Object-relational databases: the next wave in pharmaceutical data management

John F. Cargill and Norah E. MacCuish

Ontogen Corporation and Daylight Chemical Information Systems have recently integrated a molecule data type in an object-relational database management system. The authors explain how this innovation unites chemical and biological data management into a single universal database. All chemical information storage, searching and retrieval is performed using a standard structured query language. The authors describe using rapid application development methods for quickly building an application to access the database and they demonstrate that these techniques can simplify and standardize the management of pharmaceutical data.

High-throughput screening, combinatorial chemistry and bioinformatics continue to pose data integration challenges to the pharmaceutical industry. Screening data have primarily been managed using relational database systems, such as Oracle™ (Oracle Corporation, Redwood City, CA, USA); chemical inventories have traditionally been managed in proprietary non-relational systems, such as ISIS™ (MDL Information Systems, San Leandro, CA, USA); and bioinformatics data, such as protein and genetic sequences, are typically stored in simple flat file systems.

In response to the need for better integration of data, some vendors of proprietary chemical information systems have developed gateways to relational database systems. For example, the ISIS Relational Gateway provides a mechanism to link tables from an Oracle database to chemical structures in the proprietary ISIS database. A similar relational gateway system, called RS^3 – which is available from Oxford Molecular (Oxford, UK) – facilitates the storage of structural connection tables in Oracle tables. Chemical structure searching, however, is enabled through callouts to functions external to the Oracle 7 server. These two gateway solutions suffer from a lack of true integration with the underlying relational database. For example, it is difficult to optimize queries because the proprietary database cannot participate in the query optimization performed in the relational database. Moreover, such gateway solutions cannot be easily extended to manage new types of data. For example, to add the capability to perform chemical spectral searching, both a separate proprietary database designed for this purpose and another relational gateway are required.

There is a need in the pharmaceutical industry for a 'universal' database. Such a database would have the capability to store and query specialized types of scientific data; it would allow the user to 'plug in' modules for management of specialized pharmaceutical data types, such as chemical structures, spectra and genetic sequences; and each of these plug-in modules would run as a native component of the underlying database.

This need is about to become a reality in a form called the 'object-relational database'. Database vendors such as Oracle and Informix (Menlo Park, CA, USA) are currently introducing this new technology. It is evident that there will be a

John F. Cargill*, Ontogen Corporation, 2325 Camino Vida Roble, Carlsbad, CA 92009, USA and **Norah E. MacCuish‡**, Daylight Chemical Information Systems, 419 E. Palace, Santa Fe, NM 87501, USA. *tel: 760 930 0100, fax: +760 930 0200, e-mail: john.cargill@ontogen.com ‡e-mail: norah@daylight.com

1359-6446/98/$ – see front matter © Elsevier Science. All rights reserved. PII: S1359-6446(98)01264-1

widespread adoption of this technology in the pharmaceutical industry over the next five years. This review examines how one object-relational database was used to create a fully integrated chemical information database solution.

Database management systems

Relational database management systems

The majority of database systems in use today in the pharmaceutical industry are relational database management systems (RDBMS)[1]. Such systems were designed for handling high-speed transactions for structured query language (SQL) queries on simple data types. The SQL-92 standard has given the RDBMS industry a common query language to support interoperability and familiarity.

A major limitation of RDBMS systems is that they support a limited set of data types such as character, integer, floating point and date. Many of the major RDBMS commercial providers (e.g. IBM, Oracle and Informix) have found the need to extend or rewrite their current systems to handle more complex types of data and more complex queries. This has led to the development of a new hybrid database technology known as object-relational database management systems (ORDBMS)[2].

Object-relational database management systems

An ORDBMS provides facilities to add objects to the tables of an RDBMS. Each object is a database entity that has a set of behaviors and a set of properties. The values of the object's properties represent the 'state' of the object and the behaviors provide a means to change that state. For example, if the object 'circle' is added to our ORDBMS, the properties of it might be radius and color, and the behaviors might be 'change size' and 'change color'. In an ORDBMS, an object is implemented as a new data type of arbitrary complexity with a set of functions that operate upon it. The ORDBMS introduce a new generation of the SQL known as SQL 3. This language provides facilities to create objects and to assign properties and behaviors to them. Some ORDBMS products include other object-oriented features such as table hierarchy and functional inheritance.

Client applications written for RDBMS may have to be rewritten to handle the new data types and functions of an ORDBMS. To take full advantage of systems with new data types, table hierarchy and functional inheritance, the data originally modeled for a relational system may require a completely new data model.

Object-oriented database management systems

In pure object-oriented database management systems (OODBMS)[3] data are modeled in terms of pure objects, which are managed through an object request broker (ORB). Unlike the ORDBMS, the OODBMS is based on a collection of objects rather than a collection of tables. Standards are required to facilitate the interoperability of heterogeneous objects through a common ORB. The Object Database Management Group (ODMG) is a consortium that has driven the standardization of object databases[4]. Bioinformatics database providers are among the first to adopt this new technology[5].

OODBMS are a promising technology but are largely incompatible with the RDBMS technology currently in use in most pharmaceutical companies. In addition, the market share and capitalization of OODBMS vendors remains a fraction of their RDBMS competitors.

Adding chemical intelligence to the Informix database

Chemical and biological data handling within an ORDBMS has been achieved by integrating the Informix Universal Server and Universal Data Option (IUS/UD) with the Daylight Chemical Toolkits (Daylight Chemical Information Systems, Mission Viejo, CA, USA). The Daylight database extensions facilitate the development of applications for chemical and biological data integration. It is arguable that the flexibility and adaptability of such a system provide a powerful motivation for adoption of an ORDBMS.

The IUS/UD can be extended to handle new data types and new query processing with database extensions called datablades. These are modules that can be plugged into a database kernel to add new data types and functions to SQL.

Datablades are developed with the Datablade Development Kit (DBDK) that is bundled with the IUS/UD. Daylight's datablade is currently in beta testing. Another commercial chemical datablade currently in development is Synopsys' Accord/Universal Server (Synopsys Scientific Systems, Leeds, UK). Chemical datablades provide chemical structure and reaction handling, storage and querying within an Informix database. The Daylight datablade adds functional extensions and new data types to the Informix SQL-92-compliant SQL. The datablade was created with the Daylight Toolkits and the DBDK. Care has been taken to integrate these two technologies at the server level, ensuring proper memory management and performance advantages.

The Daylight datablade enables the storage of information about chemical structure using a newly defined data type – the molecule data type. Molecules can be stored within a relational table and structure-searching techniques can be applied, such as substructure, superstructure, exact match and similarity searching. Molecules are individual compounds, salts or reactions. SMILES (simplified molecular input line entry specification) is a character representation of the two-dimensional connectivity of a molecule[7]. Selecting a molecule from a table results in a Molecule SMILES or a Reaction SMILES string being passed to the client application.

Within the molecule data type, information is stored to facilitate similarity, substructure and superstructure searching. Changes to the underlying molecule type can remain transparent to the client applications. An example of an SQL query substructure search in which an aromatic carbon is attached to an aliphatic carbon is:

```
select molecule from table where
day_findassub('Cc',molecule)
```

Queries can take the form of Molecule SMILES, Reaction SMILES or as SMARTS (SMILES arbitrary target specification). SMARTS is an extension to the SMILES language that allows specifications of substructures; for example, 'Cc' would query for all structures containing the 'Cc' substructure. Through the use of SMILES and SMARTS chemical queries can be combined with screening data queries within a single SQL statement.

For basic chemical information storage and retrieval, the Daylight datablade provides the majority of the requirements of a chemical database system. However, custom datablades can be written using the Daylight Toolkit programs and the DBDK. These two technologies thus provide the desired flexibility for proprietary data handling using SQL functional extensions. Also, a C toolkit programmer is able to incorporate application needs without requiring a vendor to add to an existing datablade.

Rapid application development (RAD)

Simply producing a unified database of chemical and biological data cannot solve all data management problems in the pharmaceutical industry. Applications must be developed quickly in order to use these data. Typically the development of such applications is both slow and costly.

In the past few years the software development industry has addressed these issues with the introduction of rapid application development (RAD) tools and methodologies. RAD involves two powerful strategies: first, the creation of software systems from pre-existing software components and, second, the use of powerful applications development environments (ADE). Visual Basic™ (VB) from Microsoft Corporation (Redmond, WA, USA) is probably the best-known ADE, although Java™-based products are gaining widespread use. A variety of powerful tools exist in VB for constructing, populating and querying ORDBMS.

Recently, several pre-built software components have been developed specifically for the use of RAD projects in the pharmaceutical industry. These components perform such tasks as editing and displaying chemical structures.

By combining the existing base of RAD tools with a powerful, chemically enabled ORDBMS and a set of pre-built software components, it is now feasible to reduce dramatically the time and cost involved in developing software applications for drug discovery.

We have used RAD tools and the Daylight datablade to create an SAR spreadsheet application. The following software tools were chosen to build this application:

- Visual Basic 5 (Microsoft)
- True DB Grid Pro 5.0 (Apex Software, Pittsburgh, PA, USA)
- Accord Chemistry Control (Synopsys Scientific Systems)
- ISIS Draw Chemical Structure Editor (MDL Information Systems)

These tools provide all the raw functionality necessary to store, retrieve, display and query both biological and chemical information from the Informix ORDBMS.

SAR spreadsheet

The following example demonstrates how the SAR spreadsheet application uses an object-relational database in drug discovery. Ontogen maintains a database of high-throughput screening data on >100,000 compounds synthesized by the OntoBLOCK™ technology for rapid combinatorial chemical synthesis[8]. The application queries and displays SAR data from this source.

Initially, the 'Query Builder' window (Fig. 1) is opened and the 'Substructure Search' is selected. An ISIS Draw component activates and the desired compound structure query is sketched (Fig. 2). After closing ISIS Draw the program returns to the Query Builder window. The query string reads:

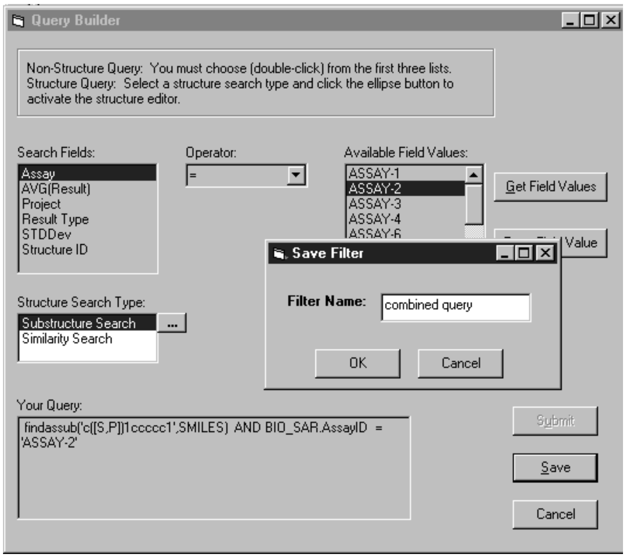
```
day_findassub('c([S,P])1ccccc1', SMILES)
```

This query string is the 'where clause' of a complete SQL statement. The full SQL statement is hidden from the user for clarity. The full SQL statement is:

select * from bio_sar where day_findassub('c([S,P])1ccccc1', SMILES)

The English translation of this SQL statement is:

Select all rows from the database table 'bio_sar' where the molecules in the column called 'SMILES' contain a six member aromatic carbon ring and either a sulphur or a phosphorous attached to the ring.


Non-Structure Query: You must choose (double-click) from the first three lists.
Structure Query: Select a structure search type and click the ellipse button to activate the structure editor.

Filter Name: combined query

Your Query:
findassub('c([S,P])1ccccc1',SMILES) AND BIO_SAR.AssayID = 'ASSAY-2'

__Figure 1.__ The Query Builder provides a simple and effective system for handling the biological and chemical information held within the Informix object-relational database management system.

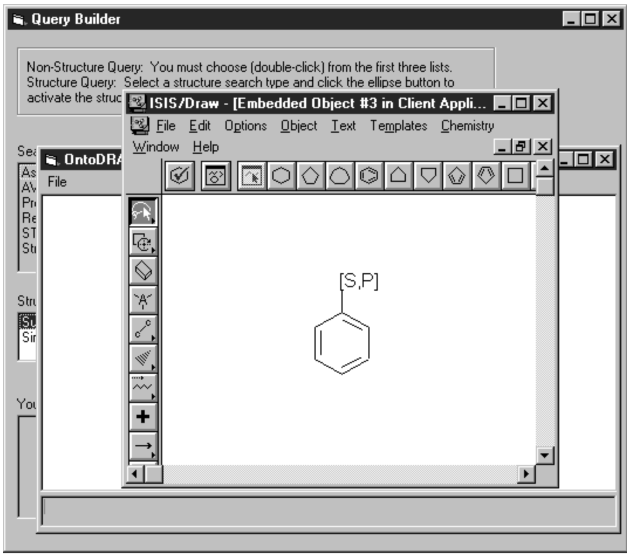

Non-Structure Query: You must choose (double-click) from the first three lists.
Structure Query: Select a structure search type and click the ellipse button to activate the struc

__Figure 2.__ ISIS Draw in the Query Builder simplifies the creation of SMARTS.

The day_findassub( ) function accepts two arguments and returns TRUE or FALSE. The first argument is a molecular query expression formulated in the SMARTS language. The second argument is the name of a column in a database table that is of the data type 'molecule'.

Although the SAR program constructs the SMARTS pattern automatically from the ISIS Draw sketch, it is possible to edit the SMARTS string and modify the search criteria – this requires knowledge of the SMARTS language.

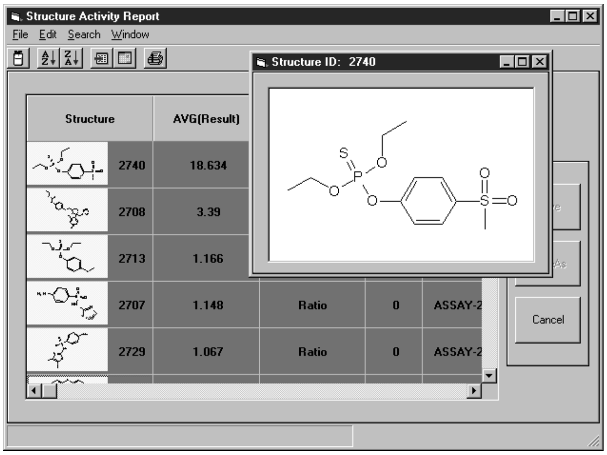
To continue building the query, one uses the data fields and data operators displayed in the Query Builder. First, 'Assay' is selected from the 'Search Fields' list box (Fig. 1). Then the operator '=' is selected from the 'Operator' list box. By pressing the button labeled 'Get Field Values', a list box is populated with all the valid choices of assay names. The values displayed in the example are ASSAY-1 to ASSAY-6. By clicking on the selection 'ASSAY-2' and clicking the 'Save' button, the SQL query now reads:

select * from bio_sar where day_findassub('c([S,P])1ccccc1', SMILES) AND BIO_SAR.AssayID = 'ASSAY-2'

The English translation of this SQL statement is:

Select all rows from the database table 'bio_sar' where the molecules in the column called 'SMILES' contain a six member aromatic carbon ring and either a sulphur or a phosphorous attached to the ring AND 'AssayID' is equal to 'ASSAY-2'.

The 'Submit' button is now active. When it is pressed the query executes and a spreadsheet of the results, including structure is displayed (Fig. 3). Each column in the spreadsheet may be dragged to another location and the sort order of each column may be changed. By double-clicking on a structure, it will be transferred to a resizable window.



Structure		AVG(Result)			
	2740	18.634			
	2708	3.39			
	2713	1.166			
	2707	1.148	Ratio	0	ASSAY-2
	2729	1.067	Ratio	0	ASSAY-2

Figure 3. *The SAR spreadsheet presents the results of a query in an adjustable format.*

Conclusion

It is demonstrable that using chemically intelligent ORDBMS in conjunction with modern RAD software tools and commercial shrink-wrapped software components can reduce development time and simplify the management of pharmaceutical data. Over the next few years it is likely that all major vendors of commercial chemistry information management systems offer their products as plug-ins to ORDBMS products from database vendors such as Informix and Oracle. Using the techniques described above it is possible to make significant progress in managing the increasingly large and complex data sets in pharmaceutical drug development programs.

ACKNOWLEDGEMENTS

We thank Al Finocchio, Judy Huang and Tina Navarro of Ontogen for assistance in coding the SAR application and Jack Delany of Daylight Chemical Information Systems for assistance in coding the Daylight datablade.

REFERENCES

1 Ullman, J.D. (1988) *Principles of Database and Knowledge-Base Systems*, Computer Science Press

2 Stonebraker, M. *et al.* (1996) *Object-Relational DBMSs The Next Great Wave*, Morgan Kaufmann Publishers

3 Cooper, R. (1997) *Object Databases: An ODMG Approach*, International Thomson Computer Press

4 Cattell, R.G.G. *et al.* (1997) *The Object Database Standard: ODMG 2.0*, Morgan Kaufmann Publishers

5 Slidel, T. (1997) *BioInformer* July, European Bioinformatics Institute

6 Hagadone, T.R., *et al.* (1993) *Chem. Struct. 2 Proc. Int. Conf.* (2nd edn) (Warr, W., ed.), pp. 257–269, Springer

7 Weininger, D. (1988) *J. Chem. Inf. Comput. Sci.* 28, 31–36

8 Cargill, J.F. *et al.* (1996) *Lab. Robotics Autom.* 8, 139–148