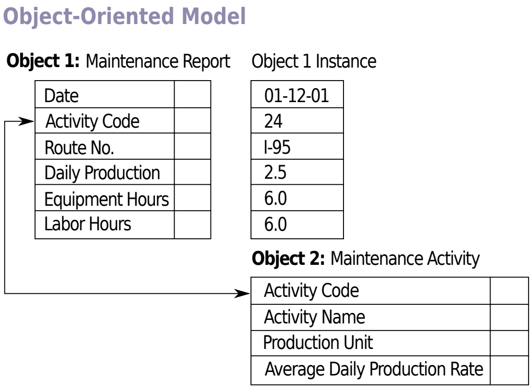
Object database



Example of an object-oriented model[1]

An **object database** (also **object-oriented database management system**) is a database management system in which information is represented in the form of objects as used in object-oriented programming. Object databases are different from relational databases which are table-oriented. Object-relational databases are a hybrid of both approaches.

Object databases have been considered since the early 1980s.[2]

1 Overview

Object-oriented database management systems (OODBMSs) combine database capabilities with object-oriented programming language capabilities. OODBMSs allow object-oriented programmers to develop the product, store them as objects, and replicate or modify existing objects to make new objects within the OODBMS. Because the database is integrated with the programming language, the programmer can maintain consistency within one environment, in that both the OODBMS and the programming language will use the same model of representation. Relational DBMS projects, by way of contrast, maintain a clearer division between the database model and the application.

As the usage of web-based technology increases with the implementation of Intranets and extranets, companies have a vested interest in OODBMSs to display their complex data. Using a DBMS that has been specifically designed to store data as objects gives an advantage to those companies that are geared towards multimedia presentation or organizations that utilize computer-aided design (CAD).[3]

Some object-oriented databases are designed to work well with object-oriented programming languages such as Delphi, Ruby, Python, Perl, Java, C#, Visual Basic .NET, C++, Objective-C and Smalltalk; others have their own programming languages. OODBMSs use exactly the same model as object-oriented programming languages.

2 History

Object database management systems grew out of research during the early to mid-1970s into having intrinsic database management support for graph-structured objects. The term “object-oriented database system” first appeared around 1985.[4] Notable research projects included Encore-Ob/Server (Brown University), EXODUS (University of Wisconsin–Madison), IRIS (Hewlett-Packard), ODE (Bell Labs), ORION (Microelectronics and Computer Technology Corporation or MCC), Vodak (GMD-IPSI), and Zeitgeist (Texas Instruments). The ORION project had more published papers than any of the other efforts. Won Kim of MCC compiled the best of those papers in a book published by The MIT Press.[5]

Early commercial products included Gemstone (Servio Logic, name changed to GemStone Systems), Gbase (Graphael), and Vbase (Ontologic). The early to mid-1990s saw additional commercial products enter the market. These included ITASCA (Itasca Systems), Jasmine (Fujitsu, marketed by Computer Associates), Matisse (Matisse Software), Objectivity/DB (Objectivity, Inc.), ObjectStore (Progress Software, acquired from eXcelon which was originally Object Design), ONTOS (Ontos, Inc., name changed from Ontologic), O_2[6] (O_2 Technology, merged with several companies, acquired by Informix, which was in turn acquired by IBM), POET (now FastObjects from Versant which acquired Poet Software), Versant Object Database (Versant Corporation), VOSS (Logic Arts) and JADE (Jade Software Corporation). Some of these products remain on the market and have been joined by new open source and commercial products such as InterSystems Caché.

Object database management systems added the concept of persistence to object programming languages. The early commercial products were integrated with various languages: GemStone (Smalltalk), Gbase (LISP), Vbase (COP) and VOSS (Virtual Object Storage System for Smalltalk). For much of the 1990s, C++ dominated the commercial object database management market. Vendors added Java in the late 1990s and more recently, C#.

Starting in 2004, object databases have seen a second growth period when open source object databases emerged that were widely affordable and easy to use, because they are entirely written in OOP languages like Smalltalk, Java, or C#, such as Versant's db4o (db4objects), DTS/S1 from Obsidian Dynamics and Perst (McObject), available under dual open source and commercial licensing.

3 Timeline

- 1966
  - MUMPS
- 1979
  - InterSystems M
- 1980
  - TORNADO – an object database for CAD/CAM[7]
- 1982
  - Gemstone started (as Servio Logic) to build a set theoretic model data base machine.
- 1985 – Term Object Database first introduced
- 1986
  - Servio Logic (Gemstone Systems) Ships Gemstone 1.0
- 1988
  - Versant Corporation started (as Object Sciences Corp)
  - Objectivity, Inc. founded
- Early 1990s
  - Servio Logic changes name to Gemstone Systems
  - Gemstone (Smalltalk)-(C++)-(Java)
  - GBase (LISP)
  - VBase (O2- ONTOS – INFORMIX)
  - Objectivity/DB
- Mid 1990's
  - InterSystems Caché
  - Versant Object Database
  - ObjectStore
  - ODABA
  - ZODB
  - Poet
  - Jade
  - Matisse
  - Illustra Informix
  - Webcrossing
- 2000's
  - db4o project started by Carl Rosenberger
  - ObjectDB
- 2001 IBM acquires Informix
- 2003 odbpp public release
- 2004 db4o's commercial launch as db4objects, Inc.
- 2008 db4o acquired by Versant Corporation
- 2010 VMware acquires GemStone[8]
- 2012 Wakanda first production versions with open source and commercial licenses
- 2013 GemTalk Systems acquires GemStone products from VMware[9]
- 2014 Realm

4 Adoption of object databases

Object databases based on persistent programming acquired a niche in application areas such as engineering and spatial databases, telecommunications, and scientific areas such as high energy physics and molecular biology.

Another group of object databases focuses on embedded use in devices, packaged software, and real-time systems.

5 Technical features

Most object databases also offer some kind of query language, allowing objects to be found using a declarative programming approach. It is in the area of object query languages, and the integration of the query and navigational interfaces, that the biggest differences between products are found. An attempt at standardization was made by the ODMG with the Object Query Language, OQL.

Access to data can be faster because joins are often not needed (as in a tabular implementation of a relational database). This is because an object can be retrieved directly without a search, by following pointers.

Another area of variation between products is in the way that the schema of a database is defined. A general characteristic, however, is that the programming language and the database schema use the same type definitions.

Multimedia applications are facilitated because the class methods associated with the data are responsible for its correct interpretation.

Many object databases, for example Gemstone or VOSS, offer support for versioning. An object can be viewed as the set of all its versions. Also, object versions can be treated as objects in their own right. Some object databases also provide systematic support for triggers and constraints which are the basis of active databases.

The efficiency of such a database is also greatly improved in areas which demand massive amounts of data about one item. For example, a banking institution could get the user's account information and provide them efficiently with extensive information such as transactions, account information entries etc. The Big O Notation for such a database paradigm drops from O(n) to O(1), greatly increasing efficiency in these specific cases.

6 Standards

The Object Data Management Group was a consortium of object database and object-relational mapping vendors, members of the academic community, and interested parties. Its goal was to create a set of specifications that would allow for portable applications that store objects in database management systems. It published several versions of its specification. The last release was ODMG 3.0. By 2001, most of the major object database and object-relational mapping vendors claimed conformance to the ODMG Java Language Binding. Compliance to the other components of the specification was mixed. In 2001, the ODMG Java Language Binding was submitted to the Java Community Process as a basis for the Java Data Objects specification. The ODMG member companies then decided to concentrate their efforts on the Java Data Objects specification. As a result, the ODMG disbanded in 2001.

Many object database ideas were also absorbed into SQL:1999 and have been implemented in varying degrees in object-relational database products.

In 2005 Cook, Rai, and Rosenberger proposed to drop all standardization efforts to introduce additional object-oriented query APIs but rather use the OO programming language itself, i.e., Java and .NET, to express queries. As a result, Native Queries emerged. Similarly, Microsoft announced Language Integrated Query (LINQ) and DLINQ, an implementation of LINQ, in September 2005, to provide close, language-integrated database query capabilities with its programming languages C# and VB.NET 9.

In February 2006, the Object Management Group (OMG) announced that they had been granted the right to develop new specifications based on the ODMG 3.0 specification and the formation of the Object Database Technology Working Group (ODBT WG). The ODBT WG planned to create a set of standards that would incorporate advances in object database technology (e.g., replication), data management (e.g., spatial indexing), and data formats (e.g., XML) and to include new features into these standards that support domains where object databases are being adopted (e.g., real-time systems). The work of the ODBT WG was suspended in March 2009 when, subsequent to the economic turmoil in late 2008, the ODB vendors involved in this effort decided to focus their resources elsewhere.

In January 2007 the World Wide Web Consortium gave final recommendation status to the XQuery language. XQuery uses XML as its data model. Some of the ideas developed originally for object databases found their way into XQuery, but XQuery is not intrinsically object-oriented. Because of the popularity of XML, XQuery engines compete with object databases as a vehicle for storage of data that is too complex or variable to hold conveniently in a relational database. XQuery also allows modules to be written to provide encapsulation features that have been provided by Object-Oriented systems.

7 Comparison with RDBMSs

An object database stores complex data and relationships between data directly, without mapping to relational rows and columns, and this makes them suitable for applications dealing with very complex data.[10] Objects have a many to many relationship and are accessed by the use of pointers. Pointers are linked to objects to establish relationships. Another benefit of an OODBMS is that it can be programmed with small procedural differences without affecting the entire system.[11]

8 See also

- Comparison of object database management systems
- Component-oriented database
- EDA database
- Enterprise Objects Framework
- NoSQL
- Object Data Management Group
- Object-relational database
- Persistence (computer science)
- Relational model

9 References

[1] Data Integration Glossary, U.S. Department of Transportation, August 2001.

[2] ODBMS.ORG :: Object Database (ODBMS) | Object-Oriented Database (OODBMS) | Free Resource Portal. ODBMS (2013-08-31). Retrieved on 2013-09-18.

[3] O'Brien, J. A., & Marakas, G. M. (2009). Management Information Systems (9th ed.). New York, NY: McGraw-Hill/Irwin

[4] Three example references from 1985 that use the term: T. Atwood, "An Object-Oriented DBMS for Design Support Applications," *Proceedings of the IEEE COMPINT 85*, pp. 299-307, September 1985; N. Derrett, W. Kent, and P. Lyngbaek, "Some Aspects of Operations in an Object-Oriented Database," *Database Engineering*, vol. 8, no. 4, IEEE Computer Society, December 1985; D. Maier, A. Otis, and A. Purdy, "Object-Oriented Database Development at Servio Logic," *Database Engineering*, vol. 18, no.4, December 1985.

[5] Kim, Won. *Introduction to Object-Oriented Databases.* The MIT Press, 1990. ISBN 0-262-11124-1

[6] Bancilhon, Francois; Delobel,Claude; and Kanellakis, Paris. *Building an Object-Oriented Database System: The Story of O_2*. Morgan Kaufmann Publishers, 1992. ISBN 1-55860-169-4.

[7] Ulfsby et al. (July 1981). "TORNADO: a DBMS for CAD/CAM systems". *Computer-Aided Design* **13** (4): 193–197.

[8] "SpringSource to Acquire Gemstone Systems Data Management Technology". WMware. May 6, 2010. Retrieved August 5, 2014.

[9] GemTalk Systems (May 2, 2013). "GemTalk Systems Acquires GemStone/S Products from VMware". PRWeb. Retrieved August 5, 2014.

[10] Radding, Alan (1995). "So what the Hell is ODBMS?". *Computerworld* **29** (45): 121–122, 129.

[11] Burleson, Donald. (1994). OODBMSs gaining MIS ground but RDBMSs still own the road. Software Magazine, 14(11), 63

10 External links

- Object DBMS resource portal
- Object-Oriented Databases – From CompTechDoc.org

11 Text and image sources, contributors, and licenses

11.1 Text

- **Object database** *Source:* https://en.wikipedia.org/wiki/Object_database?oldid=671531092 *Contributors:* Vtan, Hari, Ben-Zin~enwiki, Maury Markowitz, Leandrod, Stevertigo, W~enwiki, Modster, Kku, Pcb21, CesarB, Hofoen, Ronz, Rednblu, Furrykef, Robbot, Noldoaran, BenFrantzDale, Beardo, Mckaysalisbury, Gadfium, Beland, SimonArlott, Sam Hocevar, Indolering, Usrnme h8er, Klemen Kocjancic, Tordek ar, Pavel Vozenilek, Elwikipedista~enwiki, Enric Naval, Ejrrjs, Mdd, Tablizer, Zippanova, Rickyp, SteinbDJ, Kinema, Forderud, Karnesky, Mindmatrix, Dandv, Timosa, Ruud Koot, JIP, Icey, Jivecat, Dmccreary, FlaBot, SchuminWeb, Margosbot~enwiki, Intgr, Bgwhite, YurikBot, Foxxygirltamara, Hydrargyrum, ENeville, Oberst, SAE1962, Larsinio, Voidxor, Grafikm fr, BOT-Superzerocool, Ott2, Sandstein, BSTRhino, Talyian, Wainstead, Mhkay, Mlibby, Dybdahl, SmackBot, Pintman, Kellen, Reedy, OEP, AutumnSnow, Commander Keane bot, Bluebot, Nkaku, MyNameIsVlad, RProgrammer, Turbothy, Soumyasch, Lguzenda, IronGargoyle, RichardF, DougBarry, Hu12, Britannica~enwiki, Tawkerbot2, DBooth, FatalError, LGuzenda, Ervinn, Shreyasjoshis, Singerboi22, Yaris678, Corpx, Sestoft, Torc2, Charwing, Nick Number, Mvjs, Spencer, MER-C, Cameltrader, Jbom1, Magioladitis, Hroðulf, Soulbot, 28421u2232nfenfcenc, Gwern, Wixardy, J.delanoy, Saifali1, Kozka, Tagus, VolkovBot, Rei-bot, WikipedianYknOK, Dawn Bard, JCLately, SmallRepair, Edlich, PsyberS, Startswithj, Dinojc, ClueBot, The Thing That Should Not Be, EoGuy, Alexbot, Eeekster, Sun Creator, Lucpeuvrier~enwiki, EastTN, HarlandQPitt, Zeliboba7, Addbot, Download, Kngspook, Lightbot, Jackelfive, Yobot, Bunnyhop11, Fraggle81, Waynenilsen, AnomieBOT, Materialscientist, Xqbot, Mika au, Addbc, Garyaj, Pwwamic, FrescoBot, Sulfsby, Mark Renier, Bablind, ITOntheMind, Shewizzy2005, Cari.tenang, Maria Johansson, Alexandre.Morgaut, Gf uip, EmausBot, John of Reading, WikitanvirBot, Booler80, Minimac's Clone, Phiarc, SvetCo, Germanviscuso, ClueBot NG, Ki2010, Danim, Secured128, Razorbliss, Compfreak7, Pradiq009, Matspca, Snow Blizzard, Osiris, Eduardofeld, Khiladi 2010, FlyingPhysicist, André Miranda Moreira, DallasClarke, Rzicari and Anonymous: 210

11.2 Images

- **File:Object-Oriented_Model.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/7/7c/Object-Oriented_Model.svg *License:* Public domain *Contributors:* Data Integration Glossary. *Original artist:*
  - U.S. Department of Transportation
  - vectorization: Own work

11.3 Content license

- Creative Commons Attribution-Share Alike 3.0