ISSN 1066-8888

The VLDB Journal

The International Journal on Very Large Data Bases

Volume 4 Number 3 July 1995

Special Issue on Persistent Object Systems

Very
Large
Data
Bases

A Publication of the VLDB Endowment

The
VLDB Journal
The International Journal on Very Large Data Bases

SCOPE AND PURPOSE

The Journal is a quarterly publication of the VLDB Endowment. As a database systems journal it is dedicated to the international publication of scholarly contributions to the advancement of information system architectures, the impact of emerging technologies on information systems, and the development of novel applications. It presents significant advances in the design, implementation, and evaluation of systems for databases and for other information collections. Its scope ranges from the development of special-purpose hardware, the design of innovative software approaches, integrated system architectures, the design analysis and performance evaluation of systems to new techniques for presenting and capturing information.

Editors-in-Chief

Peter M.G. Apers
Computer Science Department
University of Twente
P.O. Box 217
NL-7500 AE Enschede
The Netherlands
e-mail: apers@cs.utwente.nl
tel: +31-53-89-37-19
(Secretary: +31-53-89-36-90)
fax: +31-53-33-96-05

Hans-J. Schek
(*Coordinating Editor-in-Chief*)
Dept. of Computer Science
Swiss Federal Institute of Technology, Zürich
ETH Zentrum
CH-8092 Zürich
Switzerland•
e-mail: schek@inf.ethz.ch
tel: +41-1-632-7240
fax: +41-1-262-3973

Jim Gray
310 Filbert St.
San Francisco, CA 94133
USA
e-mail: gray@crl.com
tel: +1-415-989-3739
fax: +1-415-989-3739

Stanley Y.W. Su
Database Systems Research & Development Center
470 CSE Building, University of Florida
P.O. Box 116125
Gainesville, FL 32611-6125
USA
e-mail: su@pacer.cis.ufl.edu
tel: +1-904-372-2693
(Secretary: +1-904-392-2680)
fax: +1-904-392-1220

Editorial Board

Serge Abiteboul
Michel Adiba
Antonio Albano
Walter A. Burkhard
Wesley Chu
Hector Garcia-Molina
Georges Gardarin
Nathan Goodman
Georg Gottlob
Peter Gray
Tadao Ichikawa
Yahiko Kambayashi
Roger King
Masaru Kitsuregawa
Tosiyasu L. Kunii
Michel Léonard
Dave Lomet
Dennis McLeod
Robert A. Meersman
C. Mohan
John Mylopoulos
Antoni Olivé
M. Tamer Özsu
Alain Pirotte
F.J. Radermacher
K. Ramamohanarao
Andreas Reuter
Ron Sacks-Davis
Peter Scheuerman
Gunter Schlageter
Joachim W. Schmidt
Arne Solvberg
Yannis Vassiliou
Hartmut Wedekind
Kyu-Young Whang
Stanley Zdonik

Publication Board

Arie Shoshani (*Chairman*)
Mail Stop 50B/3238
Lawrence Berkeley Laboratory
Berkeley, CA 94720, USA
e-mail: shoshani@lbl.gov

Peter M.G. Apers, Jim Gray, Hans-J. Schek, Stanley Y.W. Su

Thomas Wu
Department of Computer Science
Naval Postgraduate School

Michael Rys
Institut für Informationssysteme
ETH Zürich

Copyright © 1995 by the VLDB Endowment. Copying without fee is permitted provided that the copies are not made or distributed for direct commercial advantage and credit for the source is given. Abstracting with credit is permitted. For other copying of articles write to the Chairperson of the Publication Board. To copy otherwise, or to republish, requires a fee and/or specific permission.

For submission and subscription information, see inside back cover

The *VLDB* Journal

The International Journal on Very Large Data Bases

Volume 4(3) (1995)

THE·BOXWOOD·PRESS

SINCE 1952

THE BOXWOOD PRESS
PACIFIC GROVE, CA, USA

© 1995 BY THE VLDB ENDOWMENT

Copyright reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owner.

Submission of a paper to *The VLDB Journal* is understood to imply that it is not being considered for publication elsewhere and that the author's permission to publish his/her article(s) in this journal implies the exclusive authorization of the publisher to deal with all issues concerning the copyright therein.

Submission of multi-authored manuscripts to this journal implies the consent of *each* of the authors. The publisher will assume that the senior or corresponding author has specifically obtained the approval of all other co-authors to submit the manuscript to this journal.

No responsibility is assumed by the Publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.

Special regulations for authors. Upon acceptance of an article by the journal, the author(s) will be asked to transfer copyright of the article to the publisher. This transfer will ensure the widest possible dissemination of information.

Published quarterly by:

THE BOXWOOD PRESS
183 Ocean View Blvd.
Pacific Grove, CA 93950, USA
Telephone: +1 408 375 9110/Fax: +1 408 375 0430
Email: boxwood@cruzio.com

Preface to Special Issue on Persistent Object Systems

Persistent object systems are being developed to meet the requirements of persistent application builders. Persistent applications are those where the application may continue to be required for long periods; for example, medical systems where data about people are required throughout their lifetimes. In consequence, the application typically will out-live most of its components and the technologies that currently support it. Both the data and the algorithms which interpret the data remain of interest to the people who use such systems.

Persistent object systems also are used to accommodate and process the wide variety of data that is encountered in these applications. In consequence a rich, and preferably extensible, set of data types is required and all such types and the programs that work with them may require longevity. This challenges researchers to find good models and languages that describe both the data and programs and to discover persistent object store and language implementation techniques that yield adequate performance and reliability.

Persistent object systems are a synergy between the developments in strongly typed programming languages and the advances in databases. The language background contributes new type systems that are extensible yet expressive and economically checkable. It also contributes a predilection for simple rules that combine without exceptions even if this places considerable challenges on implementors. The database research contributes new object-oriented data models and bulk types which offer: improved modeling power, better abstraction, and query algebras. It also contributes implementation technology for: stores, indexing, optimized query evaluation, concurrency, and recovery.

There are now many attempts recorded in the literature to combine aspects of these two major research themes. Persistent object systems are particularly relevant now as implementations combining much larger proportions of these research themes are emerging as viable platforms for application construction.

The longevity envisioned for persistent applications can only be achieved if mechanisms are devised that allow an application to evolve without loss of the essential relationship between code and data. For example, the state of some part of a patient's body may be recorded as data derived from computer generated tomography. The interpretation of those data (e.g., to present it), normally is encoded as program. Several years later, it may be desirable to view those data again and so the relationship between those data and that program has to be retained. On the other hand, it would be unacceptable if new representations and new presentation methods couldn't be introduced during that period. This is a crucial area of research and the relevant technology that is emerging is reflection. Two forms of reflection, behavioral reflection and linguistic reflection, are under consideration and are represented in this issue.

This special issue contains a survey by the invited editors and four articles that between them represent most aspects of Persistent object system research. These aspects cover research that is developing a detailed understanding of the required functionality of persistent object technology, of methods for exploiting the new functionality, and of engineering issues encountered when building this technology.

The invited article, "Orthogonally Persistent Object Systems" by Atkinson, M. and Morrison, R., presents a survey of the research into persistent programming languages and database programming languages over the past fifteen years. It also provides an advanced tutorial for those wishing to commence research in this area from the viewpoint of the programming language research community. Indeed, it enlarges upon a tutorial on that subject at the Twentieth International Conference on Very Large Data Bases (Santiago, Chile, September 1994). It offers definitions of many of the terms used in the other articles, such as: "persistence," "orthogonal persistence," "persistence by reachability," and "swizzling." Language design principles are put forward as the criteria for designing the functionality of object systems. From this viewpoint, regular and simple rules are of dominant importance. Type systems and binding systems are presented as integrating concepts. Linguistic reflection is described as a foundation for evolvable applications. Serious engineering challenges arise during the attempts to implement persistent object stores with the capacity and robustness required for persistent applications. This leads to a catalogue of issues where further research is required. It therefore sets in context the other articles in this issue.

The first article, "An Introduction to Fibonacci: A Programming Language for Object Databases" by Albano, A., Ghelli, G. and Orsini, R., reports a good example of the fruits of persistent object system research. Fibonacci is a new database programming language that provides a nice blend of the developments in both programming languages and databases. For example, it shows the benefit of recent work in type theory and utilizes strong typing, polymorphism, and higher-order procedures, all from the programming language research stream. It also has relations, complex queries, roles, inheritance, and objects that have seen much development in the database context. Such integration of database modeling, program language typing, and computation portends future programming environments that offer clean and well-defined semantics for the application programmer.

The second article, "TIGUKAT: A Uniform Behavioral Objectbase Management System" by Özsu, T., Peters, R., Szafron, D., Irani, B., Lipka, A., and Muñoz, A., reports the design of a new object-oriented data model. As an object based system, it naturally maintains the relationship between code and data, as methods are bound to the objects and the objects' state holds the data. It has two particular areas of interest. It utilizes the same regularity that is desirable in programming language design to reduce the number of concepts and it makes significant use of behavioral reflection to permit the extension and revision necessary for system evolution. This is achieved by making every construct in the language from types to queries an object, so that other objects may manipulate them.

The third article, "Thémis: A Database Programming Language Handling Integrity Constraints" by Benzaken, V. and Doucet, A., describes the design of a language, implemented on top of O_2, that describes and maintains constraints among objects. Defining constraints in the context of objects presents additional difficulties compared with relational systems. These difficulties are similar to the extra difficulties encountered in complex object query languages. A mathematical framework for describing the semantics of constraints is developed and used to show how constraint enforcement can be optimized.

The fourth article, "Adaptable Pointer Swizzling Strategies in Object Bases: Design, Realization, and Quantitative Analysis" by Kemper, A. and Kossmann, D., illustrates the complexities of developing good engineering to support persistent object systems. Swizzling is the replacement of object identifiers by local addresses to accelerate program executions. There is a choice of times at which this may take place. The authors enumerate these and other choices and show that virtually all strategies have some circumstances in which they are optimal. Their analysis and measurements are a good example of the careful engineering research that is now appropriate to the development of high-performance and scalable persistent object systems.

Inevitably, one issue can present only a sampling of the current work in persistent object systems. The articles were selected to illustrate a variety of aspects of current research, while the survey fills in the broader background and provides pointers to many articles in the area.

Persistent object systems are demonstrating their potential to radically simplify the application programming environment, and they are, therefore, expected to reduce the life-time cost of application systems. These results should now influence the development of object-oriented databases and programming languages (e.g., there was a tentative proposal for a persistence annex to the Ada 95 standard). It is hoped that this special issue will make the particular viewpoint taken by persistent object system researchers better understood and attract more researchers to the outstanding issues. It is now an urgent matter to mount large-scale experiments to demonstrate that the potential of these systems can be realized under industrial and commercial conditions. An issue of most fundamental importance is how the longer-term persistence of applications can be supported and, at the same time, utilize an inter-dependent mix of complex data and sophisticated algorithms.

Malcolm Atkinson
Ronald Morrison