a median search rate of approximately 37500 structures per min (625 structures/s).

We believe this level of performance is adequate for most personal and project databases as well as for other structure databases ranging in size up to approximately 10 000 structures. For larger corporate or commercial databases containing from 10^5 to 10^6 or more structures, the method presently used in the add-in does not provide acceptably fast structure searching. Performance is being limited by the speed of the workstation processor, the heavy I/O burden imposed by a full sequential database read during the search, and suboptimal characteristics of the matching algorithms. To achieve better search performance on large databases another approach is required.

One solution to this problem, for attached database tables, is to move the search process from the client machine to the relatively more-powerful database server containing the attached tables. In a fashion similar to that used to create the Access add-in, it is possible to add a high performance structure search engine to a database server such as Oracle or SQL Server. With this technique, the user's structure search query is transparently routed from the client machine to the database server where the query is efficiently executed by the add-in structure search engine. Search results are then returned to the client computer for display. By employing this method to improve search speed, the add-in component approach can be extended to provide acceptable performance for the full range of database sizes.

CONCLUSIONS

In this paper we have discussed the concept of a chemical structure software component and its integration into a

relational database management system. We have attempted to show that this approach can be used to create chemical database systems that match or exceed the functionality, performance, and ease-of-use of traditional custom-designed chemical database systems. The approach promises potential benefits for both chemical software developers and their customers. Developers can benefit by being able to concentrate on advancing the chemistry-related features of their software rather than using their resources to duplicate the common database features better-implemented by general relational database system vendors. Customers benefit by having a single full-featured database system that can be used for both chemical and nonchemical applications, thereby reducing software acquisition, maintenance, and training costs.

The Cousin add-in for Access can be viewed as one example of a class of possible software extensions to Access and other advanced relational database systems for creating chemical database applications. Other types of add-in components can be envisioned as well: for example, a 3-D structure component with a full complement of 3-D structure manipulation and analysis capabilities or an IR or NMR spectrum component that supports the storage, display, analysis, and searching of spectra. Such components could be made available by software organizations specializing in individual areas of chemistry and used by third party and in-house application developers as building blocks within the relational database development environment to create a broad range of custom chemistry applications.

Supporting Information Available: The Cousin add-in for Microsoft Access described in this paper along with its associated on-line documentation and a sample Access chemical database is available for downloading from the ACS Internet server. Use of the add-in requires a PC with Microsoft Windows 3.1 and Microsoft Access 2.0. Supporting information is available to subscribers electronically via the Internet at http://pubs.acs.org (WWW) and pubs.acs.org (Gopher). For additional information on accessing supporting information on the ACS Internet server, see any current masthead page. Specific instructions for downloading and installing the add-in software are contained in the "readme" file included with the supporting information.

REFERENCES AND NOTES

(1) Hagadone, T. R.; Lajiness, M. S. Capturing Chemical Information in an Extended Relational Database System. *Tetrahedron Comput. Methodol.* **1988**, *1*, 219–230.
(2) *Accord* for Microsoft Excel from Synopsys and the ISIS SAR Table add-in for Microsoft Excel from MDL Information Systems, Inc. are examples of this type of software.
(3) Udell, J. Componentware. *BYTE* **1994**, *19(5)*, 46–56.
(4) Howe, W. J.; Hagadone, T. R. Molecular Substructure Searching: Computer Graphics and Query Entry Methodology. *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 8–15.
(5) Dalby, A.; Nourse. J. G.; Hounshell, W. D.; Gushurst, A. K. I.; Grier, D. L.; Leland, B. A.; Laufer, J. Description of Several Chemical Structure File Formats Used by Computer Programs Developed at Molecular Design Limited. *J. Chem. Inf. Comput. Sci.* **1992**, 32, 244–255.
(6) The Oracle (Oracle Corp.) and SQL Server (Sybase Inc. and Microsoft Corp.) relational database systems are enterprise-scale database managers. These systems are normally installed on high performance server machines to manage large multiuser databases.
(7) Barnard, J. M. Substructure Searching Methods: Old and New. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 532–538.
(8) Hagadone, T. R.; Howe, W. J. Molecular Substructure Searching: Minicomputer-based Query Execution. *J. Chem. Inf. Comput. Sci.* **1982**, *22*, 182–186.
(9) Willet, P. In *Similarity and Clustering in Chemical Information Systems;* Research Studies Press: Lechworth, England, 1987; Chapter 2, p 54.

CI950043C