Thomas Gschwind
Cecilia Mascolo (Eds.)

LNCS 3437

# Software Engineering and Middleware

4th International Workshop, SEM 2004
Linz, Austria, September 2004
Revised Selected Papers

Springer

Thomas Gschwind   Cecilia Mascolo (Eds.)

# Software Engineering and Middleware

4th International Workshop, SEM 2004
Linz, Austria, September 20-21, 2004
Revised Selected Papers

Springer

Volume Editors

Thomas Gschwind
IBM Research, Zurich Research Laboratory
Säumerstrasse 4, 8803 Rüschlikon, Switzerland
E-mail: thomasg at ieee.org

Cecilia Mascolo
University College of London, Department of Computer Science
Gower Street, London WC1E 6BT, UK
E-mail: c.mascolo at cs.ucl.ac.uk

Library of Congress Control Number: 2005923878

CR Subject Classification (1998): D.2, C.2.4, D.3.4

ISSN 0302-9743
ISBN-10 3-540-25328-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-25328-0 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11407386 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3437

*Commenced Publication in 1973*
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

# Lecture Notes in Computer Science

For information about Vols. 1–3358

please contact your bookseller or Springer

Vol. 3467: J. Giesl (Ed.), Term Rewriting and Applications. XIII, 517 pages. 2005.

Vol. 3461: P. Urzyczyn (Ed.), Typed Lambda Calculi and Applications. XI, 433 pages. 2005.

Vol. 3459: R. Kimmel, N. Sochen, J. Weickert (Eds.), Scale Space and PDE Methods in Computer Vision. XI, 634 pages. 2005.

Vol. 3456: H. Rust, Operational Semantics for Timed Systems. XII, 223 pages. 2005.

Vol. 3455: H. Treharne, S. King, M. Henson, S. Schneider (Eds.), ZB 2005: Formal Specification and Development in Z and B. XV, 493 pages. 2005.

Vol. 3454: J.-M. Jacquet, G.P. Picco (Eds.), Coordination Models and Languages. X, 299 pages. 2005.

Vol. 3453: L. Zhou, B.C. Ooi, X. Meng (Eds.), Database Systems for Advanced Applications. XXVII, 929 pages. 2005.

Vol. 3452: F. Baader, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XI, 562 pages. 2005. (Subseries LNAI).

Vol. 3450: D. Hutter, M. Ullmann (Eds.), Security in Pervasive Computing. XI, 239 pages. 2005.

Vol. 3449: F. Rothlauf, J. Branke, S. Cagnoni, D.W. Corne, R. Drechsler, Y. Jin, P. Machado, E. Marchiori, J. Romero, G.D. Smith, G. Squillero (Eds.), Applications on Evolutionary Computing. XX, 631 pages. 2005.

Vol. 3448: G.R. Raidl, J. Gottlieb (Eds.), Evolutionary Computation in Combinatorial Optimization. XI, 271 pages. 2005.

Vol. 3447: M. Keijzer, A. Tettamanzi, P. Collet, J.v. Hemert, M. Tomassini (Eds.), Genetic Programming. XIII, 382 pages. 2005.

Vol. 3444: M. Sagiv (Ed.), Programming Languages and Systems. XIII, 439 pages. 2005.

Vol. 3443: R. Bodik (Ed.), Compiler Construction. XI, 305 pages. 2005.

Vol. 3442: M. Cerioli (Ed.), Fundamental Approaches to Software Engineering. XIII, 373 pages. 2005.

Vol. 3441: V. Sassone (Ed.), Foundations of Software Science and Computational Structures. XVIII, 521 pages. 2005.

Vol. 3440: N. Halbwachs, L.D. Zuck (Eds.), Tools and Algorithms for the Construction and Analysis of Systems. XVII, 588 pages. 2005.

Vol. 3439: R.H. Deng, F. Bao, H. Pang, J. Zhou (Eds.), Information Security Practice and Experience. XII, 424 pages. 2005.

Vol. 3437: T. Gschwind, C. Mascolo (Eds.), Software Engineering and Middleware. X, 245 pages. 2005.

Vol. 3436: B. Bouyssounouse, J. Sifakis (Eds.), Embedded Systems Design. XV, 492 pages. 2005.

Vol. 3434: L. Brun, M. Vento (Eds.), Graph-Based Representations in Pattern Recognition. XII, 384 pages. 2005.

Vol. 3433: S. Bhalla (Ed.), Databases in Networked Information Systems. VII, 319 pages. 2005.

Vol. 3432: M. Beigl, P. Lukowicz (Eds.), Systems Aspects in Organic and Pervasive Computing - ARCS 2005. X, 265 pages. 2005.

Vol. 3431: C. Dovrolis (Ed.), Passive and Active Network Measurement. XII, 374 pages. 2005.

Vol. 3429: E. Andres, G. Damiand, P. Lienhardt (Eds.), Discrete Geometry for Computer Imagery. X, 428 pages. 2005.

Vol. 3427: G. Kotsis, O. Spaniol, Wireless Systems and Mobility in Next Generation Internet. VIII, 249 pages. 2005.

Vol. 3423: J.L. Fiadeiro, P.D. Mosses, F. Orejas (Eds.), Recent Trends in Algebraic Development Techniques. VIII, 271 pages. 2005.

Vol. 3422: R.T. Mittermeir (Ed.), From Computer Literacy to Informatics Fundamentals. X, 203 pages. 2005.

Vol. 3421: P. Lorenz, P. Dini (Eds.), Networking - ICN 2005, Part II. XXXV, 1153 pages. 2005.

Vol. 3420: P. Lorenz, P. Dini (Eds.), Networking - ICN 2005, Part I. XXXV, 933 pages. 2005.

Vol. 3419: B. Faltings, A. Petcu, F. Fages, F. Rossi (Eds.), Constraint Satisfaction and Constraint Logic Programming. X, 217 pages. 2005. (Subseries LNAI).

Vol. 3418: U. Brandes, T. Erlebach (Eds.), Network Analysis. XII, 471 pages. 2005.

Vol. 3416: M. Böhlen, J. Gamper, W. Polasek, M.A. Wimmer (Eds.), E-Government: Towards Electronic Democracy. XIII, 311 pages. 2005. (Subseries LNAI).

Vol. 3415: P. Davidsson, B. Logan, K. Takadama (Eds.), Multi-Agent and Multi-Agent-Based Simulation. X, 265 pages. 2005. (Subseries LNAI).

Vol. 3414: M. Morari, L. Thiele (Eds.), Hybrid Systems: Computation and Control. XII, 684 pages. 2005.

Vol. 3412: X. Franch, D. Port (Eds.), COTS-Based Software Systems. XVI, 312 pages. 2005.

Vol. 3411: S.H. Myaeng, M. Zhou, K.-F. Wong, H.-J. Zhang (Eds.), Information Retrieval Technology. XIII, 337 pages. 2005.

Vol. 3410: C.A. Coello Coello, A. Hernández Aguirre, E. Zitzler (Eds.), Evolutionary Multi-Criterion Optimization. XVI, 912 pages. 2005.

Vol. 3409: N. Guelfi, G. Reggio, A. Romanovsky (Eds.), Scientific Engineering of Distributed Java Applications. X, 127 pages. 2005.

Vol. 3408: D.E. Losada, J.M. Fernández-Luna (Eds.), Advances in Information Retrieval. XVII, 572 pages. 2005.

Vol. 3407: Z. Liu, K. Araki (Eds.), Theoretical Aspects of Computing - ICTAC 2004. XIV, 562 pages. 2005.

Vol. 3406: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing. XVII, 829 pages. 2005.

Vol. 3404: V. Diekert, B. Durand (Eds.), STACS 2005. XVI, 706 pages. 2005.

Vol. 3403: B. Ganter, R. Godin (Eds.), Formal Concept Analysis. XI, 419 pages. 2005. (Subseries LNAI).

Vol. 3401: Z. Li, L.G. Vulkov, J. Waśniewski (Eds.), Numerical Analysis and Its Applications. XIII, 630 pages. 2005.

Vol. 3399: Y. Zhang, K. Tanaka, J.X. Yu, S. Wang, M. Li (Eds.), Web Technologies Research and Development - APWeb 2005. XXII, 1082 pages. 2005.

Vol. 3398: D.-K. Baik (Ed.), Systems Modeling and Simulation: Theory and Applications. XIV, 733 pages. 2005. (Subseries LNAI).

Vol. 3397: T.G. Kim (Ed.), Artificial Intelligence and Simulation. XV, 711 pages. 2005. (Subseries LNAI).

Vol. 3396: R.M. van Eijk, M.-P. Huget, F. Dignum (Eds.), Agent Communication. X, 261 pages. 2005. (Subseries LNAI).

Vol. 3395: J. Grabowski, B. Nielsen (Eds.), Formal Approaches to Software Testing. X, 225 pages. 2005.

Vol. 3394: D. Kudenko, D. Kazakov, E. Alonso (Eds.), Adaptive Agents and Multi-Agent Systems III. VIII, 313 pages. 2005. (Subseries LNAI).

Vol. 3393: H.-J. Kreowski, U. Montanari, F. Orejas, G. Rozenberg, G. Taentzer (Eds.), Formal Methods in Software and Systems Modeling. XXVII, 413 pages. 2005.

Vol. 3392: D. Seipel, M. Hanus, U. Geske, O. Bartenstein (Eds.), Applications of Declarative Programming and Knowledge Management. X, 309 pages. 2005. (Subseries LNAI).

Vol. 3391: C. Kim (Ed.), Information Networking. XVII, 936 pages. 2005.

Vol. 3390: R. Choren, A. Garcia, C. Lucena, A. Romanovsky (Eds.), Software Engineering for Multi-Agent Systems III. XII, 291 pages. 2005.

Vol. 3389: P. Van Roy (Ed.), Multiparadigm Programming in Mozart/OZ. XV, 329 pages. 2005.

Vol. 3388: J. Lagergren (Ed.), Comparative Genomics. VII, 133 pages. 2005. (Subseries LNBI).

Vol. 3387: J. Cardoso, A. Sheth (Eds.), Semantic Web Services and Web Process Composition. VIII, 147 pages. 2005.

Vol. 3386: S. Vaudenay (Ed.), Public Key Cryptography - PKC 2005. IX, 436 pages. 2005.

Vol. 3385: R. Cousot (Ed.), Verification, Model Checking, and Abstract Interpretation. XII, 483 pages. 2005.

Vol. 3383: J. Pach (Ed.), Graph Drawing. XII, 536 pages. 2005.

Vol. 3382: J. Odell, P. Giorgini, J.P. Müller (Eds.), Agent-Oriented Software Engineering V. X, 239 pages. 2005.

Vol. 3381: P. Vojtáš, M. Bieliková, B. Charron-Bost, O. Sýkora (Eds.), SOFSEM 2005: Theory and Practice of Computer Science. XV, 448 pages. 2005.

Vol. 3380: C. Priami, Transactions on Computational Systems Biology I. IX, 111 pages. 2005. (Subseries LNBI).

Vol. 3379: M. Hemmje, C. Niederee, T. Risse (Eds.), From Integrated Publication and Information Systems to Information and Knowledge Environments. XXIV, 321 pages. 2005.

Vol. 3378: J. Kilian (Ed.), Theory of Cryptography. XII, 621 pages. 2005.

Vol. 3377: B. Goethals, A. Siebes (Eds.), Knowledge Discovery in Inductive Databases. VII, 190 pages. 2005.

Vol. 3376: A. Menezes (Ed.), Topics in Cryptology – CT-RSA 2005. X, 385 pages. 2005.

Vol. 3375: M.A. Marsan, G. Bianchi, M. Listanti, M. Meo (Eds.), Quality of Service in Multiservice IP Networks. XIII, 656 pages. 2005.

Vol. 3374: D. Weyns, H.V.D. Parunak, F. Michel (Eds.), Environments for Multi-Agent Systems. X, 279 pages. 2005. (Subseries LNAI).

Vol. 3372: C. Bussler, V. Tannen, I. Fundulaki (Eds.), Semantic Web and Databases. X, 227 pages. 2005.

Vol. 3371: M.W. Barley, N. Kasabov (Eds.), Intelligent Agents and Multi-Agent Systems. X, 329 pages. 2005. (Subseries LNAI).

Vol. 3370: A. Konagaya, K. Satou (Eds.), Grid Computing in Life Science. X, 188 pages. 2005. (Subseries LNBI).

Vol. 3369: V.R. Benjamins, P. Casanovas, J. Breuker, A. Gangemi (Eds.), Law and the Semantic Web. XII, 249 pages. 2005. (Subseries LNAI).

Vol. 3368: L. Paletta, J.K. Tsotsos, E. Rome, G.W. Humphreys (Eds.), Attention and Performance in Computational Vision. VIII, 231 pages. 2005.

Vol. 3367: W.S. Ng, B.C. Ooi, A. Ouksel, C. Sartori (Eds.), Databases, Information Systems, and Peer-to-Peer Computing. X, 231 pages. 2005.

Vol. 3366: I. Rahwan, P. Moraitis, C. Reed (Eds.), Argumentation in Multi-Agent Systems. XII, 263 pages. 2005. (Subseries LNAI).

Vol. 3365: G. Mauri, G. Păun, M.J. Pérez-Jiménez, G. Rozenberg, A. Salomaa (Eds.), Membrane Computing. IX, 415 pages. 2005.

Vol. 3363: T. Eiter, L. Libkin (Eds.), Database Theory - ICDT 2005. XI, 413 pages. 2004.

Vol. 3362: G. Barthe, L. Burdy, M. Huisman, J.-L. Lanet, T. Muntean (Eds.), Construction and Analysis of Safe, Secure, and Interoperable Smart Devices. IX, 257 pages. 2005.

Vol. 3361: S. Bengio, H. Bourlard (Eds.), Machine Learning for Multimodal Interaction. XII, 362 pages. 2005.

Vol. 3360: S. Spaccapietra, E. Bertino, S. Jajodia, R. King, D. McLeod, M.E. Orlowska, L. Strous (Eds.), Journal on Data Semantics II. XI, 223 pages. 2005.

Vol. 3359: G. Grieser, Y. Tanaka (Eds.), Intuitive Human Interfaces for Organizing and Accessing Intellectual Assets. XIV, 257 pages. 2005. (Subseries LNAI).

# Preface

Middleware provides an integration framework for multiple and potentially diverse computing platforms. It allows developers to engineer distributed applications more easily, providing abstractions and primitives to handle distribution and coordination.

Middleware is constantly facing new challenges. Today's advances in computing, including development of pervasive applications, exacerbates the diversity problem, introducing variations not only in terms of performance, but also in terms of environments and device characteristics. Software engineers are therefore challenged both in the area of the development of new and scalable middleware systems, where open, heterogeneous, component-based platforms should provide richer functionality and services, and in the area of application development, where tools to simplify the use of middleware solutions are necessary.

Software Engineering and Middleware is the premier workshop for the research and practice community of software engineering working in both areas to present and discuss new ideas in this field. SEM 2004 was the fourth international workshop on software engineering and middleware of the EDO/SEM workshop series. Previous workshops of this series were successfully held in 2002, 2000 and 1999. Most of the proceedings have been published by Springer in the Lecture Notes in Computer Science series.

The program consisted of a keynote given by Prof. Gustavo Alonso and 16 technical paper presentations. The technical papers were carefully selected from a total of 44 submitted papers. Each paper was thoroughly peer reviewed by at least three members of the Program Committee and consensus on acceptance was achieved by means of an electronic PC discussion. Among the accepted papers, the Program Committee selected the paper "Formally Designing an Event-Based Application for Mobile Collaboration: A Case Study," by Pascal Fenkam and Mehdi Jazayeri, for the Best Paper Award, and the paper "Towards the Development of Ubiquitous Middleware Product Lines," by Sven Apel and Klemens Böhm, for the Best Student Paper Award.

The organizers would like to express their appreciation to a large number of people without whom this event would not have been possible: the authors of submitted papers; the Steering Committee, the Program Committee, and the external referees for their careful reviews and active participation in the paper selection process; and Michael Fischer who managed the electronic submission and reviewing service. We would also like to thank Paul Gruenbacher, in his role as General Chair of ASE, and Stefan Tai and George Spanoudakis, in their roles as ASE Workshops Chairs; they simplified our task considerably by scheduling our work and providing us with templates and instructions. Especially, we would like to thank the Steering Committee for giving us the opportunity to lead this instance of the SEM workshop, and for their invaluable advice.

Finally, we are extremely grateful to IBM for the continuous support given to the workshop: this has allowed us to offer prizes and to sponsor students' participation.

November 2004

Thomas Gschwind and Cecilia Mascolo
Program Co-chairs
SEM 2004

# Organization

This year's Software Engineering and Middleware Workshop (SEM 2004) was held on September 20-21, 2004, in Linz, Austria, as a co-located event of the International Conference on Automated Software Engineering 2004. SEM 2004 was the fourth international workshop on software engineering and middleware of the EDO/SEM workshop series.

## Executive Committee

| | |
|---|---|
| General Chair | Gerti Kappel (Technische Universität Wien, Austria) |
| Program Co-chairs | Thomas Gschwind (IBM Research, Switzerland) |
| | Cecilia Mascolo (University College London, UK) |
| Web Chair | Michael Fischer (Technische Universität Wien, Austria) |
| Steering Committee | Alberto Coen-Porisini (Università dell'Insubria, Italy) |
| | Premkumar Devanbu (University of California, Davis, USA) |
| | Wolfgang Emmerich (University College London, UK) |
| | Volker Gruhn (Universität Leipzig, Germany) |
| | Stefan Tai (IBM Research, USA) |
| | André van der Hoek (University of California, Irvine, USA) |

## Program Committee

Judith Bishop (University of Pretoria, South Africa)
Gordon Blair (Lancaster University, UK)
Licia Capra (University College London, UK)
Antonio Carzaniga (University of Colorado at Boulder, USA)
Jun Han (Swinburne University of Technology, Australia)
Katsuro Inoue (Osaka University, Japan)
Paola Inverardi (Università dell'Aquila, Italy)
Valerie Issarny (INRIA, France)
Arno Jacobson (University of Toronto, Canada)
Mehdi Jazayeri (Technische Universität Wien, Austria)
Doug Lea (State University of New York at Oswego, USA)
Amy Murphy (University of Rochester, USA)
Koichiro Ochimizu (Japan Institute of Technology, Japan)
Gian Pietro Picco (Politecnico di Milano, Italy)

Isabelle Rouvellou (IBM Watson Research, USA)
Steve Vinoski (IONA, USA)
Eric Wohlstadter (University of British Columbia, Canada)

## External Referees

Celeste Campo
Mauro Caporuscio
Alan Colman
Yan Jin
Heiko Ludwig
Johann Oberleitner
Patrizio Pelliccione
Francoise Sailhan
Masato Suzuki
Massimo Tivoli
Jian Yin
Stefanos Zachariadis

## Sponsoring Institutions

International Business Machines (IBM), Armonk, NY, USA

# Table of Contents

## Keynote

## Middleware Services

## Ubiquitous Computing

## Performance and QOS

## Building Distributed Applications

# Dynamic Software Adaptation: Middleware for Pervasive Computing

Gustavo Alonso

Information and Communication Systems Group,
Institute for Pervasive Computing,
Department of Computer Science,
ETH Zürich

**Abstract.** The many different application scenarios found in pervasive and ubiquitous computing have one aspect in common: software will be confronted with continuously changing execution environments. To guarantee seamless service, protocols, infrastructure, and applications will have to be able to adapt to changes in, e.g., networks, system configuration, available resources, varying policies, etc. In other words, adaptation will have to be a key feature of any mobile software system. In this talk I will discuss the problems encountered when designing middleware for pervasive computing and the role that software engineering could play in solving those problems. I will mostly focus on dynamic software adaptation and how it can be used to great effect to provide much more flexible software platforms. The talk will revolve around the work done on the PROSE system, a modified Java Virtual Machine that uses dynamic Aspect Oriented Programming to extend a running application with new functionality as dictated by the context where such an application runs. The extensions are code fragments that transparently adapt the underlying application. The extensions can be used to modified every aspect of the software hierarchy, e.g., they can be use both to replace a routing protocol as well as to change an application's behavior. In the talk, I will discuss the advantage of such an approach as well as the many challenges it poses in terms of software development and maintenance, security, software modularity, and even the accepted perception of what constitutes a software application.

## Biography

Gustavo Alonso is professor in the Department of Computer Science at the Swiss Federal Institute of Technology in Zurich (ETHZ). Gustavo Alonso is from Madrid, Spain, where he completed in 1989 his undergraduate studies in Telecommunications Engineering at the Madrid Technical University (UPM-ETSIT). As a Fulbright student, he did his graduate studies in computer science (M.S. 1992, Ph.D. 1994) in the University of California at Santa Barbara. After graduating, he was a visiting scientist in the IBM Almaden Research Laboratory in San Jose, California, where he worked within the Exotica project in areas

T. Gschwind and C. Mascolo (Eds.): SEM 2004, LNCS 3437, pp.1–2, 2005.

such as workflow management and transaction processing. In September 1995 he joined ETH where he has since then lead several projects in databases, workflow management, replication, and advanced applications. Currently, Gustavo Alonso leads the Information and Communication Systems Research Group. The research interests of the group include Web Services, grid and cluster computing, databases, workflow management, scientific applications of database and workflow technology (for geographic, astronomical, and biochemical data), pervasive computing and dynamic aspect oriented programming. Gustavo Alonso is co-author of a recently published book on Web Services (Springer Verlag, Berlin 2004, ISBN 3-540-44008-9) and has participated in numerous conferences, panels and projects related to the topic. He also regularly works as an independent consultant in areas like enterprise application integration, Web Services, and middleware.

# Here's Your Lego$^{\mathrm{TM}}$ Security Kit: How to Give Developers All Protection Mechanisms They Will Ever Need

Konstantin Beznosov

Department of Electrical and Computer Engineering, University of British Columbia
beznosov@ece.ubc.ca

**Abstract.** By presenting a protection architecture for ASP.NET Web services, this paper demonstrates the feasibility of creating middleware mechanisms in the form of composable, flexible, and extensible building blocks. Like Lego$^{\mathrm{TM}}$ constructor parts, such blocks enable the reduction of the effort of constructing, extending, and adjusting the application properties and middleware services in response to requirements or environment changes.

## 1 Introduction

The main premise of this paper is that the developers and owners of distributed applications need and can be provided with three things: 1) Lego$^{\mathrm{TM}}$-like reusable and versatile building blocks, 2) middleware architectures and tools for composing useful customized solutions out of such blocks, and 3) the means of creating their own inexpensive and error-proof building blocks. They could then create custom distributed applications suitable to their needs and environments, while avoiding costly reinvention and reconstruction of generic and, more often than not, quite complex functionality common across applications. And we are not referring to the business logic, which could arguably be included in the list. The focus is on the nonfunctional properties and services (fault tolerance, performance, security, etc.) of distributed applications.

The above needs have been determined from the author's experience of working for end-user, consulting, and vendor organizations. Working on the end-user side showed that no vendor could ever satisfy all requirements for customizing their solutions to our needs and constraints. Vendors' customization mechanisms required too much effort and expertise from in-house developers. Experience as a consultant, product developer and architect gave convincing evidence that this problem was common to many end-user organizations.

To demonstrate that useful building blocks, architectures, and extension means can indeed be provided for customizing nonfunctional properties of distributed applications without demanding seasoned expertise in the subject matter from application developers, we present an *authentication and authorization (A &A)* architecture for ASP.NET Web services. This architecture, we believe, features all three desired characteristics. It builds on the results of several years of applied research and practical

T. Gschwind and C. Mascolo (Eds.): SEM 2004, LNCS 3437, pp. 3 – 18, 2005.

experience, giving the hope that similar architectures can be developed for easy customization of other properties and services for distributed applications.

The paper is organized as follows: section 2 provides background and discusses related work; Section 3 explains technical motivations for the architecture and gives its overview; Section 4 highlights those design decisions that made the architecture easy to customize; discussion is in Section 5; and we conclude with Section 6.

## 2 Background and Related Work

Research on composition and customization for middleware has been largely focused on three areas: core functionality; domain-specific properties and characteristics; and middleware services. Research in core functionality concentrates on data (un)marshaling, invocation dispatching, object life-cycle, data transport, etc. (TAO [1], Quarterware [2], COMERA [3], Spring [4]). Examples from the work in domain-specific properties and characteristics are real-time [5], load-balancing [6], QoS [7, 8], performance and consistency [9]. Our work is on composable and customizable A&A mechanisms and belongs to middleware services research, which concentrates on such services as event notification [10], transactions and concurrency [11, 12], and security.

Work on customizable security mechanisms in middleware has been conducted at least since DCE [13]. A wider known example is CORBA, which has a Security Service [14] architecture that enables customization by supporting interceptors as well as making authorization and audit decision objects, security context and some other elements replaceable. However, because the granularity of CORBA Security replaceable parts is too coarse it takes too much effort to customize the service. This drawback can also be viewed as low degree of *composability*. Besides DCE and CORBA, other examples of architectures with replaceable security logic but low degree of composability are more modern JAAS [15], Java Authorization Contract for Containers architecture [16], and Legion [17]. Our approach achieves fine granularity of the replaceable parts and therefore a higher degree of composability.

What our approach (intentionally) leaves unanswered is how to express A&A policies and map them into a composition of A&A building blocks. Andersen et al. [18] approach the problem from the other end and propose "programmable security" approach that uses Obol language to "program" middleware security protocols without addressing the issue of translating such programs into compositions of specific elements of the middleware security architecture.

Design of the authorization mechanism described in this paper is largely based on the Resource Access Decision (RAD) architecture [19, 20], which we follow more in the spirit than in detail—rather as an architectural style. Briefly reviewed in Appendix A, RAD is one of the first attempts to compose and customize authorization logic out of simpler parts.

Although, neither RAD nor this work address the issue of conflicts that could arise as a result of authorization logic composition, several solutions have been proposed elsewhere. Jajodia et al. [21] have proposed an access control model in which inconsistencies among authorizations can be resolved using rules. The framework for access control policy enforcement developed by Siewe et al. [22] allows multiple policies to be enforced through policies composition. It provides a way to specify complex policies and to reason about their properties.

## 3 Architecture Motivation and Overview

The ASP.NET container is a popular hosting environment for Web services built and run on Microsoft Windows and .NET platforms. However, the ASP.NET security architecture [23], as provided out-of-the-box is not sufficiently flexible and extensible to be adequate for enterprise applications. As we describe in [24], ASP.NET supports limited authentication and group/user-based authorization, both bound to Microsoft proprietary technologies. If an application needs to be protected with enterprise A&A services, the developers have two options: The first, is to develop home-grown container security extensions, which are hard for average application developers to get right. The second option is to program the security logic into the Web service business logic, but the resulting application is costly to evolve and support. In both cases, the development of security-specific parts by average application developers is commonly believed to result in high vulnerability rates due to security-related bugs that are hard to avoid and catch.

Due to its flexibility and extensibility, the protection architecture described in this paper makes ASP.NET easier to integrate with organizational security infrastructure with a reduced effort on the side of Web service developers. The architecture is flexible because it allows configuring of machine-wide authentication and authorization functions, and overriding them for a subtree of the Web services (up to an individual Web service application) in the directory-based ASP.NET hierarchy. Its extensibility is revealed through the support of wide variety of A&A logic, as long as the logic can be programmed as a .NET class and/or accessed (possibly via a proxy) through a predefined .NET API. Furthermore, one can reuse other instances of such logic by combining authorization decisions from them according to predefined or custom rules.

## 4 The Architecture

The architecture details are described elsewhere [25]. This section focuses mainly on those features of the architecture that enable the composition of more complex A&A functionality from basic, reusable, building blocks. There are five features:

1. the separation of A&A enforcement logic from the decision logic,
2. the employment of the RAD architecture style, which makes creation of custom authorization decision logic easier and avoids the need for a general-purpose policy evaluation engine,
3. flexible configuration-driven construction of the authorization decision information,
4. fine-grained replaceable modules that enable support for a wide range of A&A functionalities, and
5. the support for the scalability, extensibility, and reusability in the configuration part of the architecture.