

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

TU Dortmund University, Germany

Madhu Sudan

Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max-Planck Institute of Computer Science, Saarbruecken, Germany

James Noble Ralph Johnson (Eds.)

Transactions on Pattern Languages of Programming I



Springer

Editors-in-Chief

James Noble
Victoria University of Wellington
School of Engineering and Computer Science
Wellington 6140, New Zealand
E-mail: kjx@ecs.vuw.ac.nz

Ralph Johnson
Siebel Center for Computer Science
201 North Goodwin Avenue, Urbana, IL 61801, USA
E-mail: rjohnson@illinois.edu

Managing Editors

Uwe Zdun
Vienna University of Technology, Information Systems Institute
Argentinierstrasse 8/184-1, 1040 Wien, Austria
E-mail: zdun@infosys.tuwien.ac.at

Eugene Wallingford
University of Northern Iowa, Department of Computer Science
Cedar Falls, IA 50613, USA
E-mail: wallingf@cs.uni.edu

Library of Congress Control Number: 2009942032

CR Subject Classification (1998): D.2, D.3, D.1, K.6, K.6.3

ISSN	0302-9743 (Lecture Notes in Computer Science)
ISSN	1869-6015 (Transactions on Pattern Languages of Programming)
ISBN-10	3-642-10831-8 Springer Berlin Heidelberg New York
ISBN-13	978-3-642-10831-0 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

springer.com

© Springer-Verlag Berlin Heidelberg 2009
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12804610 06/3180 5 4 3 2 1 0

Preface

From conception to birth is traditionally nine months. The first emails regarding the volume you now hold in your hands—or the bits you have downloaded onto your screen—are dated 11 June 2005. From conception to birth has taken over four years.

Springer's LNCS Transactions on Pattern Languages of Programming is dedicated, first and foremost, to promoting, promulgating, presenting, describing, critiquing, interrogating, and evaluating all aspects of the use of patterns in programming. In the 15 years or so since Gamma, Helm, Johnson, Vlissides's *Design Patterns* became widely available, software patterns have become a highly effective means of improving the quality of programming, software engineering, system design, and development. Patterns capture the best practices of software design, making them available to everyone participating in the production of software.

A key goal of the Transactions Series is to present material that is validated. Every contributed paper that appears in this volume has been reviewed by both patterns experts and domain experts, by researchers and practitioners. This reviewing process begins long before the paper is even submitted to Transactions. Every paper in the Series is first presented and critiqued at one of the Pattern Languages of Programming (PLoP) conferences held annually around the world. Prior to the conference, each submitted paper is assigned a shepherd who works with the authors to improve the paper. Based on several rounds of feedback, a paper may proceed to a writers' workshop at the conference itself. In a writers' workshop, pattern authors are given concentrated feedback on their papers in a focused discussion that considers all aspects of the papers, from the most abstract ideas down to the use of whitespace. The papers are revised once again, before being collected in a patterns conference proceedings, often six months after the conference.

Finally, after all these stages of editing, often going back to conferences more than once, the paper may be ready for submission to the *Transactions on Pattern Languages of Programming*. At this stage, every paper receives at least three final reviews before publication. Our reviewers include both experts in patterns and experts from the domains of the patterns, ranging across all the disciplines of software. Only when these reviewers are satisfied that the paper is as good as it can be—perhaps after several further review cycles—are the papers ready for publication. To avoid conflicts of interest, Editors-in-Chief may not submit papers to regular issues of the journal, and similarly Guest Editors of special issues may not submit papers to volumes that they edit. Submissions by associate editors, or by close colleagues of Editors-in-Chief or Guest Editors, are assigned to an editor without conflicts, and the results of the editorial process are only reported back when it is complete. For the record, this first volume was handled primarily by James Noble as Editor-in-Chief, and Uwe Zdun as Associate Editor.

The eight papers in this volume are the first to have emerged from this highly selective reviewing process. In the first paper, Andrew Black and Daniel Vainsencher present a substantial pattern language for constructing an important part of an

integrated development environment: the underlying representation of the program. With a careful analysis of the underlying forces, Andrew and Daniel describe how models can be made extensible, while also performing efficiently and correctly.

The second and third papers are primarily concerned with larger scale systems. Francisco J. Ballesteros, Fabio Kon, Marta Patiño, Ricardo Jiménez, Sergio Arévalo, and Roy H. Campbell describe how systems can be made both more efficient, and more flexible, by batching individual requests between clients and servers, and Titos Saridakis describes how systems can be made more reliable via Graceful Degradation.

While many of the papers in this volume are relatively large, we have also aimed to publish short, focused patterns that address a single problem. In the next paper, Michael J. Pont, Susan Kurian, and Ricardo Bautista-Quintero show how Sandwich Delays can ensure real-time code that always takes exactly the same time to execute.

The last two patterns papers in this volume address service-oriented architectures. Carsten Hentrich presents a small pattern language for synchronization, and then he is joined by Uwe Zdun to describe a second pattern language for process integration and macro/micro workflow design.

Our last two papers do not present new patterns themselves. Rather, they show how patterns can be combined into systems, and then used to document those systems' designs. Mark Mahoney and Tzilla Elrad describe how the State, Interceptor, and Factory Method pattern can be used to model complex state transition systems, and James Siddle shows how the documentation of a complex system can be structured around the patterns in that system's design.

We believe the papers in this volume represent some of the best work that has been carried out in design patterns and pattern languages of programming over the last few years. We thank all the authors for taking the effort to prepare their papers for publication, and our indefatigable reviewers for their many suggestions for improvement. As with any birth, we are justly proud of the result: and we hope that you will enjoy the fruits of our labors!

August 2009

James Noble
Ralph Johnon
Uwe Zdun
Eugene Wallingford

Organization

Editorial Board

Paris Avgeriou	University of Groningen, The Netherlands
Joe Bergin	Pace University, New York, USA
Robert Biddle	Carleton University, Ottawa, Canada
Grady Booch	IBM, USA
Frank Buschmann	Siemens AG, Germany
Jim Coplien	Nordija, Denmark
Ward Cunningham	AboutUS, USA
Jutta Eckstein	Consultant, Germany
Susan Eisenbach	Imperial College London, UK
Richard P. Gabriel	IBM Research, USA
Erich Gamma	IBM, Switzerland
Neil B. Harrison	Utah Valley State College, USA
Kevlin Henney	Curbralan Ltd., UK
Doug Lea	SUNY Oswego, USA
Mary Lynn Manns	University of North Carolina at Asheville, USA
Michael J. Pont	The University of Leicester, UK
Lutz Prechelt	Free University Berlin, Germany
Dirk Riehle	SAP Research, SAP Labs LLC, USA
Mary Beth Rosson	Pennsylvania State University, USA
Andreas Rueping	Consultant, Germany
Doug Schmidt	Vanderbilt University, TN, USA
Peter Sommerlad	Institute for Software at HSR Rapperswil, Switzerland
Jenifer Tidwell	Consultant, USA
Joseph W. Yoder	Consultant, USA

Reviewers

Frank Buschmann	Olaf Zimmermann
Andreas Rüping	Carsten Hentrich
Alessandro Garcia	Michael Stal
Dirk Riehle	Arno Haase
Till Schümmer	Christa Schwanninger
Lise Hvatum	Stefan Hanenberg
Paris Avgeriou	James Siddle
Jonathan Aldrich	Tim Wellhausen
Juha Pärssinen	Jorge Luis Ortega-Arjona
Kristian Eloff Sørensen	Karl Michael Goeschka
Peter Sommerlad	Paul Adamczyk

Michael Weiss
Munawar Hafiz
Neil Harrison
Robert Hanmer
Ademar Aguiar
Stephan Lukosch
Linda Rising

Pavel Hruby
Huy Tran
Tanya Crenshaw
Gabor Kasai
Dick Gabriel
Philipp Leitner
Erich Gamma

Table of Contents

A Pattern Language for Extensible Program Representation	1
<i>Daniel Vainsencher and Andrew P. Black</i>	
Batching: A Design Pattern for Efficient and Flexible Client/Server Interaction	48
<i>Francisco J. Ballesteros, Fabio Kon, Marta Patiño, Ricardo Jiménez, Sergio Arévalo, and Roy H. Campbell</i>	
Design Patterns for Graceful Degradation	67
<i>Titos Saridakis</i>	
Meeting Real-Time Constraints Using “Sandwich Delays”	94
<i>Michael J. Pont, Susan Kurian, and Ricardo Bautista-Quintero</i>	
Synchronization Patterns for Process-Driven and Service-Oriented Architectures	103
<i>Carsten Hentrich</i>	
A Pattern Language for Process Execution and Integration Design in Service-Oriented Architectures	136
<i>Carsten Hentrich and Uwe Zdun</i>	
A Pattern Story for Combining Crosscutting Concern State Machines	192
<i>Mark Mahoney and Tzilla Elrad</i>	
An Example of the Retrospective Patterns-Based Documentation of a Software System	207
<i>James Siddle</i>	
Author Index	231