Martin Odersky (Ed.)

LNCS 3086

ECOOP 2004 – Object-Oriented Programming

18th European Conference
Oslo, Norway, June 2004
Proceedings

2004
ecoop
OSLO, NORWAY

Springer

Martin Odersky (Ed.)

ECOOP 2004 – Object-Oriented Programming

18th European Conference
Oslo, Norway, June 14-18, 2004
Proceedings

Springer

Volume Editor

Martin Odersky
École Polytechnique Fédérale de Lausanne (EPFL)
School of Computer and Communication Sciences
1015 Lausanne, Switzerland
E-mail: martin.odersky@epfl.ch

Library of Congress Control Number: 2004106985

CR Subject Classification (1998): D.1, D.2, D.3, F.3, C.2, K.4, J.1

ISSN 0302-9743
ISBN 3-540-22159-X Springer-Verlag Berlin Heidelberg New York

Springer-Verlag is a part of Springer Science+Business Media

springeronline.com

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Protago-TeX-Production GmbH
Printed on acid-free paper SPIN: 11012146 06/3142 5 4 3 2 1 0

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Preface

ECOOP is the premier forum in Europe for bringing together practitioners, researchers, and students to share their ideas and experiences in a broad range of disciplines woven with the common thread of object technology. It is a collage of events, including outstanding invited speakers, carefully refereed technical papers, practitioner reports reflecting real-world experience, panels, topic-focused workshops, demonstrations, and an interactive posters session.

The 18th ECOOP 2004 conference held during June 14–18, 2004 in Oslo, Norway represented another year of continued success in object-oriented programming, both as a topic of academic study and as a vehicle for industrial software development. Object-oriented technology has come of age; it is now the commonly established method for most software projects. However, an expanding field of applications and new technological challenges provide a strong demand for research in foundations, design and programming methods, as well as implementation techniques. There is also an increasing interest in the integration of object-orientation with other software development techniques. We anticipate therefore that object-oriented programming will be a fruitful subject of research for many years to come.

This year, the program committee received 132 submissions, of which 25 were accepted for publication after a thorough reviewing process. Every paper received at least 4 reviews. Papers were evaluated based on relevance, significance, clarity, originality, and correctness. The topics covered include: programming concepts, program analysis, software engineering, aspects and components, middleware, verification, systems and implementation techniques. These were complemented by two invited talks, from Matthias Felleisen and Tom Henzinger. Their titles and abstracts are also included in these proceedings.

The success of a major conference such as ECOOP is due to the dedication of many people. I would like to thank the authors for submitting a high number of quality papers; selecting a subset of papers to be published from these was not easy. I would also like to thank the 22 members of the program committee for producing careful reviews, and for sometimes lengthy discussions during the program committee meeting, which was held February 5th and 6th in Lausanne. I thank the general chair of the conference, Birger Møller-Pedersen, and the local organizer, Arne Maus, for productive collaborations in planning the conference and for helping on a number of logistical issues. The AITO Executive Board gave useful guidance. Richard van de Stadt provided invaluable computer-assisted support for the electronic reviewing process, the PC meeting, as well as the production of these proceedings. Finally, Yvette Dubuis at EPFL provided administrative and logistic assistance for running the PC meeting.

April 2004 Martin Odersky

Organization

ECOOP 2004 was organized by the University of Oslo, the Norwegian Computing Center and Sintef, under the auspices of AITO (Association Internationale pour les Technologies Objets), and in cooperation with ACM SIGPLAN.

Executive Committee

Conference Chair
  Birger Møller-Pedersen (University of Oslo)
Program Chair
  Martin Odersky (École Polytechnique Fédérale de Lausanne)
Organizing Chair
  Arne Maus (University of Oslo)

Organizing Committee

Workshop Chairs
  Jacques Malenfant (Université Pierre et Marie Curie)
  Bjarte M. Østvold (Norwegian Computing Center)
Tutorial Chairs
  Arne-Jørgen Berre (Sintef)
  Hanspeter Mössenböck (Johannes Kepler Universität)
PhD Workshop/Doctoral Symposium Chairs
  Susanne Jucknath (Technische Universität Berlin)
  Eric Jul (University of Copenhagen)
Poster/Exhibition/Demonstration Chair
  Ole Smørdal (University of Oslo)
Practitioners' Reports Chair
  Egil P. Andersen (DNV Software)
Memorial Exhibition
  Håvard Hegna (Norwegian Computing Center)
Room Allocation
  Stein Krogdahl (University of Oslo)
Treasurer
  Øystein Haugen (University of Oslo)
Webmaster, Student Volunteers
  Dag Langmyhr (University of Oslo)

Program Committee

Uwe Assmann	Linköpings Universitet, Sweden
Don Batory	University of Texas at Austin, USA
Gilad Bracha	Sun Microsystems, USA
Luca Cardelli	Microsoft Research Cambridge, UK
Charles Consel	LaBRI/INRIA, France
Giuseppe Castagna	École Normale Supérieure, France
Peter Dickman	University of Glasgow, UK
Sophia Drossopoulou	Imperial College, UK
Erik Ernst	University of Aarhus, Denmark
Manuel Fähndrich	Microsoft Research, USA
Giovanna Guerrini	University of Pisa, Italy
Urs Hölzle	Google, USA
Mehdi Jazayeri	Technical University of Vienna, Austria
Shriram Krishnamurthi	Brown University, USA
Doug Lea	SUNY Oswego, USA
Mira Mezini	Darmstadt University of Technology, Germany
Oscar Nierstrasz	University of Bern, Switzerland
Atushi Ohori	JAIST, Japan
Douglas Schmidt	Vanderbilt University, USA
Luís Rodrigues	Universidade de Lisboa, Portugal
Clemens Szyperski	Microsoft Research, USA
Philip Wadler	University of Edinburgh, UK

Referees

Peter von der Ahé
Jonathan Aldrich
Eric Allen
Davide Ancona
Christopher Anderson
Pedro Antunes
Filipe Araujo
Gabriela Arévalo
Kenichi Asai
David F. Bacon
Thomas Ball
Véronique Benzaken
Alexandre Bergel
Lorenzo Bettini
Andrew P. Black
Viviana Bono
Ruth Breu
Kim Bruce
Michele Bugliesi
João Cachopo
Denis Caromel
Luis Carriço
Robert Chatley
Yoonsik Cheon
Jong-Deok Choi
Dave Clarke
Gregory Cobena
Alessandro Coglio
Dario Colazzo
Alexandre di Costanzo
William Cook
Greg Cooper
Erik Corry
Manuvir Das
Rob DeLine
Giorgio Delzanno
Amer Diwan
Damien Doligez
Alan Donovan
Karel Driesen
Stéphane Ducasse
Dominic Duggan
Michael Eichberg
Susan Eisenbach
Sonia Fagorzi
Pascal Fenkam
Robby Findler
Bernd Fischer
Kathleen Fisher
Markus G"alli
Jacques Garrigue
Tudor Girba
Patrice Godefroid
Georges Gonthier
Mark Grechanik
Orla Greevy
Thomas Gschwind
Masahito Hasegawa
Johannes Henkel
Michael Hind
Tom Hirschowitz
Pete Hopkins
Haruo Hosoya
Jim Hugunin
Atsushi Igarashi
Paul Kelly
Graham Kirby
Günter Kniesel
Juliana Küster-Filipe
Giovanni Lagorio
Patrick Lam
Michele Lanza
James Larus
Rustan Leino
Dan Licata
Adrian Lienhard
Jørgen Lindskov Knudsen
Roberto Lopez-Herrejon
Darko Marinov
Francisco Martins
Brian McNamara
Erik Meijer
Marco Mesiti
Todd Millstein
Hugo Miranda
Kjeld H. Mortensen
Gilles Muller
Curran Nachbar

Markus Noga
Isabel Nunes
Robert O'Callahan
Klaus Ostermann
Ulrik Pagh Schultz
Jens Palsberg
David Pearce
João Pedro Neto
Randy Pollack
Laura Ponisio
Seth Proctor
Shaz Qadeer
Sean Quinlan
Gianna Reggio
Jakob Rehof
Geral Reif
Lukas Renggli
Martin Rinard
António Rito-Silva
Kenneth Russell
Alexandru Salcianu
Don Sannella
Nathanael Schärli
Michael I. Schwartzbach
Nahid Shahmehri
Yannis Smaragdakis
V.C. Sreedhar
John Tang Boyland
Tachio Terauchi
Peter Thiemann
Michel Tilman
Mads Torgersen
Vasco T. Vasconcelos
Jan Vitek
Aino Vonge Corry
Jérôme Vouillon
David Walker
Dan Wallach
Stephanie Weirich
Lisa Wells
Toshiaki Yasue
Masahiro Yasugi
Steve Zdancewic
Ying Zhang
Elena Zucca

Lecture Notes in Computer Science

For information about Vols. 1–2989

please contact your bookseller or Springer-Verlag

Vol. 3096: G. Melnik, H. Holz (Eds.), Advances in Learning Software Organizations. X, 173 pages. 2004.

Vol. 3093: S.K. Katsikas, S. Gritzalis, J. Lopez (Eds.), Public Key Infrastructure. XIII, 380 pages. 2004.

Vol. 3092: J. Eckstein, H. Baumeister (Eds.), Extreme Programming and Agile Processes in Software Engineering. XVI, 358 pages. 2004.

Vol. 3091: V. van Oostrom (Ed.), Rewriting Techniques and Applications. X, 313 pages. 2004.

Vol. 3089: M. Jakobsson, M. Yung, J. Zhou (Eds.), Applied Cryptography and Network Security. XIV, 510 pages. 2004.

Vol. 3086: M. Odersky (Ed.), ECOOP 2004 – Object-Oriented Programming. XIII, 611 pages. 2004.

Vol. 3085: S. Berardi, M. Coppo, F. Damiani (Eds.), Types for Proofs and Programs. X, 409 pages. 2004.

Vol. 3084: A. Persson, J. Stirna (Eds.), Advanced Information Systems Engineering. XIV, 596 pages. 2004.

Vol. 3083: W. Emmerich, A.L. Wolf (Eds.), Component Deployment. X, 249 pages. 2004.

Vol. 3078: S. Cotin, D.N. Metaxas (Eds.), Medical Simulation. XVI, 296 pages. 2004.

Vol. 3077: F. Roli, J. Kittler, T. Windeatt (Eds.), Multiple Classifier Systems. XII, 386 pages. 2004.

Vol. 3076: D. Buell (Ed.), Algorithmic Number Theory. XI, 451 pages. 2004.

Vol. 3074: B. Kuijpers, P. Revesz (Eds.), Constraint Databases and Applications. XII, 181 pages. 2004.

Vol. 3073: H. Chen, R. Moore, D.D. Zeng, J. Leavitt (Eds.), Intelligence and Security Informatics. XV, 536 pages. 2004.

Vol. 3070: L. Rutkowski, J. Siekmann, R. Tadeusiewicz, L.A. Zadeh (Eds.), Artificial Intelligence and Soft Computing - ICAISC 2004. XXV, 1208 pages. 2004. (Subseries LNAI).

Vol. 3068: E. André, L. Dybkj\ae r, W. Minker, P. Heisterkamp (Eds.), Affective Dialogue Systems. XII, 324 pages. 2004. (Subseries LNAI).

Vol. 3067: M. Dastani, J. Dix, A. El Fallah-Seghrouchni (Eds.), Programming Multi-Agent Systems. X, 221 pages. 2004. (Subseries LNAI).

Vol. 3066: S. Tsumoto, R. S lowiński, J. Komorowski, J.W. Grzymala-Busse (Eds.), Rough Sets and Current Trends in Computing. XX, 853 pages. 2004. (Subseries LNAI).

Vol. 3065: A. Lomuscio, D. Nute (Eds.), Deontic Logic in Computer Science. X, 275 pages. 2004. (Subseries LNAI).

Vol. 3064: D. Bienstock, G. Nemhauser (Eds.), Integer Programming and Combinatorial Optimization. XI, 445 pages. 2004.

Vol. 3063: A. Llamosí, A. Strohmeier (Eds.), Reliable Software Technologies - Ada-Europe 2004. XIII, 333 pages. 2004.

Vol. 3062: J.L. Pfaltz, M. Nagl, B. Böhlen (Eds.), Applications of Graph Transformations with Industrial Relevance. XV, 500 pages. 2004.

Vol. 3060: A.Y. Tawfik, S.D. Goodwin (Eds.), Advances in Artificial Intelligence. XIII, 582 pages. 2004. (Subseries LNAI).

Vol. 3059: C.C. Ribeiro, S.L. Martins (Eds.), Experimental and Efficient Algorithms. X, 586 pages. 2004.

Vol. 3058: N. Sebe, M.S. Lew, T.S. Huang (Eds.), Computer Vision in Human-Computer Interaction. X, 233 pages. 2004.

Vol. 3056: H. Dai, R. Srikant, C. Zhang (Eds.), Advances in Knowledge Discovery and Data Mining. XIX, 713 pages. 2004. (Subseries LNAI).

Vol. 3054: I. Crnkovic, J.A. Stafford, H.W. Schmidt, K. Wallnau (Eds.), Component-Based Software Engineering. XI, 311 pages. 2004.

Vol. 3053: C. Bussler, J. Davies, D. Fensel, R. Studer (Eds.), The Semantic Web: Research and Applications. XIII, 490 pages. 2004.

Vol. 3052: W. Zimmermann, B. Thalheim (Eds.), Abstract State Machines 2004. Advances in Theory and Practice. XII, 235 pages. 2004.

Vol. 3051: R. Berghammer, B. Möller, G. Struth (Eds.), Relational and Kleene-Algebraic Methods in Computer Science. X, 279 pages. 2004.

Vol. 3050: J. Domingo-Ferrer, V. Torra (Eds.), Privacy in Statistical Databases. IX, 367 pages. 2004.

Vol. 3047: F. Oquendo, B. Warboys, R. Morrison (Eds.), Software Architecture. X, 279 pages. 2004.

Vol. 3046: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.K. Tan, O. Gervasi (Eds.), Computational Science and Its Applications – ICCSA 2004. LIII, 1016 pages. 2004.

Vol. 3045: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.K. Tan, O. Gervasi (Eds.), Computational Science and Its Applications – ICCSA 2004. LIII, 1040 pages. 2004.

Vol. 3044: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.K. Tan, O. Gervasi (Eds.), Computational Science and Its Applications – ICCSA 2004. LIII, 1140 pages. 2004.

Vol. 3043: A. Laganà, M.L. Gavrilova, V. Kumar, Y. Mun, C.K. Tan, O. Gervasi (Eds.), Computational Science and Its Applications – ICCSA 2004. LIII, 1180 pages. 2004.

Vol. 3042: N. Mitrou, K. Kontovasilis, G.N. Rouskas, I. Iliadis, L. Merakos (Eds.), NETWORKING 2004, Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications. XXXIII, 1519 pages. 2004.

Vol. 3039: M. Bubak, G.D.v. Albada, P.M. Sloot, J.J. Dongarra (Eds.), Computational Science - ICCS 2004. LXVI, 1271 pages. 2004.

Vol. 3038: M. Bubak, G.D.v. Albada, P.M. Sloot, J.J. Dongarra (Eds.), Computational Science - ICCS 2004. LXVI, 1311 pages. 2004.

Vol. 3037: M. Bubak, G.D.v. Albada, P.M. Sloot, J.J. Dongarra (Eds.), Computational Science - ICCS 2004. LXVI, 745 pages. 2004.

Vol. 3036: M. Bubak, G.D.v. Albada, P.M. Sloot, J.J. Dongarra (Eds.), Computational Science - ICCS 2004. LXVI, 713 pages. 2004.

Vol. 3035: M.A. Wimmer (Ed.), Knowledge Management in Electronic Government. XII, 326 pages. 2004. (Subseries LNAI).

Vol. 3034: J. Favela, E. Menasalvas, E. Chávez (Eds.), Advances in Web Intelligence. XIII, 227 pages. 2004. (Subseries LNAI).

Vol. 3033: M. Li, X.-H. Sun, Q. Deng, J. Ni (Eds.), Grid and Cooperative Computing. XXXVIII, 1076 pages. 2004.

Vol. 3032: M. Li, X.-H. Sun, Q. Deng, J. Ni (Eds.), Grid and Cooperative Computing. XXXVII, 1112 pages. 2004.

Vol. 3031: A. Butz, A. Krüger, P. Olivier (Eds.), Smart Graphics. X, 165 pages. 2004.

Vol. 3030: P. Giorgini, B. Henderson-Sellers, M. Winikoff (Eds.), Agent-Oriented Information Systems. XIV, 207 pages. 2004. (Subseries LNAI).

Vol. 3029: B. Orchard, C. Yang, M. Ali (Eds.), Innovations in Applied Artificial Intelligence. XXI, 1272 pages. 2004. (Subseries LNAI).

Vol. 3028: D. Neuenschwander, Probabilistic and Statistical Methods in Cryptology. X, 158 pages. 2004.

Vol. 3027: C. Cachin, J. Camenisch (Eds.), Advances in Cryptology - EUROCRYPT 2004. XI, 628 pages. 2004.

Vol. 3026: C. Ramamoorthy, R. Lee, K.W. Lee (Eds.), Software Engineering Research and Applications. XV, 377 pages. 2004.

Vol. 3025: G.A. Vouros, T. Panayiotopoulos (Eds.), Methods and Applications of Artificial Intelligence. XV, 546 pages. 2004. (Subseries LNAI).

Vol. 3024: T. Pajdla, J. Matas (Eds.), Computer Vision - ECCV 2004. XXVIII, 621 pages. 2004.

Vol. 3023: T. Pajdla, J. Matas (Eds.), Computer Vision - ECCV 2004. XXVIII, 611 pages. 2004.

Vol. 3022: T. Pajdla, J. Matas (Eds.), Computer Vision - ECCV 2004. XXVIII, 621 pages. 2004.

Vol. 3021: T. Pajdla, J. Matas (Eds.), Computer Vision - ECCV 2004. XXVIII, 633 pages. 2004.

Vol. 3019: R. Wyrzykowski, J.J. Dongarra, M. Paprzycki, J. Wasniewski (Eds.), Parallel Processing and Applied Mathematics. XIX, 1174 pages. 2004.

Vol. 3018: M. Bruynooghe (Ed.), Logic Based Program Synthesis and Transformation. X, 233 pages. 2004.

Vol. 3016: C. Lengauer, D. Batory, C. Consel, M. Odersky (Eds.), Domain-Specific Program Generation. XII, 325 pages. 2004.

Vol. 3015: C. Barakat, I. Pratt (Eds.), Passive and Active Network Measurement. XI, 300 pages. 2004.

Vol. 3014: F. van der Linden (Ed.), Software Product-Family Engineering. IX, 486 pages. 2004.

Vol. 3012: K. Kurumatani, S.-H. Chen, A. Ohuchi (Eds.), Multi-Agnets for Mass User Support. X, 217 pages. 2004. (Subseries LNAI).

Vol. 3011: J.-C. Régin, M. Rueher (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. XI, 415 pages. 2004.

Vol. 3010: K.R. Apt, F. Fages, F. Rossi, P. Szeredi, J. Váncza (Eds.), Recent Advances in Constraints. VIII, 285 pages. 2004. (Subseries LNAI).

Vol. 3009: F. Bomarius, H. Iida (Eds.), Product Focused Software Process Improvement. XIV, 584 pages. 2004.

Vol. 3008: S. Heuel, Uncertain Projective Geometry. XVII, 205 pages. 2004.

Vol. 3007: J.X. Yu, X. Lin, H. Lu, Y. Zhang (Eds.), Advanced Web Technologies and Applications. XXII, 936 pages. 2004.

Vol. 3006: M. Matsui, R. Zuccherato (Eds.), Selected Areas in Cryptography. XI, 361 pages. 2004.

Vol. 3005: G.R. Raidl, S. Cagnoni, J. Branke, D.W. Corne, R. Drechsler, Y. Jin, C.G. Johnson, P. Machado, E. Marchiori, F. Rothlauf, G.D. Smith, G. Squillero (Eds.), Applications of Evolutionary Computing. XVII, 562 pages. 2004.

Vol. 3004: J. Gottlieb, G.R. Raidl (Eds.), Evolutionary Computation in Combinatorial Optimization. X, 241 pages. 2004.

Vol. 3003: M. Keijzer, U.-M. O'Reilly, S.M. Lucas, E. Costa, T. Soule (Eds.), Genetic Programming. XI, 410 pages. 2004.

Vol. 3002: D.L. Hicks (Ed.), Metainformatics. X, 213 pages. 2004.

Vol. 3001: A. Ferscha, F. Mattern (Eds.), Pervasive Computing. XVII, 358 pages. 2004.

Vol. 2999: E.A. Boiten, J. Derrick, G. Smith (Eds.), Integrated Formal Methods. XI, 541 pages. 2004.

Vol. 2998: Y. Kameyama, P.J. Stuckey (Eds.), Functional and Logic Programming. X, 307 pages. 2004.

Vol. 2997: S. McDonald, J. Tait (Eds.), Advances in Information Retrieval. XIII, 427 pages. 2004.

Vol. 2996: V. Diekert, M. Habib (Eds.), STACS 2004. XVI, 658 pages. 2004.

Vol. 2995: C. Jensen, S. Poslad, T. Dimitrakos (Eds.), Trust Management. XIII, 377 pages. 2004.

Vol. 2994: E. Rahm (Ed.), Data Integration in the Life Sciences. X, 221 pages. 2004. (Subseries LNBI).

Vol. 2993: R. Alur, G.J. Pappas (Eds.), Hybrid Systems: Computation and Control. XII, 674 pages. 2004.

Vol. 2992: E. Bertino, S. Christodoulakis, D. Plexousakis, V. Christophides, M. Koubarakis, K. Böhm, E. Ferrari (Eds.), Advances in Database Technology - EDBT 2004. XVIII, 877 pages. 2004.

Vol. 2991: R. Alt, A. Frommer, R.B. Kearfott, W. Luther (Eds.), Numerical Software with Result Verification. X, 315 pages. 2004.

Vol. 2990: J. Leite, A. Omicini, L. Sterling, P. Torroni (Eds.), Declarative Agent Languages and Technologies. XII, 281 pages. 2004. (Subseries LNAI).

Table of Contents

Invited Talk 1

Middleware

Types

Verification

Invited Talk 2

Systems

Ownership Domains: Separating Aliasing Policy from Mechanism

Jonathan Aldrich[1] and Craig Chambers[2]

[1] Carnegie Mellon University, Pittsburgh, PA 15217, USA,
jonathan.aldrich@cs.cmu.edu

[2] University of Washington, Seattle, WA, 98195, USA,
chambers@cs.washington.edu

Abstract. Ownership types promise to provide a practical mechanism for enforcing stronger encapsulation by controlling aliasing in object-oriented languages. However, previous ownership type proposals have tied the aliasing policy of a system to the mechanism of ownership. As a result, these proposals are too weak to express many important aliasing constraints, yet also so restrictive that they prohibit many useful programming idioms.

In this paper, we propose *ownership domains*, which decouple encapsulation policy from the mechanism of ownership in two key ways. First, developers can specify multiple ownership domains for each object, permitting a fine-grained control of aliasing compared to systems that provide only one ownership domain for each object. Second, developers can specify the permitted aliasing between each pair of domains in the system, providing more flexibility compared to systems that enforce a fixed policy for inter-domain aliasing. Because it decouples policy from mechanism, our alias control system is both more precise and more flexible than previous ownership type systems.

1 Introduction

One of the primary challenges in building and evolving large object-oriented systems is reasoning about aliasing between objects. Unexpected aliasing can lead to broken invariants, mistaken assumptions, security holes, and surprising side effects, which in turn may cause defects and complicate software evolution.

Ownership types are one promising approach to addressing the problems of uncontrolled aliasing [23,13,10,4,8,11]. With ownership types, the developer of an abstract data type can encapsulate objects used in the internal representation of the ADT, and use static typechecking to ensure that clients of the ADT cannot access its representation.

Despite the potential of ownership types, current ownership type systems have serious limitations, both in the kinds of aliasing constraints they can express and in their ability to support important programming idioms. These limitations

M. Odersky (Ed.): ECOOP 2004, LNCS 3086, pp. 1–25, 2004.

can be understood by looking at ownership types as a combination of a *mechanism* for dividing objects into hierarchical groups, and a *policy* for constraining references between objects in those groups.

In previous ownership type systems, each object defines a single group to hold its private state. We will call these groups *ownership domains*. The ownership mechanism is useful for separating the internals of an abstract data type from clients of the ADT, but since each object defines only one ownership domain, ownership types cannot be used to reason about aliasing between different subsystems within an object.

The aliasing policy in previous ownership type systems is fixed: the private state of an object can refer to the outside world, but the outside world may not refer to the private state of the object. This policy is known as owners-as-dominators, because it implies that all paths to an object in a system must go through that object's owner.

This fixed policy is useful for ensuring that clients cannot access the internals of an abstract data type. However, the policy is too restrictive to support common programming idioms such as iterator objects or event callbacks. In these idioms, the iterator or event callback objects must be outside of the ADT so that clients can use them, but they must be able to access the internals of the ADT to do their jobs. Thus, iterators or callbacks create paths to the inside of an ADT that do not go through the ADT object itself, violating owners-as-dominators.

In this paper, we propose ownership domains, an extension of ownership types that separates the alias control policy of a system from the mechanism of ownership. Our system generalizes the mechanism of ownership to permit multiple ownership domains in each object. Each domain represents a logically related set of objects. Thus, developers can use the ownership domain mechanism to divide a system object into multiple subsystems, and can separately specify the policy that determines how those subsystems can interact.

Instead of hard-wiring an aliasing policy into the ownership mechanism, our system allows engineers to specify in detail the permitted aliasing relationships between domains. For example, a Sequence ADT can declare one domain for its internal representation, and a second domain for its iterators. The aliasing policy for the Sequence ADT can be written to allow clients to access the iterators, and to allow the iterators to access the internal representation of the Sequence, while prohibiting clients from accessing the internal representation directly.

As a result of separating the mechanism for dividing objects into domains from the policy of how objects in those domains may interact, our system is both more precise and more flexible than previous ownership type systems. It is more precise in allowing developers to control aliasing between the sub-parts of an object. Furthermore, while our system can be used to statically enforce the owners-as-dominators property, it also supports more flexible alias control policies that permit idioms like iterators or events.

The rest of this paper is organized as follows. In the next section we introduce ownership domains by example, showing how they can express aliasing policies and code idioms that were not expressible in previous systems. We have im-