Chapter 1

Overview

1.1 Background

This document describes the continuing work by members of the Object Data Management Group (ODMG) on specifications for persistence of object-oriented programming language objects in databases. The specification applies to two types of products: Object Database Management Systems (ODBMSs) that store objects directly and Object-to-Database Mappings (ODMs) that convert and store the objects in a relational or other database system representation. The two types of products will be referred to as object data management systems or ODMSs.

This document describes Release 3.0 of the ODMG specification, commonly known as ODMG 3.0, and is an enhancement to ODMG 2.0.

1.1.1 Importance of a Standard

Before the ODMG specification, the lack of a standard for storing objects in databases was a major limitation to object application portability across database systems. ODMG enables many vendors to support and endorse a common object interface to which customers write their database applications.

1.1.2 Goals

Our primary goal is to put forward a set of specifications allowing a developer to write portable applications, that is, applications that could run on more than one product. The data schema, programming language binding, and data manipulation and query languages must be portable. We are striving to bring programming languages and database systems to a new level of integration, moving the industry forward as a whole through the practical impetus of real products that conform to a comprehensive specification.

The ODMG member companies, representing almost the entire ODMS industry, are supporting this specification. Thus, our proposal has become a de facto standard for this industry. We have also used our specification in our work with standards efforts such as the Java Community Process, OMG, and the INCITS X3H2 (SQL) committee.

We do not wish to produce identical products. Our goal is source code portability; there is a lot of room for future innovation in a number of areas. There will be differences between products in performance, languages supported, functionality unique to particular market segments (e.g., version and configuration management), accompanying

programming environments, application construction tools, small versus large scale, multithreading, networking, platform availability, depth of functionality, suites of predefined type libraries, GUI builders, design tools, administration tools, and so on.

Wherever possible, we have used existing work as the basis for our proposals, from standards groups and from the literature. But, primarily, our work is derived by combining the strongest features of the products currently available. These products offer demonstrated implementations of our specifications that have been tried in the field.

1.1.3 Definition

It is important to define the scope of our efforts. An ODMS transparently integrates database capability with the application programming language. We define an *ODBMS* to be a DBMS that integrates database capabilities with object-oriented programming language capabilities. We define an *ODM* to be a system that integrates relational or other nonobject DBMSs with object-oriented programming language capabilities. ODMs include object-relational mapping products and object application servers. Either type of ODMS (ODBMS or ODM) makes database objects appear as programming language objects, in one or more existing programming languages. ODMSs extend the programming language with transparently persistent data, concurrency control, data recovery, associative queries, and other database capabilities. For more extensive definition and discussion of ODBMSs and ODMs, we refer you to textbooks in this area.

1.2 Major Components

The major components of ODMG 3.0 are described in subsequent chapters:

Object Model. The common data model to be supported by ODMG implementations is described in Chapter 2. We have used the OMG Object Model as the basis for our model. The OMG core model was designed to be a common denominator for object request brokers, object database systems, object programming languages, and other applications. In keeping with the OMG Architecture, we have designed an ODMS *profile* for their model, adding components (e.g., relationships) to the OMG core object model to support our needs.

Object Specification Languages. The specification languages for ODMSs are described in Chapter 3. The two described in this chapter are the Object Definition Language (ODL) and Object Interchange Format (OIF) languages. ODL is a specification language used to define the object types that conform to the ODMG Object Model. OIF is a specification language used to dump and load the current state of an ODMS to or from a file or set of files.

Object Query Language. We define a declarative (nonprocedural) language for querying and updating ODMS objects. This object query language, or OQL, is described in Chapter 4. We have used the relational standard SQL as the basis for OQL, where possible, though OQL supports more powerful capabilities.

C++ Language Binding. Chapter 5 presents the binding of ODMG implementations to C++; it explains how to write portable C++ code that manipulates persistent objects. This is called the C++ OML, or object manipulation language. The C++ binding also includes a version of the ODL that uses C++ syntax, a mechanism to invoke OQL, and procedures for operations on ODMSs and transactions.

Smalltalk Language Binding. Chapter 6 presents the binding of ODMG implementations to Smalltalk; it defines the binding in terms of the mapping between ODL and Smalltalk, which is based on the OMG Smalltalk binding for IDL. The Smalltalk binding also includes a mechanism to invoke OQL and procedures for operations on databases and transactions.

Java Language Binding. Chapter 7 defines the binding between the ODMG Object Model (ODL and OML) and the Java programming language as defined by the Java™ 2 Platform. The Java language binding also includes a mechanism to invoke OQL and procedures for operations on ODMSs and transactions. This chapter has been submitted to the Java Community Process as a basis for the Java Data Objects Specification.

It is possible to read and write the same database from C++, Smalltalk, and Java, as long as the programmer stays within the common subset of supported datatypes. More chapters may be added at a future date for other language bindings. Note that unlike SQL in relational systems, the ODMG data manipulation languages are tailored to specific application programming languages, in order to provide a single, integrated environment for programming and data manipulation. We don't believe exclusively in a universal DML syntax. We go further than relational systems, as we support a unified object model for sharing data across programming languages, as well as a common query language.

1.3 Participants

As of September 1999, the participants in the ODMG are

- Rick Cattell (ODMG chair, Release 1.0 editor), Sun Microsystems
- Jeff Eastman (ODMG vice-chair, Object Model workgroup chair, Smalltalk editor), Robert Hirschfeld, Windward Solutions
- Douglas Barry (ODMG executive director, Release 1.1, 1.2, 2.0, and 3.0 editor), Barry & Associates

- Mark Berler (Object Model and Object Specification Languages editor), American Management Systems
- Suad Alagic (invited staff), Wichita State University
- Jack Greenfield (invited staff), InLine Software
- François Bancilhon, Fernando Velez (OQL editor), voting member, Ardent Software
- Dirk Bartels, Olaf Schadow (C++ workgroup chair), voting member, POET Software
- David Jordan (C++ and Java editor), voting member, Ericsson
- Henry Parnell, voting member, Object Design
- Ron Raffensperger, voting member, Objectivity
- Craig Russell (Java workgroup chair), voting member, Sun Microsystems
- Henry Strickland, voting member, Versant Corporation
- Zaid Al-Timimi, reviewer member, Advanced Language Technologies
- Lougie Anderson, reviewer member, GemStone Systems
- Ole Anfindsen, reviewer member, Telenor R&D
- Tony Kamarainen, reviewer member, Lawson Software
- Yutaka Kimura, reviewer member, NEC
- Paul Lipton, reviewer member, Computer Associates
- Jon Reid, reviewer member, Micro Data Base Systems
- Jamie Shiers, reviewer member, CERN
- Torsten Stanienda (OQL workgroup chair), reviewer member, Baan
- Satoshi Wakayama, reviewer member, Hitachi

It is to the personal credit of all participants that the ODMG standard has been produced and revised expeditiously. All of the contributors put substantial time and personal investment into the meetings and this document. They showed remarkable dedication to our goals; no one attempted to twist the process to his or her company's advantage.

1.4 History and Status

Some of the history and methodology of ODMG may be helpful in understanding our work and the philosophy behind it. We learned a lot about how to make quick progress in standards in a new industry while avoiding "design by committee."

ODMG was conceived at the invitation of Rick Cattell in the summer of 1991, in an impromptu breakfast with ODBMS vendors frustrated at the lack of progress toward ODBMS standards. Our first meeting was at Sun Microsystems in the fall of 1991.

The group adopted rules that have been instrumental to our quick progress. We wanted to remain small and focused in our work, yet be open to all parties who are interested in our work. The structure evolved over time. Presently, we have established workgroups, one for each chapter of the specification. Each workgroup is intended to remain small, allowing for good discussion. The specifications adopted in each workgroup, however, must go before the ODMG Board for final approval. The Board usually holds open meetings for representatives from all members to attend and comment on our work.

We have worked outside of traditional standards bodies in order to make quick progress. Standards groups are well suited to incremental changes to a proposal once a good starting point has been established, but it is difficult to perform substantial creative work in such organizations due to their lack of continuity, large membership, and infrequent meetings. For our work, we have picked and combined the best features of implementations we had available to us.

The people who come to our meetings from our member companies are called Technical Representatives. They are required to have a technical background in our industry. We also have established rules requiring the same Technical Representatives come repeatedly to our meetings to maintain continuity of our work.

Voting membership is open to organizations that utilize or have announced utilization of the ODMG specification. Reviewer members are individuals or organizations having a direct and material interest in the work of the ODMG.

1.4.1 Accomplishments

Since the publication of Release 1.0, a number of activities have occurred.

1. Incorporation of the ODMG and the establishment of an office.
2. Affiliation with the Object Management Group (OMG), OMG adoption (February 1994) of a Persistence Service endorsing ODMG-93 as a standard interface for storing persistent state, and OMG adoption (May 1995) of a Query Service endorsing the ODMG OQL for querying OMG objects.
3. Establishment of liaisons with INCITS X3H2 (SQL), X3J16 (C++), and X3J20 (Smalltalk), and ongoing work between ODMG and X3H2 for converging OQL and SQL3 (now known as SQL-99).
4. Addition of reviewer membership to allow the user community to participate more fully in the efforts of the ODMG.
5. Publication of articles written by ODMG participants that explain the goals of the ODMG and how they will affect the industry.
6. Collection of feedback on Release 1.0, 1.1, 1.2, and 2.0, of which much was used in this release.

7. Co-submission of an OMG Persistent State Service with IONA, Inprise, and others, which incorporates an ODMG transparent persistence option. This submission was accepted by the OMG's Platform Technology Committee in November, 1999.
8. Submittal of the ODMG Java Binding to the Java Community Process as a basis for the Java Data Objects Specification.

1.4.2 Next Steps

We now plan to proceed with several actions in parallel to keep things moving quickly.

1. Distribute Release 3.0 through this book.
2. Complete implementation of the specifications in our respective products.
3. Collect feedback and corrections for the next release of our standards specification.
4. Continue to maintain and develop our work.
5. Continue to submit our work to the Java Community Process, OMG, or INCITS, as appropriate.

1.4.3 Suggestion and Proposal Process

If you have suggestions for improvements in future versions of our document, we welcome your input. We recommend that change proposals be submitted as follows:

1. State the essence of your proposal.
2. Outline the motivation and any pros/cons for the change.
3. State exactly what edits should be made to the text, referring to page number, section number, and paragraph.
4. Send your proposal to *proposal@odmg.org*.

1.4.4 Contact Information

If you have questions on ODMG 3.0, send them to *question@odmg.org*.

If you have additional questions, or if you want membership information for the ODMG, please contact ODMG's executive director, Douglas Barry, at *dbarry@odmg.org*, or contact

Object Data Management Group
13504 4th Avenue South
Burnsville, MN 55337 USA
Voice:+1-612-953-7250
Fax: +1-612-397-7146
Email:*info@odmg.org*
Web: *www.odmg.org*

1.4.5 Related Standards

There are references in this book to INCITS X3 documents, including SQL specifications (X3H2), Object Information Management (X3H7), the X3/SPARC/DBSSG OODB Task Group Report (contact *fong@ecs.ncsl.nist.gov*), and the C++ standard (X3J16). INCITS documents can be obtained from

X3 Secretariat, CBEMA
1250 Eye Street, NW, Suite 200
Washington, DC 20005-3922 USA

There are also references to Object Management Group (OMG) specifications, from the Object Request Broker (ORB) Task Force (also called CORBA), the Object Model Task Force (OMTF), and the Object Services Task Force (OSTF). OMG can be contacted at

Object Management Group
Framingham Corporate Center
492 Old Connecticut Path
Framingham, MA 01701 USA
Voice:+1-508-820-4300
Fax: +1-508-820-4303
Email: *omg@omg.org*
Web: *www.omg.org*

The Java Community Process is the formalization of the open process that Sun Microsystems, Inc. has been using since 1995 to develop and revise Java technology specifications in cooperation with the international Java community. The Java Community Process can be found at

Web: *java.sun.com/aboutJava/communityprocess/*