SIGMOD INSTITUTIONAL SPONSORS†

Digital Equipment Corporation
1175 Chapel Hills Drive
Colorado Springs, CO 80920-3952

Electronics and Telecommunications Research Institute
Post Office Box 8, Daedog
Danji Daejoen
Korea 305-606

Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304

IBM Almaden Research Center
K55/802 - 650 Harry Road
San Jose, CA 95120-6099

Lotus Development Corporation
One Rogers Street
Cambridge, MA 02141

Microelectronics & Computer Technology Corporation (MCC)
3500 West Balcones Center Drive
Austin, TX 78759

Mitsubishi Electric Company
Computer Works
325 Kamimachiya
Kamakura, Kanagawa
247 Japan

† Benefits of sponsorship include two subscriptions to the SIGMOD Newsletter "SIGMOD RECORD" and the privilege of sending employees of your organization to conferences and workshops sponsored by SIGMOD at reduced member rates. In addition, your organization will be listed in the SIGMOD Newsletter as an Institutional Sponsor.
Editor’s Notes

The regular articles section includes three contributions. The first paper by M. Rafanelli and F.L. Ricci presents a new functional model for statistical macro data. The short note by John T. Robinson deals with the performance analysis of the Essential Blocking Recurrence concurrency control method, and the Minimal Covers problem is revisited in Jim Diederich’s note.

This issue contains two excellent bibliographies, the first on Temporal Databases by Michael D. Soo, and the second on Object-Oriented Database Management by Gottfried Vossen. Readers are invited to contact the authors if they know of additional references or to receive updated versions (note that the bibliography on Temporal Databases is available online). Following these bibliographies there is a report by Satish Thatte which provides information on the U.S. Army’s contract awarded to Texas Instruments Inc. to develop a Modular and Open Object-Oriented Database System.

The regular section on Database Research Centers presents a report on the Altair Database Project in France, and the the issue is concluded with calls for papers and participation including a summary of upcoming database conferences prepared by Felix Saltor.

Arie Segev
January, 1991
1 Introduction

This is the fourth bibliography to appear concerning time in databases. It is a supplement to the 1988 bibliography by Stam and Snodgrass ("A Bibliography on Temporal Databases," IEEE Database Engineering, 7, 1988, pp. 231-239), which supplemented the 1986 survey by McKenzie ("Bibliography: Temporal Databases," ACM SIGMOD Record, 15, No. 4, Dec. 1986, pp. 40-52), which was in turn a supplement to a 1982 survey (Bolour, A., T.L. Anderson, and H.K.T. Wong, "The Role of Time in Information Processing: A Survey," SIGART Newsletter, 80, Apr. 1982, pp. 28-48). This bibliography identifies research published since the 1988 survey, as well as older papers which did not appear in the previous collections.

This bibliography concentrates on time-related research in databases, particularly efforts to support the representation of time-varying information in database management systems. Interest in this field has been and continues to be strong as evidenced by the number of researchers in the area. The approximate size of the temporal database research community over time is shown in Figure 1. The graph was derived by counting, for each point in time, the number of people who had authored a paper prior to that point in time.

The entries are classified according to a taxonomy of time in databases developed by Snodgrass and Ahn ("Temporal Databases," IEEE Computer, 19, No. 9, Sep. 1986, pp. 35-42). Papers that propose augmenting conventional database management systems to represent transaction time (that is, the time when information is entered into a database) are listed in Section 2. (53 papers are listed in Section 2.) Papers that propose augmenting conventional database management systems to represent valid time (that is, the time that information models in the real world) are listed in Section 3. (83 papers are listed in Section 3.) Section 4 lists papers describing temporal databases i.e., database systems supporting both aspects of time. (11 papers are listed in Section 4.) The final section lists additional papers concerning time which lie outside of the above classification.

This bibliography is incremental. The previous bibliographies must also be consulted to gather all papers on temporal databases published to date. For a brief summary of research in temporal databases, see "Temporal Databases: Status and Research Directions," ACM SIGMOD Record, 19, No. 4, Dec. 1990, pp. 83-89, by Richard Snodgrass. The entries in this and the previous bibliographies are available online for anonymous ftp from node cs.arizona.edu in the subdirectory named bib.

Figure 1: Size of Temporal DB Research Community.
We wish to thank the many people who reviewed drafts of this bibliography and supplied us with additional references. We apologize in advance for any omissions or misclassifications and would appreciate receiving corrections and additions.

2 Transaction Time

A relation incorporating transaction time may be rolled back to a previous state in time. Such a relation is append-only.


3 Valid Time

A relation incorporating valid time records a history of the enterprise being modeled, as is currently known. Such a relation can be modified as errors are identified and as history progresses.


4 Both Transaction and Valid Time

A relation incorporating both kinds of time supports rollback, historical queries. Errors can be corrected yet the relation is still append-only.


5 Other Papers Dealing with Time

Most of the papers in this section deal with temporal reasoning or object histories.


Introduction

This is a bibliography on object-orientation in databases and object-oriented database systems, a field in which the literature has been exploding over the past several years. I compiled it primarily for my personal usage, but hopefully it is also useful for other people. The list contains papers which appeared since 1981 and are generally accessible (so that only very few entries represent "grey" literature). It does not claim to be exhaustive; in particular, it does not contain entries for work on nested relations, complex objects or, more generally, purely structural object-orientation (which I collect separately), nor does it contain papers on object-oriented design, programming, and programming languages.

The division of the material into sections follows my personal perception of the field. For simplicity, each section is sorted alphabetically. Several papers are repeated in more than one section, since they deal, for example, with a general topic as well as with a specific system. However, papers focusing on a particular system are always listed in the corresponding subsection of Section 5 in the first place.

The bibliography currently contains 492 entries (where repetitions are counted separately), and regularly covers the following journals and proceedings (beginning with whenever these started to publish relevant work):

- ACM Transactions on Database Systems (TODS)
- ACM SIGMOD Record
- IEEE Transactions on Software Engineering (TSE)
- IEEE Transactions on Knowledge and Data Engineering (TKDE)
- IEEE Bulletin of the Computer Society Technical Committee on Data(base) Engineering
- Springer LNCS, including Proc. International Conference on Database Theory (ICDT)
- Proc. Int. Conf. on Extending Database Technology (EDBT)
- Proc. Int. Conf. on Foundations of Data Organization (FODO)
- Proc. ACM SIGMOD International Conference on Management of Data
- Proc. ACM Symposium on Principles of Database Systems (PODS)
- Proc. ACM Conf. on Object-Oriented Programming Systems, Languages and Applications (OOPSLA)
- Proc. IEEE Int. Data Engineering Conf. (DEC)
- Proc. International Conference on Very Large Data Bases (VLDB)

In addition, many papers which appeared elsewhere (like in Proc. German GI Conference on "Datenbanken für Büro, Technik und Wissenschaft" (BTW), in DOOD\(^1\), PARBASE\(^2\) or DEXA\(^3\) Proceedings) are listed.

Remarks, corrections, updates, additions etc. are most welcome. Readers who are interested in customizing the list for their own purposes can obtain a

\(^1\)Proc. Int. Conf. on Deductive and Object-Oriented Databases
\(^2\)IEEE Int. Conf. on Databases, Parallel Architectures, and Their Applications
\(^3\)Int. Conf. on Data Base and Expert Systems Applications
copy of the file, which is in \LaTeX\ format, by sending me electronic mail.

Table of Contents

1 General Literature
   1.1 Books and Paper Collections
   1.2 Chapters in Books
   1.3 Survey Articles
   1.4 Workshop Reports and Reviews
   1.5 Other

2 Fundamental Issues
   2.1 Data Models for Objects
   2.2 Object Identity
   2.3 Operations on Objects
   2.4 Modeling Behavior
   2.5 Query Processing and Optimization
   2.6 Design of Objectbases

3 Object-Oriented Database Programming Languages
   3.1 Type Systems and Inheritance
   3.2 Persistence
   3.3 Impedance Mismatch
   3.4 Generalized Logic Programming
   3.5 Tools for Layered DBS Development
   3.6 Specific Languages
   3.7 Other Relevant Work

4 Commercially Available Systems
   4.1 GemStone
   4.2 Ontos/Vbase
   4.3 VERSANT ODBMS
   4.4 VISION

5 Experimental Systems
   5.1 Alltalk
   5.2 CACTIS
   5.3 DAMOKLES
   5.4 ENCORE/Observer
   5.5 EXODUS and derived systems
   5.6 HiPAC
   5.7 Iris
   5.8 LOGRES
   5.9 Ode
   5.10 O₂
   5.11 ORION
   5.12 OZ+
   5.13 POSTGRES
   5.14 PROBE
   5.15 RAD
   5.16 R²D²
   5.17 ROSE
   5.18 3DIS
   5.19 VODAK
   5.20 ZEITGEIST
   5.21 Others

6 Implementation Aspects
   6.1 Storage Organization
   6.2 Concurrency Control and Recovery
   6.3 Performance Evaluation

7 Miscellaneous
1 General Literature

1.1 Books and Paper Collections

1. F. Bancilhon, P. Buneman (eds.): *Advances in Database Programming Languages*; Addison-Wesley (ACM Press) 1990

2. W. Kim: *Introduction to Object-Oriented Databases*; MIT Press, to appear


4. F. Lochovsky (ed.): *Special Issue on Object-Oriented Systems*; *IEEE Database Engineering Bulletin* 8 (4) 1985

5. S.B. Zdonik, D. Maier (eds.): *Readings in Object-Oriented Database Systems*; Morgan Kaufmann Publishers 1990

1.2 Chapters in Books

1. C.J. Date: *An Introduction to Database Systems* Vol.1; 5th edition, Addison-Wesley 1990, Chapter 25

2. S. Khoshafian, R. Abnous: *Object Orientation — Concepts, Languages, Databases, User Interfaces*; Wiley 1990, Chapter 7

3. E. Ozkarahan: *Database Management*; Prentice-Hall 1990, Chapter 11


6. G. Vossen: *Data Models, Database Languages and Database Management Systems*; Addison-Wesley 1991, Chapter 22

7. A.L. Winblad et al.: *Object-Oriented Software*; Addison-Wesley 1990, Chapter 5

1.3 Survey Articles


2. F. Bancilhon: Object-Oriented Database Systems; Proc. 7th ACM PODS 1988, 152–162


4. The Committee for Advanced DBMS Function: Third-Generation Database System Manifesto; ACM SIGMOD Record 19 (3) 1990, 31–44


7. R. Hull, R. Morrison, D. Stemple: Current Directions in Database Programming Languages; *IEEE Data Engineering Bulletin* 12 (3) 1989, 3–4


10. W. Kim: Object-Oriented Databases: Definition and Research Directions; *IEEE TKDE* 2, 1990, 327–341


12. C.M. Stone, D. Hentschel: Database Wars Revisited; Byte October 1990, 233–242


1.4 Workshop Reports and Reviews


26 SIGMOD RECORD, Vol. 20, No. 1, March 1991

3. H. Kilov: Reviews of Object-Oriented Papers; ACM SIGMOD Record 18 (1) 1989, 12–15

4. H. Kilov: Reviews of Object-Oriented Papers, 2; ACM SIGMOD Record 18 (4) 1989, 50–55

5. The Laguna Beach Participants: Future Directions in DBMS Research; ACM SIGMOD Record 18 (1) 1989, 17–26

6. R.P. van de Riet: Introduction to the Special Issue on deductive and object-oriented databases; Data & Knowledge Engineering 5, 1990, 255–261

7. S.M. Thatte: Report on The Object-Oriented Database Workshop: Implementation Aspects; ACM SIGMOD Record 17 (2) 1988, 95–107

1.5 Other

1. G. Booch: Object-Oriented Development; IEEE TSE 12, 1986, 211–221


5. N. Prabhakaran: The Outstanding Problem for Today's Database Technology; Proc. PARBASE-90, 268–273


7. A. Snyder: Encapsulation and Inheritance in Object-Oriented Programming Languages; Proc. 1st OOPSLA 1986, 38–45


11. J.D. Ullman: Database Theory — Past and Future; Proc. 6th ACM PODS 1987, 1–10


2 Fundamental Issues

2.1 Data Models for Objects


5. C. Beeri: A formal approach to object-oriented databases; Data & Knowledge Engineering 5, 1990, 353–382


11. S. Hong, F. Maryanski: Using a Meta Model to Represent Object-Oriented Data Models; Proc. 6th IEEE DEC 1990, 11-19
28. R. Zicari: Incomplete Information in Object-Oriented Databases; ACM SIGMOD Record 19 (3) 1990, 5-16

2.2 Object Identity


2.3 Operations on Objects


2.4 Modeling Behavior

1. S. Abiteboul, P.C. Kanellakis, E. Waller: Method Schemas; Proc. 9th ACM PODS 1990, 16–27


2.5 Query Processing and Optimization


7. G. Graefe, D. Maier, S. Daniels, T. Keller: A Software Architecture for Efficient Query Processing in Object-Oriented Database Systems with Encapsulated Behavior; manuscript, April 1990


19. S.B. Zdonik: Data Abstraction and Query Optimization; Proc. 2nd Worksh. on OODBS 1988, Springer LNCS 334, 368-373

2.6 Design of Objectbases


8. L. Cardelli: Types for Data-Oriented Languages; Proc. 1st EDBT 1988, Springer LNCS 303, 1–15


10. R. Connor et al.: Existentially Quantified Types as a Database Viewing Mechanism; Proc. 2nd EDBT 1990, Springer LNCS 416, 301–315


17. A. Ohori, P. Buneman: Static Type Inference for Parametric Classes; Proc. 4th OOPSLA 1989, 445–456


19. L. Tan, T. Katayama: Meta Operations for Type Management in Object-Oriented Databases; Proc. 1st DOOD 1989, 58–75

3.2 Persistence


9. D. Maier: Representing Database Programs as Objects; in: F. Bancilhon, P. Buneman (eds.): Advances in Database Programming Languages; Addison-Wesley (ACM Press) 1990, 377–386

11. S.B. Zdonik: Object-Oriented Type Evolution; in: F. Bancilhon, P. Buneman (eds.): Advances in Database Programming Languages; Addison-Wesley (ACM Press) 1990, 277–288

3.3 Impedance Mismatch


3.4 Generalized Logic Programming


5. F. Bancilhon: A Logic-Programming/ Object-Oriented Cocktail; ACM SIGMOD Record 15 (3) 1986, 11–21

6. F. Cacace et al.: Integrating Object-Oriented Data Modeling with a Rule-Based Programming Paradigm; Proc. ACM SIGMOD 1990, 225–236

7. Y. Caseau: A Formal System for Producing Demons from Rules in an Object-Oriented Database; Proc. 1st DOOD 1989, 188–204


19. M. Kifer, J. Wu: A Logic for Object-Oriented Logic Programming (Maier’s O-Logic Revisited); Proc. 8th ACM PODS 1989, 379–393


22. D. Maier: A Logic for Objects; OGC Techn. Report CS/E-86-012, Oregon Graduate Center 1986

23. S. Naqvi: How Stratification is Used in LDL; IEEE Data Engineering Bulletin 12 (3) 1989, 45–52


3.5 Tools for Layered DBS Development


3.6 Specific Languages

3.6.1 ADABTBL


3.6.2 Galileo


3. A. Albano et al.: A Strongly Typed, Interactive Object-Oriented Database Programming Language; Proc. 1st Workshop on OODBS 1986, 94–103


3.6.3 Trellis/Owl

1. P. O'Brien et al.: Persistent and Shared Objects in Trellis/Owl; Proc. 1st Worksh. on OODBS 1986, 113–123


5. C. Schaffert et al.: An Introduction to Trellis/Owl; Proc. 1st OOPSLA 1986, 9–16

3.6.4 Others


3. F. Bancilhon et al.: FAD, a Powerful and Simple Database Language; Proc. 13th VLDB 1987, 97–105


3.7 Other Relevant Work


3. A. Heuer, J. van den Bussche: Using SQL with Object-Oriented Databases; Informatik-Bericht 90/3, TU Clausthal, 103–121


4 Commercially Available Systems

4.1 GemStone (Servio Logic)

1. Ullman (1988) (see 1.2, No. 5), Sections 5.6, 5.7

2. Vossen (1991) (see 1.2, No. 6), Section 22.2


5. G. Copeland, D. Maier: Making Smalltalk a Database System; Proc. ACM SIGMOD 1984, 316–325


12. Servio Logic Corp.: GemStone Product Overview; Beaverton, OR, 1988


15. Servio Logic Corp.: Programming in OPAL; Beaverton, OR, 1988

16. Servio Logic Corp.: GemStone Product Overview; Alameda, CA 1990

17. Servio Logic Corp.: GemStone Tools; Alameda, CA, 1990


4.2 Ontos/Vbase (Ontologic)


2. T. Andrews, C. Harris, K. Sinkel: The Ontos Object Database; manuscript 1989


5. Ontologic Inc.: Ontos Object Database SQL Command Reference; 1989


7. Ontologic Inc.: Ontos Object Database Client Library Reference; 1989

8. Ontologic Inc.: Vbase Technical Overview; Version 1.0, 1987

9. Ontologic Inc.: Vbase Product Overview; 1987

10. Ontologic Inc.: Vbase Guided Tour Tutorial; 1987


18. Ontologic Inc.: Vbase Application Notes (Concurrency Control, Schema Modification and Data Migration, Version 0.93 Release Note); 1987
19. Ontologic Inc.: Ontos Release 2.0 Product Description; 1990

4.3 VERSANT ODBMS (Versant)
1. Versant Object Technology Corp.: Product Profile; Menlo Park, CA, 1990

4.4 VISION (Innovative Systems)
3. E. Sciore: Object Specialization; ACM TOIS 7, 1989, 103–122

5 Experimental Systems

5.1 Alltalk

5.2 CACTIS
2. R. King: A Database Management System Based on an Object-Oriented Model; in: L. Kerschberg (ed.): Expert Database Systems; Benjamin/Cummings 1986, 443–468
4. S.E. Hudson, R. King: Object-Oriented Support for Software Environments; Proc. ACM SIGMOD 1987, 491–503
5. S.E. Hudson, R. King: An Adaptive Derived Data Manager for Distributed Databases; Proc. 2nd Worksh. on OODBS 1988, Springer LNCS 334, 193–203

5.3 DAMOKLES
5. B. Schiefer, S. Rehm: Eine Anfragesprache für ein strukturell-objektorientiertes Datenmodell; Proc. BTW 89, Springer IFB 204, 373–388

5.4 ENCORE/Observer
5.5 EXODUS and derived systems

1. A. Biliris: Modeling Design Object Relationships in PEGASUS; Proc. 6th DEC 1990, 228-236
2. M. Carey, D. DeWitt: An Overview of the EXODUS Project; IEEE Database Engineering Bulletin 10 (2) 1987, 47-54
5.6 HiPAC

2. U. Dayal et al.: The HiPAC Project: Combining Active Databases and Timing Constraints; ACM SIGMOD Record 17 (1) 1988, 51–70

5.7 Iris

2. D. Beech: A Foundation for Evolution from Relational to Object Databases; Proc. 1st EDBT 1988, Springer LNCS 303, 251–270

5.8 LOGRES

1. F. Cacace et al.: Integrating Object-Oriented Data Modeling with a Rule-Based Programming Paradigm; Proc. ACM SIGMOD 1990, 225–236
5.9 Ode


5.10 O2

6. W. Cellary, G. Jomier: Consistency of Versions in Object-Oriented Databases; Proc. 16th VLDB 1990, 432-441
12. C. Lecluse, P. Richard: The O2 Database Programming Language; Proc. 15th VLDB 1989, 411-422
15. R. Zicari: Incomplete Information in Object-Oriented Databases; ACM SIGMOD Record 19 (3) 1990, 5-16

5.11 ORION


5.12 OZ+


5.13 POSTGRES


4. M. Stonebraker et al.: QUEL as a Data Type; Proc. ACM SIGMOD 1984, 208–214


8. M. Stonebraker, E.N. Hanson, S. Potamianos: The POSTGRES Rule Manager; IEEE TSE 14, 1988, 897–907


5.14 PROBE


5. F. Manola, J. Orenstein: Toward a General Spatial Data Model for an Object-Oriented DBMS; Proc. 12th VLDB 1986, 326–335


5.15 RAD


5.16 R²D²

1. M. Dürr, A. Kemper: Transaction Control Mechanism for the Object Cache Interface of R²D²; Proc. 3rd Int. Conf. on Data and Knowledge Bases: Improving Usability and Responsiveness, Jerusalem 1988, 81–89

2. A. Kemper, M. Wallrath: Konzepte zur Integration abstrakter Datentypen in R²D²; Proc. BTW 87, Springer IFB 136, 344–359


5.17 ROSE


5.18 3DIS


5.19 VODAK


5.20 ZEITGEIST


5.21 Others

1. R. Ahad: The Object Shell: An Extensible System to Define an Object-Oriented View of an Existing Database; Proc. 2nd Worksh. on OODBS 1988, Springer LNCS 334, 174–192


5. Th. Bode et al.: Coupling the Complex-Relational Data Base CoReDB with the Object Management System OMS; Proc. 1st DEXA 1990, 208–214


13. R. Gagliardi et al.: An Operational Approach to the Integration of Distributed Heterogeneous Environments; Proc. PARBASE-90, 368–377


6 Implementation Aspects

6.1 Storage Organization

1. M.C. Atkins, L.R. Nackman: The Active Deallocation of Objects in Object-Oriented Systems; IBM Research Report RC 13091, Yorktown Heights 1987


5. J. Freitag, Th. Bode: Ein allgemeiner Speicherobjektmanager als Implementierungsbasen für komplexe Objekte in einem Objektmanagementsystem; Proc. BTW 89, Springer IFB 204, 328-349


7. S. Khoshafian, D. Frank: Implementation Techniques for Object Oriented Databases; Proc. 2nd Worksh. on OODBS 1988, Springer LNCS 334, 60-79

8. J.E.B. Moss, S. Sinofsky: Managing Persistent Data with Mneme: Designing a Reliable, Shared Object Interface; Proc. 2nd Worksh. on OODBS 1988, Springer LNCS 334, 298-316


13. B.A. Weston: Segmenting an Object-Oriented Database; IBM Research Report RC 12662, Yorktown Heights 1987

6.2 Concurrency Control and Recovery


2. F. Bancilhon et al.: A Model of CAD Transactions; Proc. 11th VLDB 1985, 25–33


5. M. Cart, J. Ferrie, H. Richy: Optimistic Multi-level Concurrency Control for Nested Typed Objects; Proc. PARBASE-90, 532


15. M.P. Herlihy, W.E. Weihl: Hybrid Concurrency Control for Abstract Data Types; Proc. 7th ACM PODS 1988, 201–210


20. B. Martin: Concurrent Nested Object Computations; Ph. D. Dissertation, UCSD 1988


25. M. Roesler, W.A. Burkhard: Concurrency Control Scheme for Shared Objects: A Peephole Approach Based on Semantics; Proc. 7th ICDCS 1987, 224–231


31. W. Weihl: Data-dependent Concurrency Control and Recovery; Proc. 2nd ACM PODC 1983, 63–75

32. W. Weihl: Local Atomicity Properties: Modular Concurrency Control for Abstract Data Types; ACM TOPLAS 11, 1989, 249–282


### 6.3 Performance Evaluation


### 7 Miscellaneous


2. A.P. Buchman et al.: A Generalized Constraint and Exception Handler for an Object-Oriented CAD-DBMS; Proc. 1st Workshop on OODBS 1986, 38–49


8. G.S. Landis: Design Evolution and History in an Object-Oriented CAD/CAM Database; Proc. 31st IEEE COMPCON (Spring) 1986, 297–303


A Modular and Open Object-Oriented Database System

Satish M. Thatte
Information Technologies Laboratory
Texas Instruments Incorporated
P.O. Box 655474, M/S 238
Dallas, Texas 75265

On 30 August 1990, Texas Instruments Incorporated, Dallas, TX was awarded a three year contract (Contract No. DAAB07-90-C-B920) to develop a modular and open object-oriented database system. The contract is funded by DARPA/ISTO and is being managed by the U.S. Army, CECOM, Fort Monmouth, N.J. The contract is being executed at TI's Information Technologies Laboratory, Computer Science Center, Dallas, Texas. So far, we have received an outstanding response from interested parties (database research community, OODB application developers, OODB builders) to our contract award announcement. This communication is a collection of most commonly asked questions and answers to them.

Questions and Answers

Q: What is a modular and open OODB?

A: It is an OODB based on a modular architecture to support applications written in object-oriented programming languages, initially C++ and Common Lisp Object System (CLOS). Example modules are: Persistent Object Store, Object Manager, Transaction Manager, Change Manager, Object Query Processor. Module interfaces will be fully documented and published. The architecture will be open in the sense that modules can be replaced with other modules, and new modules that are compliant with the Open OODB architecture can be added. Modules that are not needed for an application can also be deleted, supporting a "pay-as-you-need" philosophy. This is in sharp contrast with most database systems, which have proprietary, closed, and monolithic architectures.

Q: What are the main objectives of Open OODB?

A: Complex applications, such as Computer-Aided Design, Computer-Aided Manufacturing (CAM), Computer-Aided Software Engineering (CASE), Command, Control, Communications, and Intelligence (C3I), and knowledge-based expert systems, can derive benefits from object-oriented programming: abstraction, modularity, extensibility, and reuse. However, object-oriented languages do not provide direct support for storing, sharing, and reusing persistent objects. Today's relational databases provide some help. However, since they do not provide data models that match those in programming languages, considerable code must be developed to translate information between the relational and object-oriented worlds (the "impedance mismatch" problem). Various proposed object extensions to the relational data model do not address the impedance mismatch problem between a host object-oriented programming language and an extended relational data model. Further, today's relational databases are usually implemented as monolithic black boxes with internal interfaces that are not public. Special needs frequently arise that cause designers to reimplement special purpose systems at great expense. This contract is aimed at addressing these problems.

An important objective of the contract is to develop technology for building a
modular and open object-oriented database system. It will be arrived at through the consensus of the DARPA community. The system will have a modular architecture which will allow the addition of new modules, the deletion of unneeded modules, and the substitution of modules with user unique modules. The word "open" will be interpreted as follows: all modules of Open OODB will have well-documented interfaces. These modules can be changed or replaced to improve performance or vary functionality.

Q: Can you give us more technical information on Open OODB?
A: Open OODB will be implemented on a standard Unix platform (e.g., Sun SPARC) to support both C++ and CLOS applications. Open OODB will be using standard software (e.g., Unix, C++, CLOS, SQL, and the X Window System), and off-the-shelf language processors and storage servers. It will include an Application Programmer's Interface (API) to support C++ and CLOS applications that require persistent object management facilities (persistence, queries, transactions, change management). The majority of the modules will be implemented in C++ for portability. Open OODB system will be a working and usable system. Open OODB will have the following major modules:

**Persistent Object Management Module:** Provide seamless access to persistent C++ and CLOS objects, via object faulting, and base-level support for query processing, for change management, and for extended transactions.

**Extended Transaction Module:** Support cooperative work that builds on the lower level transactions in the Transactional Store Module to provide nested transactions, long transactions, and a check-in/check-out facility for persistent objects.

**Change Management System Module:** provide support for versions and configurations of objects.

**Object Query Module:** Support set-oriented queries on objects. Complex objects, objects identify, inheritance, and behavioral queries are supported by this module. The query optimizer will be open to allow extensions like semantic optimization, cooperative response, materialized views, and user-supplied indices.

**Data Dictionary Module:** Provide run-time access to object schema (type/class) definitional information.

**Hypermedia Interface:** Interface to an existing Hypermedia Interface Module to provide a way to browse, link, query, and change user-defined objects, demonstrating how a next-generation User interface Toolkit/Management System can couple with Open OODB.

The Open OODB architecture will provide a framework for suppliers of these and future modules to plug in and out the modules. This will allow the builders as well as users of OODB technology to build a customizable OODB based on a "pay-as-you-need" philosophy.
Q: Why concentrate on only persistent C++ and CLOS?

A: C++ and CLOS are two popular object-oriented languages being used today. Both C++ and CLOS are candidates for ANSI standardization effort. Development of a yet another object-oriented language to support persistence and database amenities is outside the scope of this contract. C++ and CLOS represent our initial choice of object-oriented languages. We believe that the Open OODB architecture would allow us to support other object-oriented languages in the future. Many of the Open OODB modules, such as the storage system, will be common to C++ and CLOS clients.

We believe that Persistent (X), where X = C++ and CLOS at the present time, will not be limited to a 'niche' market. Just take a look at how rapidly C++ is gaining popularity; large number of production applications in C++ already exist. We would like to emphasize that Open OODB is not limited to Persistent (X); it will also support transactions, queries, and change management (schema evolution and version control) for Persistent (X) objects.

Q: Why start from a language-based instead of a language-independent object model?

A: We believe that there is a good potential of reaching consensus on Persistent (X), where X = C++ and CLOS. Several proposals have been made to develop language-independent object-oriented data models; some of them are ground-up efforts and some are extensions to the relational data model. None of these object data models have achieved any critical mass. We do not believe that the community is ready to reach a consensus on a language-independent object data model. Moreover, even if such a consensus develops, it will not address the impedance mismatch problem between a host object-oriented programming languages and a language-independent object data model. People will not simply stop using C++ and other popular object-oriented languages, just because a language-independent object data model seems to gain momentum. This does not mean that the development of language-independent object data models is not a noble goal; in fact, it is highly desirable, but its role will be complementary to that of Persistent (X). We believe that both approaches can co-exist and complement each other.

Q: Can you tell us about the process that you will use to execute the project?

A: A major thrust of the contract will be to develop consensus on specifications of OODB modules and interfaces. TI will solicit inputs from DARPA database research community, OODB implementors (prototype builders and commercial vendors), and OODB users (application developers, such as CAD, CAM, hypermedia, expert systems, CASE) to help develop the consensus. Such consensus is essential for developing standards in the rapidly evolving field of OODBs. Without such consensus and eventual standards, the OODB field will remain fragmented, preventing it from reaching its full potential. Similarly, proprietary specifications and subsystems (such as proprietary object models, compilers, storage servers) will also hurt the OODB community. TI will complete an implementation conforming to the consensus on Open OODB in support of C++ and CLOS applications. At DARPA and CECOM requests, TI will distribute the specifications of Open OODB and software implementing those specifications within the DARPA community (up to 25 sites).

To ensure that the specifications of Open OODB are of lasting value, that the Open OODB modules are useful to the DARPA community, and to accelerate agreement within the DARPA community, TI will
conduct four workshops. These workshops will serve to gain acceptance for an open, modular object-oriented database design specification and to obtain consensus on key module interface specifications (e.g., Object Management Module, Extended Transaction Module, Change Management System Module, and object Query Module). Attendees will include key researchers, application developers, end users, and DARPA/DoD representatives (approximately 30). They will have expertise in OODB and conventional databases, object-oriented programming languages, or in application areas such as Computer-Aided Design, Software Computer-Aided Design, or Computer-Integrated Manufacturing. These workshops are described below.

Functional Architecture Validation Workshop: Obtain DARPA community review of the requirements and preliminary Functional Architecture Specification (scheduled on March 13-15, 1991 at Dallas, TX). A "Call for Participation" invitation has been sent to over 100 invitees who had responded to our initial contract award announcement over electronic networks. Final attendees will be selected based on the position papers submitted and directives from DARPA/ISTO and CECOM, subject to the limitation of 30 attendees.


Q: What is the schedule for the Open OODB contract?
A: Phase I of Open OODB (September 1990 - March 1991) will focus on the overall architecture of an OODB and will result in a Preliminary Architecture Specification.

Phase II (March 1991 - December 1991) will focus on module specification.

Phase III (January 1992 - December 1992) will result in module implementations and a demonstration that the modules work together. Distribution to sites for "government purposes" (as defined in the contract) will occur after this phase.

Phase IV in 1993 is an evaluation and consistency phase which will result in a Final Report, including the Final Architecture and Module Interface Specifications.

One workshop is associated with each phase as described earlier. Dates in the above schedule are approximate.

Q: What is the level of funding?
A: The contract award is for $2.6 million over the three year period.

Q: How many people at TI are working on the contract?
A: We cannot reveal that information in compliance with the company policy.

Q: What role will TI's Zeitgeist OODB play in Open OODB?
A: TI has invested its own substantial research effort on its Zeitgeist OODB. Many of the same people that have worked on Zeitgeist OODB will also work on Open OODB. Wherever appropriate, portions of the existing technology base are expected to be reused.
when designing and implementing the Open OODB system.

**Q: Does TI have a plan to productize Open OODB?**

A: As a company policy, TI does not reveal its future product plans.

**Q: How will TI distribute Open OODB software?**

A: The Government shall have the right to distribute to a maximum of 25 sites and to permit up to 100 government, academia, or corporate users to use the Open OODB software for Government Purposes only. The Government may modify and compile any source code furnished as it sees fit so long as the use of such modified or compiled source code is consistent with Government Purpose License Rights.

**Q: What is the relationship of Open OODB to X3/SPARC/DBSSG/ OODBTG, OMG, and other standardization efforts?**

A: Well before the DARPA contract award, the Zeitgeist OODB project at TI has been active in the OODB Task Group, an X3 study group to determine whether and what standards should emerge for OODBs. We are coeditors of the OODBTG Final Report, including the Recommendations for Standards section, due in mid 1991, and of the OODBTG OODB Reference Model, which provides dimensions of comparisons for OODB systems. In addition, we drafted the Survey of Vendors and organized the first of the two OODBTG workshops, held at the SIGMOD '90 Conference in Atlantic City in May, 1990.

In addition, we have been active in Object Management Group and in OMG OODB SIG (Special Interest Group). We participated in drafting the Object Management Architecture that is the backbone of OMG's Standards Manual.

DARPA Open OODB project plans to remain synchronized with both groups. We will adopt the OODBTG OODB Reference Model as a base document. Any revisions suggested by our work will be provided to OODBTG for consideration. Our OODB Requirements Document, a Phase I deliverable, will complement OODBTG and OMG work and make it easier for application developers and OODB developers to communicate their common requirements and coverage. Our Phase I OODB Architecture Document will provide a roadmap of how an OODB can be viewed by OMG either as a "foreign tool" (a complete service) or as a collection of services using the OMG Object Request Broker for communication. Finally, our Phase II deliverables will provide detailed descriptions of the modules of an OODB. They will be useful in comparing different approaches to Persistent C++ and Lisp, Transactions, Object Queries, Change Management, and other Open OODB modules.

**Q: How can I participate in the DARPA Open OODB effort funded at TI?**

A: If you have not already done so, please send the following e-mail message to Dr. Satish Thatte, Program Manager. You will be kept informed based on the response you provide.
To: Thatte@csc.ti.com

Cc: Meares@csc.ti.com

Yes, I am interested in the DARPA-funded Open OODB project at Texas Instruments. Here is the information you requested.

**Location Information**

Name:
Title:
Organization:
Postal address:
E-mail address:
Telephone number:
Fax number:

**Interest Profile**

Please answer by placing an "X" against your interests.

General Interest: receive periodic "news letter" on Open OODB progress from TI
- Interested? Yes -- No -- Maybe --

Consensus building: discuss and critique Open OODB specifications and design (mostly by e-mail)
- Interested? Yes -- No -- Maybe --

Workshop participation: Participate in DARPA Open OODB workshops (subject to TI and DARPA approval, and space limitation)
- Interested? Yes -- No -- Maybe --

Application development on Open OODB: Develop applications on DARPA Open OODB
- Interested? Yes -- No -- Maybe --
(If yes, please indicate what applications you are interested in implementing on top of DARPA Open OODB).

Modules for Open OODB: Develop modules that can be plugged in DARPA Open OODB
- Interested? Yes -- No -- Maybe --
(If yes, please indicate what kind of modules)

Other areas of interests: Please describe briefly.