3-1993

Connecting Scientific Programs and Data Using Object Databases

Judith Bayard Cushing
Oregon Graduate Institute of Science and Technology

David Hansen
George Fox University, dhansen@georgefox.edu

David Maier
Oregon Graduate Institute of Science & Technology

Carlton Pu
Oregon Graduate Institute of Science and Technology

Connecting Scientific Programs and Data Using Object Databases

Judith Bayard Cushing *David Hansen* *David Maier* *Calton Pu*
Dept. of Computer Science and Engineering
Oregon Graduate Institute
Beaverton, Oregon 97006-1999

Abstract

Scientific applications and databases rarely interoperate easily. That is, scientific researchers who use computers expend significant time and effort writing special procedures to use their program with someone else's data, or their data with someone else's programs. These problems are exacerbated in modern computing environments, which consist of multiple computers of possibly different types. Database researchers at the Scientific Database Laboratory at the Oregon Graduate Institute are using object-oriented databases to address problems of program and data interoperability. For the domain of computational chemistry, we are extending an existing object database system to facilitate the invocation, monitoring, and output capture of a variety of independently developed programs (aka legacy applications). A complementary project in materials science explores providing application programs with a common interface to a variety of separately published datasets. We are also developing an object-oriented toolbox to access the contents of a database of protein structures. We describe these three projects, then discuss their status and our future directions.

1 The Interoperability Problem for Scientific Computing

In a perfect world, data from one program could be transparently used as input to another. The world of scientific computation unfortunately is far from perfect, and its rich legacy of data and programs carries a major disadvantage: a plethora of data formats and input conventions. Business data processing has worked to solve this problem through common data models and shared databases, but current record-oriented database technology (such as relational database systems) does not match scientific applications well. Scientific data types such as multi-dimensional matrices or crystal structures cannot be implemented efficiently and directly using record-oriented models. We believe that object-oriented systems avoid such shortcomings, and we have identified computational chemistry, materials science, and protein structure analysis as areas in which to explore object-oriented systems that integrate diverse programs and data. Our approach constructs, for each domain, a unifying data model that encompasses a range of programs and data sources, creating a "plug-and-play" environment.

2 Diverse Computational Chemistry Programs

In the realm of computational chemistry, programs implementing quantum mechanics algorithms compute molecular properties given basic molecular structure data. These computationally intensive applications require the storing, viewing, and sharing of large amounts of specialized quantitative information, and could benefit from

using database systems. The environment of the computational chemist is further complicated in that these standalone applications run on different kinds of computers, and data must be transferred between them. The computational chemistry database project, a joint effort with Battelle Pacific Northwest Laboratories, aims to provide a database of past experiments and to render results computed by different codes comparable[CMR+92b, CMR92a].

Capturing both inputs and outputs in common formats, however, requires connecting computational chemistry programs directly to the database. To that end, we are currently defining and implementing a mechanism, dubbed "computational proxy", that relies on a common computational model along with descriptions of the applications' input and output files to provide the required interfaces. A computational proxy object "stands-in", within the database, for a computational experiment in preparation, currently in process, or recently completed. Using the proxy mechanism and the information in the proxy, a user is able to start up and control ongoing computational processes, and capture information about a given computational experiment. When the user schedules a run, a proxy uses a description of the application to automatically transform experimental attributes held in the database into textual inputs appropriate for a given application. If necessary, the input files are transferred to the computer on which that application is to run[CMR93]. Figure 1 illustrates the computational proxy encapsulation of syntactic detail of different programs and computer types.

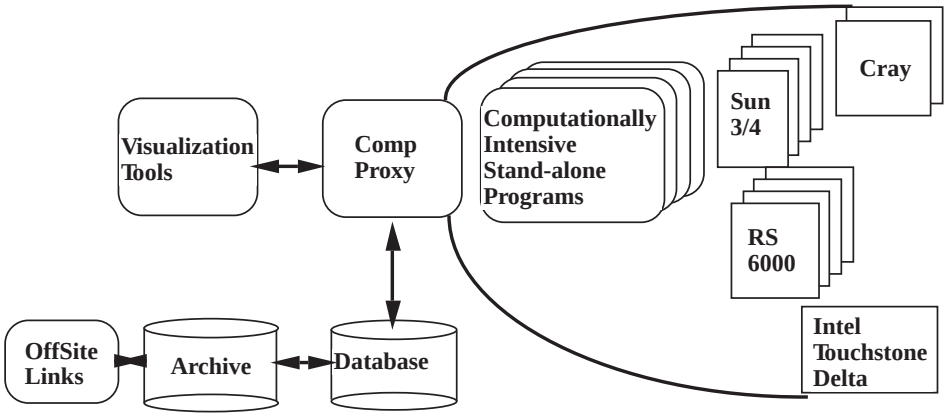

Figure 1: Computational proxies encapsulate computational applications.

Computational proxies aid in interfacing standalone, heterogeneous applications to a common database, simplifying the computing environment for domain scientists and masking syntactic differences among scientific applications. Proxies also help in the capture of inputs, intermediate results, and outputs of computational experiments, as well as associated descriptive data. Since the proxy mechanism renders these data into a common format, data from different applications can be compared, and the output of one program can be more easily used as the input to another. Computational proxies have been implemented in C++ and the object-oriented database system ObjectStore for the General Atomic and Molecular Electronic Structure System (GAMESS)[Rao93]. The project will later support additional application programs such as Gaussian. We believe proxies are generalizable to other computational programs, different domain sciences, and any object-oriented database system.

3 Integrating Materials Science Data Sources

Materials scientists are prolific users of computers. Modeling techniques and algorithms are well known, and refined and widely available computer-readable factual databases abound. Unfortunately, any given materials science application is typically developed in isolation, using a specifically tailored data model. Furthermore, scientists typically access available computerized databases manually, in an off-line fashion. Thus, researchers

repeatedly construct and populate new custom databases for each application. Our materials science database research bridges the gulf between applications and multiple sources of data by providing a uniform object interface to datasets in diverse formats.

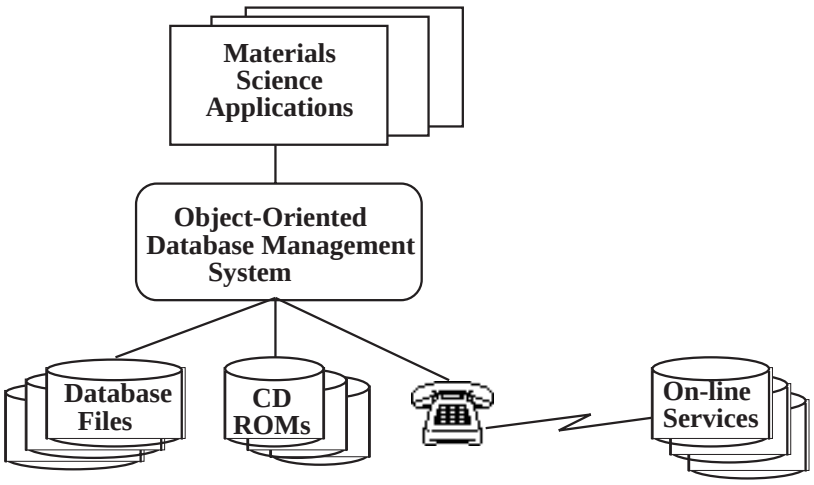

Figure 2: A single interface between programs and sources of data.

We have developed a unifying object-oriented data model to meet the needs of several materials science applications. This data model captures the essence of molecular and crystalline structure from a materials science perspective. We have implemented this data model in an object-oriented materials science database using the GemStone object-oriented database management system[HMSW93].

The database stores materials science data generated by users and user applications and provides transparent access to heterogeneous, commercial data sources. (See Figure 2.) The database currently provides access to the Electron Diffraction Database, distributed by the National Institute of Standards (NIST) on CD-ROM, as well as to files generated by the CAChe computer-aided chemistry system and the Desktop Microscopist. External data is cast into objects of the data model, providing users of the database with a single, object-oriented model of both internal and external data.

4 The PDB Toolbox

The Protein Data Bank (PDB) is a depository for the atomic structure of protein (and other) macro molecules. Currently with about 900 molecules occupying about 270 megabytes, the PDB is expected to grow to more than 6000 molecules by year 2000. Because of the complex structure of protein molecules, attempts to use relational DBMS's have not been totally successful. Jointly with Columbia University and Brookhaven National Laboratories, we are developing an object-oriented toolbox for PDB that will use modern software engineering principles and object-oriented DBMS technology[PSO+92]. The toolbox consists of software instruments to access the contents of PDB and will have the reliability, performance, usability, and extensibility of modern computer software. Now under development, the first demonstration program, PDBTool, will be used to verify molecular structures.

Because of the complex structure of PDB data, we have chosen an object-oriented approach and have found object identities, encapsulation, data abstraction, and inheritance to be critical features. Requirements of the PDB Toolbox include: (1) inspecting protein structures at different abstraction levels, i.e., chain, secondary structure, residue, and atom; (2) graphically presenting raw data and derived data; and (3) interfacing with other data sources such as the original PDB file format, relational databases such as SESAM, and standard formats such as the Crystallographic Interchange Format.

We have developed a three-level general architecture for the management of scientific data, applicable to many domains. To build the *user interface*, we used SUIT software developed at the University of Virginia, which provides graphical widgets to represent data in a windowed environment. We have implemented *verification algorithms* in C and C^{++} that calculate information such as torsion angles and Ramachandran plots. To implement the *storage manager* we relied on PDB files, the current standard storage system for crystallographic data.

The prototype PDBTool runs on any Sun platform and has been remotely run on an IBM RISC 6000 and an SGI Indigo workstation. The currently implemented verification tools accessible through PDBTool include the Ramachandran plot tool, which calculates and displays the ϕ and ψ angles of each residue, a 3-D graphical display of molecular structure, and histogram displays of geometric factors such as chain bond length, torsion angles, and chiral volumes. New tools as well as a protein query language are being actively developed and implemented.

The PDB Toolbox project has a scope larger than scientific data management, and a critical component is data storage in backends and efficient access to them. We have designed an abstract interface to backends, which consists of sequential access and direct access. Sequential access is implemented in the PDBTool as C^{++} iterators. We have implemented C^{++} iterators for the original PDB file format, in ObjectStore (using DML), and in the SESAM relational database.

5 Status and Future Work.

The current implementation of the proxy mechanism for GAMESS involves considerable custom code in the database. The main task currently is to develop a means to register new programs without the custom coding, and to explain how to adapt computational proxies to other program types and other domains. Among other things, the registration process involves specifying how inputs are to be formatted and how outputs are to be extracted. Some of this work is already being transferred to PNL and incorporated in the design of their laboratory support database for the Environmental and Molecular Sciences Laboratory.

In the materials science area, we have designed an initial domain model for crystallographic data, implemented that model in an object database, and constructed connections to external data sources. The connection to external sources uses a layered software architecture to hide format differences and give control over caching policies. We have been experimenting with the performance available with this architecture, and have been comparing that to a reimplementation in a different OODBMS. The next step is to look at approaches to connect the database to the Desktop Microscopist application. We are looking at extending our model and database support to accommodate information needed to calculate phase-structure diagrams.

The PDBTool architecture has been validated by the success of PDBTool development, demonstrated in the 1992 ACA meeting. Adding successive tools has been reasonably smooth. We are on schedule for the beta release of PDBTool as a software instrument for molecular verification in 1993.

Our long-range goal is a *Hybrid Data Manager* that contains generic facilities for connecting scientific programs and datasets in a variety of domains. However, our experience indicates that such database support is effective only after careful construction of a conceptual model to give a well-defined semantic basis to underlie the datasets and programs in use[MCPH93].

6 Acknowledgements

The authors acknowledge the collaboration of Dr. David Feller, Dr. Mark Thompson and D. Michael DeVaney of PNL in computational chemistry; Prof. James Stanley and Ramachandran Venkatesh of OGI in materials science; and Phil Bourne, Edwardo Aleesio and others of Columbia University, as well as members of the Brookhaven National Laboratory, in protein structure. Meenakshi Rao and Donald Abel, of OGI and Portland State, programmed

components of the computational chemistry database. Prof. Jonathan Walpole and Prof. Michael Wolfe of OGI are working with us on the design of the Hybrid Data Manager.

This work is supported by NSF grants IRI-9117008 and IRI-9116798, additional grants from the Oregon Advanced Computing Institute (OACIS) and PNL, and software grants from Object Design, Inc., and Servio Corporation. CAChe is a registered trademark of CAChe Scientific; Gaussian of Gaussian, Inc.; GemStone of Servio Corporation; and ObjectStore of Object Design, Inc. GAMESS is distributed by North Dakota State University and the USDOE Ames Laboratory; and the Desktop Microscopist by Virtual Laboratories.

References

[CMR92a] J. B. Cushing, D. Maier, and M. Rao. Computational chemistry database prototype: ObjectStore. Technical Report CS/E-92-002, OGI, Beaverton, OR, 1992.

[CMR+92b] J. B. Cushing, D. Maier, M. Rao, D. M. DeVaney, and D. Feller. Object-oriented database support for computational chemistry. *Sixth International Working Conference on Statistical and Scientific Database Management (SSDBM)*, June 9-12 1992.

[CMR93] J. B. Cushing, D. Maier, and M. Rao. Computational proxies: Modeling scientific applications in object databases. Technical Report CS/E-92-020, OGI, Beaverton, OR, 1993.

[HMSW93] D. M. Hansen, D. Maier, J. Stanley, and J. Walpole. An object oriented heterogeneous database for materials science. *Scientific Programming*, to appear 1993.

[Ken92] B. Kennedy. Architectural alternatives for connecting a persistent programming language and an object-oriented database. Master's thesis, OGI, Beaverton, OR, 1992.

[MCPH93] D. Maier, J. B. Cushing, G. Purvis, and D. Hansen. Object data models for shared molecular structures. *to be presented at the First International Symposium on Computerized Chemical Data Standards: Databases, Data Interchange, and Information Systems, Atlanta GA*, May 5-7 1993.

[Ohk93] H. Ohkawa. *Object-Oriented Database Support for Scientific Data Management: a System for Experimentation*. PhD thesis, OGI, Beaverton, OR, 1993.

[PSO+92] C. Pu, K.P. Sheka, J. Ong, L. Chang, A. Chang, E. Alessio, I.N. Shindyalov, W. Chang, and P.E. Bourne. A prototype object-oriented toolkit for protein structure verification. Technical Report CUCS-048-92, Dept. of Computer Science, Columbia University, 1992.

[Rao93] M. Rao. Computational proxies for computational chemistry: A proof of concept. Master's thesis, OGI, Beaverton, OR, expected: June, 1993.