1994

Applications of Object Database Technology in Thermodynamics and Materials Science

R Venkatesh
Oregon Graduate Institute of Science & Technology

David Hansen
George Fox University, dhansen@georgefox.edu

David Maier
Oregon Graduate Institute of Science & Technology

James T. Stanley II
Oregon Graduate Institute of Science & Technology

Applications of Object Database Technology in Thermodynamics and Materials Science

R. Venkatesh [*] David Hansen, [†] David Maier, [‡]
James T. Stanley II [§]

Oregon Graduate Institute of Science & Technology
20000 NW Walker Road, Portland, OR 97291

Abstract

Over the last few decades there has been a continual development of computer programs and databases directed at storage and manipulation of scientific data. The data these programs manipulate are typically kept in a proprietary format with little thought given to portability, upgrading or interfacing with other software. Hence, there is an inability to share data across programs or between data sources. We believe that many of the difficulties with sharing scientific data and building on previous computational efforts result from the incompatibility of programs and data in the scientific domain. We are involved in exploring an alternative data management technology, namely object-oriented databases (OODB) to develop solutions to these questions. This paper outlines the difficulties in the current representation of scientific data and discusses some features of OODBs that help overcome these difficulties. Then, a data model is presented that effectively models both the crystallographic and thermodynamic domains allowing easily extensible and maintainable data and programs.

[*] venkar@cse.ogi.edu, Department of Materials Science & Engineering

[†] dhansen@cse.ogi.edu, Department of Computer Science & Engineering

[‡] maier@cse.ogi.edu, Department of Computer Science & Engineering

[§] jstanly@mse.ogi.edu, Department of Materials Science & Engineering

Introduction

Managing scientific data effectively is an implicit requirement in most research efforts in thermodynamics and materials science. The objectives of using computerized solutions for data management in materials science include

- *traditional* objectives such as performance and efficiency,
- *current* needs such as functionality and inter-operability, and
- *future* wish-lists including information sharing, dissemination, maintainability and portability.

Information dissemination is gaining importance with the increased accessability of systems via online networks such as the internet. The role of electronic information dissemination will be increasingly crucial in the scientific community.

The levels to which these objectives are supported will depend in large part on the physical level data storage, the logical level data representation, and the kinds of user-level data manipulation.

Data to manage in materials science includes compiled databases, outputs/results from computer models (applications) and simulations, and laboratory experiments. Current solutions in materials science adopt combinations of three major approaches for data management:

- Hardcopy in the form of reports, publications, and lab notebooks.
- Flat file systems to store raw data.
- Spreadsheets and other tabular representations for data analysis.

Experimental results in materials science are usually stored in flat data files on computers. The experimental setup and parameters are typically stored in laboratory notebooks. Search and retrieval of the data is manual. Data is transferred between analyses tools using manual techniques or special skeletal file conversion programs in some cases. Most researchers use spreadsheets or relational database software to store the results from experiments.

The current solution scheme essentially involves storing the different parts of an experiment (the conditions, the data and the analyses) on different media (archives, hard copy, and magnetic media respectively) with different logical and physical representations. Hence the integrated analysis of a complete experiment is inefficient and in some cases impossible. Associative (content-based) searching of information is not well supported. In such a setup it is also difficult to enforce standards, which are important for data with projected long term utility, such as weather information or radioactive waste monitoring.

Storing scientific data in a unified database offers several advantages compared to any of these methods, including data independence, reliability, concurrency control and efficient access methods. Current database solutions are relational, that is, the data is organized as *tuples*, or rows in a table, in a manner analogous to spreadsheets. Relational databases also come with a well defined *query language* that can be used to compose queries that return selected parts of the table, in response to user-defined criteria.

However, using relational databases for managing scientific data is not an ideal solution. There is a lot of inherent structure and connections associated with scientific data that are not preserved when they are captured as just entries in tables. The structure is important because it can be used in subsequent manipulations of the data.

We are currently exploring an alternate technology, object-oriented databases, for data management in the domain of thermodynamics and materials science. Data considered include data from laboratory experiments, numerical models, simulation studies and published materials science databases. This paper recounts some of our experiences with using object databases in materials science.

Object-Oriented Database Technology

Object-oriented database technology [1] provides persistence and storage support for objects. An object is a computer representation of a real-world entity. The object representation captures both structure, the data members, and behavior, the methods (procedures) that operate on the data. Operations are performed by sending *messages* to objects. On the receipt of a message, an appropriate method is invoked. A *class* can be viewed as a template or cookie-cutter for objects of a particular type. *Instances* are the objects that are created from the class definitions, that is, the cookies themselves. An example class definition for an element from the periodic table and an instance from that class definition are shown below to illustrate these concepts:

```
Class Element

Structural Components:

Name                  a String;
Chemical Formula      a String;
Atomic Number         a Real Number;
Isotopes              a Dictionary of Names and Atomic Weights;

Behavioral Components:

printName   /* a function which displays the name */
printNumber /* a function which prints the atomic number */

Instance Sulphur

Structural Components:

Name                   Sulphur;
Chemical Formula       S;
Atomic Number          16.0;
Isotopes               {
                          32S  31.97
                          33S  32.93
                       }
```

A client can print the name of an element object by sending a *printName* message to it. Some of the other features of most object data models include support for inheritance, encapsulation and polymorphism. Inheritance is the ability to *specialize* from base classes to define *sub* classes. In this way, a base class with many properties can be defined first, and from that class, derived classes with special properties can be realized. Hence inheritance provides a good mechanism for representing taxonomic classification and also for code reuse. For example, two sub-classes of class Element can be defined as follows:

Class Interstitial Element: It has all the properties of an element with an additional parameter, the atomic radius, which is defined as

```
Atomic Radii          a Dictionary of Co-ordination Numbers and Values;
```

Class Transition Element: This class also has all the properties of an element with the stipulation that it must have an unfilled *d* orbital in its electronic configuration, additional information that is not associated with the base class Element.

```
Electron Configuration          a Set of Orbitals;
```

Encapsulation has to do with information hiding. Since all external interactions with clients occur via the message interface, the internal representation of an object is completely hidden or encapsulated from the users of the object. As long as the message interface is kept consistent, the internal representation can be changed without the users of the object being affected by the change. For example, in the Element class definition, if the only way that users can access the Atomic Number information is via the print-Number message, as long as that message does the right thing, the internal representation of Atomic Number can be changed from Real Number to Integer without external users being affected. Encapsulation is a powerful feature for controlling incremental changes in systems and providing good maintainability of data.

Polymorphism can again be understood in terms of the message interface. Since users only send messages to objects, and the actual task of carrying out the requested service rests with the object, the same message when sent to objects of different kinds may result in different actions being performed. This property allows operators to operate across different types as long as they share some common behavior, which is useful in that existing applications can work with newly defined classes.

Integrating Materials Science Data Sources

Although materials scientists are prolific users of computerized data, materials science application programs are typically developed in isolation, each using a specifically tailored data model. Even with the advent of data interchange standards such as CIF, materials scientists typically access computerized data manually in an off-line fashion, then load relevant data into their application programs. Our materials science database research bridges the gulf between users' application programs and multiple sources of data. It provides a uniform object interface to datasets in diverse formats through a single, powerful database management system interface [3, 2].

We began by developing a simple, unifying object-oriented data model to meet the needs of several materials science applications (see Figure 1). This data model captures the essence of molecular and crystalline structure from a materials science perspective. The data model was then implemented in an object-oriented materials science database using the GemStone[1] object-oriented database management system (OODBMS).

[1] GemStone is a registered trademark of Servio Corporation

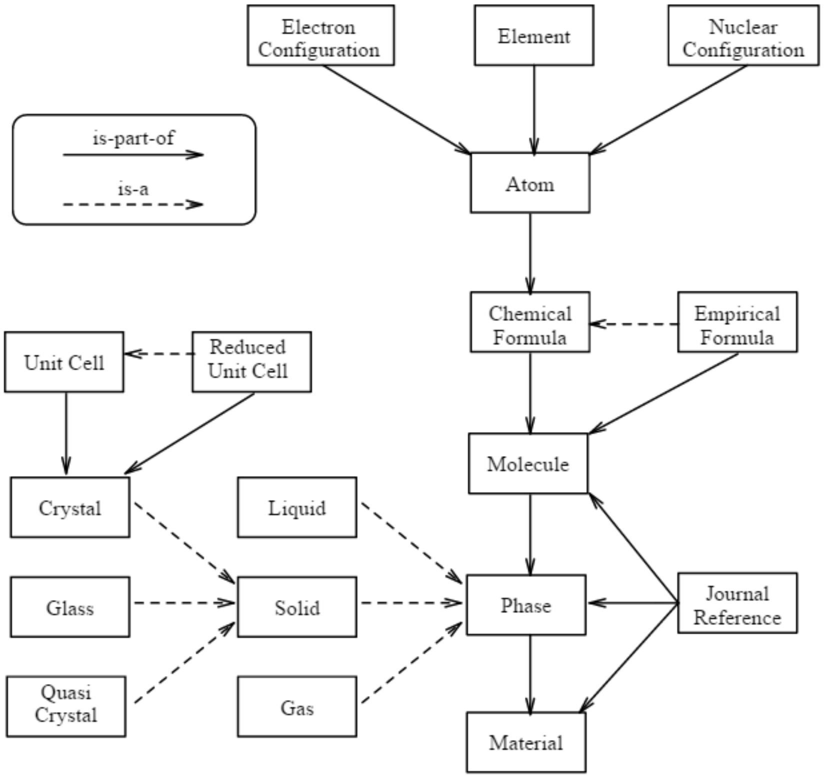

Figure 1: Object-Oriented Data Model for Materials Science

The OODBMS is used to construct an object-oriented heterogeneous database (OOHDB) for accessing materials science databases [4]. The OOHDB provides users and their application programs with a uniform object-oriented model of heterogeneous databases as depicted in Figure 2. Users and their application programs access data through the OOHDB, which transparently accesses relevant external databases and casts the data into objects of the data model. Query processing can be tuned by constructing indices within the OOHDB, and by caching frequently accessed data within the OOHDB.

The database currently provides access to the EDD and PDF-2 published databases as well as to CIF formatted files, files generated by the Desktop Microscopist[2], and files generated by the CAChe[3] computer-aided chemistry system. New databases can be added to the OOHDB incrementally without having to modify applications that are using the OOHDB.

Integrating Data Intensive Applications

The Desktop Microscopist

The Desktop Microscopist is a commercial program for numerical and graphical representations for x-ray and electron diffraction. It can be used in conjunction with diffraction data from compiled datasets and TEM/X-Ray analyses for crystal structure and space group identification.

An experimental version of the Desktop Microscopist is being modified to access files and databases through the OOHDB. The modified program will be able to query all rel-

[2] The Desktop Microscopist is distributed by Virtual Laboratories

[3] CAChe is a registered trademark of CAChe Scientific

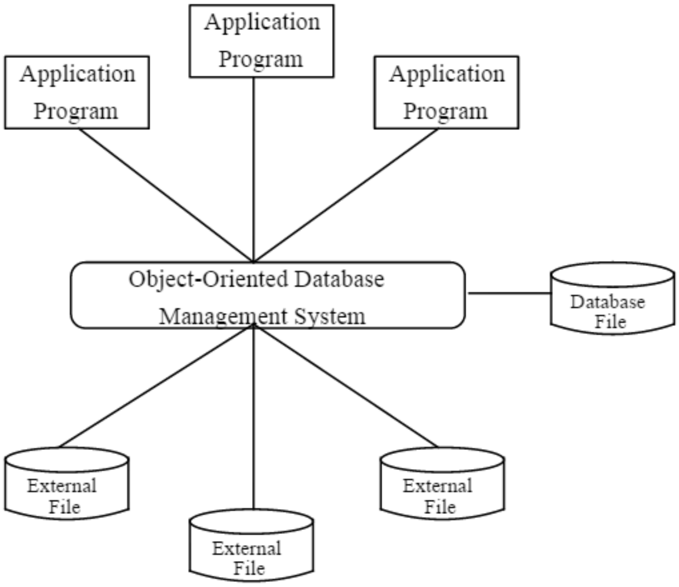

Figure 2: A Single Interface Between Programs and Sources of Data

evant databases simultaneously, thus increasing the search space used by the program to make predictions about crystal structure. Also, instead of manipulating the datasets individually, it can express queries in the high-level query language provided by the database and pass on the queries to the database for execution. This allows for a more extensible design (new databases can be added to the OOHDB without requiring any modifications in the Desktop Microscopist). Since the execution of the queries is controlled by the OOHDB, they can be optimized by the use of indices or other access strategies. Such optimizations are completely encapsulated from the program and can be modified without requiring changes in the program.

The Phase Diagram Calculator

The Phase Diagram Calculator is a computer model is based on the Engel Brewer Correlation[5]. The Engel Brewer Correlation is an example of a general semi-empirical calculation scheme. Semi-empirical schemes involve identifying correlations between data and properties. Such schemes are characterized by two features namely limited computational complexity and extensive comparisons with large datasets.

We believe that such programs have to be closely coupled with a database engine in order to be efficient. The better the underlying support for encapsulation and heterogeneity, the more extensible the model. With time the kinds of databases that the model uses to correlate its predictions can change and the model has to be able to handle the new data (sources). For this reason we chose object-oriented technology to implement the computer model and the database. The system was implemented using the GemStone OODBMS and the GemStone Smalltalk[4] Interface.

Integrating Experimental Data

Managing experimental data effectively is a challenging task. The reasons for this include:

[4] Objectworks\Smalltalk is a registered trademark of Parc Place Systems

- The complex structure associated with the data: Flat representations such as files and relations do not capture the logical structure of an experiment effectively.

- Volume: Many scientific experiments run for periods of hours, days or even months. During this time, data is collected using electronic data acquisition systems. Depending on the domain and the sampling rates, the raw data from individual runs can vary from a few kilobytes to tens, even hundreds of megabytes.

- The complexity of operations performed on data in the scientific domain: Most analyses of experimental data involve performing complex calculations on selected parts of the data. Hence there have to be effective ways to identify relevant parts of experimental data and also to specify complex computations on the data.

- The need for interoperability and sharing: Most experiments are conducted by small research teams who collaborate with other research groups, laboratories, funding agencies and industry. Hence sharing data and results between such groups is of prime importance. Also, data from such experiments is typically analyzed using a variety of supplemental tools such as plotting programs, statistical packages and sophisticated image analysis programs. Hence any scientific data manager has to have a high degree of flexibility in being able to directly interact with or produce output appropriate for these supplemental tools.

Our Approach

The goal of our project was to uniformly represent, store and analyze the parts of an experiment. Uniform representation involves managing the entire life cycle of an experiment including the experiment setup, equipment used, parameters varied, sensors used, the raw data generated by the experiment and subsequent analysis performed with the data. To accomplish this, the experiment is modeled as an *object*. The advantages of an integrated object-oriented approach include:

- Queries can be posed against all parts of the experiment in a uniform manner. The ability to pose such queries is a powerful facility, because it allows for associative access to the raw data based on a number of experimental conditions, and to collect outputs which can be used for subsequent analyses. For example, in a DTA experiment a query of the form

  ```
  return all phase transition points from experiments on Indium
  samples with different heating rates
  ```

  can be used to analyze the effect of heating rate on the melting point in a simple manner.

- Both structure (data) and behavior (functions that operate on the data) can be stored in the database. Users can browse and query experiments both based on data values and also using derived data computed using the functions. Hence the expressivity of queries is increased enormously. For example, in a DTA experiment a query of the form

```
return all samples where composition = 100 % indium and the
energy of transformation = x Cal/g
```

selects information from experiments based on both composition (data) and energy of transformation (a computed parameter).

- Queries can be posed on individual experiments, and also on collections of experiments. Queries over collections of experiments are especially useful in identifying correlations between the data and experiment conditions.

The Experiment Manager has been used to analyze data from laboratory Differential Thermal Analysis experiments. The experiment setup involves collecting data from a Mettler2000 DTA system using a data acquisition system. The manager records information about the sample, the DTA apparatus, and the data acquisition system in addition to the raw data. We are in the process of incorporating Transmission Electron Microscopy and Electroslag Remelting experiments into the experiment manager.

Conclusions

We have used object database technology to model, store and compute data in the materials science and thermodynamics domains. Prototype applications to demonstrate various features of the technology such as data modeling, information sharing, support for heterogenity and data analysis were designed and implemented.

Our object data model allows seamless access to multiple sources of data in diverse formats through a single powerful interface. Semi-empirical calculational programs which need efficient access to large volumes of data can benefit enormously from this technology. An object-oriented tool for managing data from laboratory experiments was developed. The tool provides storage support for the entire life cycle of an experiment and can be used for the integrated analysis of experiments.

Based on our experiences, we conclude that object database technology is eminently suited for managing data in the materials science domain.

Acknowledgements

Partial funding for this work was provided by the National Science Foundation grant number NSF 9117008. The first author acknowledges the guidance and support provided by Prof. David Atteridge of the Materials Science & Engineering department.

References

[1] Zdonik, S. and Maier, D., *Readings in Object-Oriented Database Systems*, Morgan Kaufman, 1990.

[2] Judith Bayard Cushing, David Hansen, David Maier, and Calton Pu, "Connecting scientific programs and data using object databases", *Bulletin of the Technical Committee on Data Engineering*, 16(1):9–13, March 1993.

[3] David M. Hansen and David Maier, "Using an object-oriented database to encapsulate heterogeneous scientific data sources", In Jay F. Nunamaker and Ralph H. Sprague, editors, *Proceedings of the Twenty-Seventh Annual*

Hawaii International Conference on System Sciences, volume III, pages 408 417, Maui, Hawaii, January 1994. The University of Hawaii, IEEE Computer Society Press.

[4] David Hansen, David Maier, James Stanley, and Jonathan Walpole, "An object-oriented heterogeneous database for materials science", *Scientific Programming*, 1(2):115 – 131, Winter 1992.

[5] Brewer, L., "Prediction of High Temperature Metalic Phase Diagrams", In V. F. Zackay, editor, *High Strength Materials*, New York, John Wiley, 1965.