



The 1998 Pattern Languages
of Programs Conference

PLOP'98 Program Committee, Shepherds and Organizers

PLOP'98 represents the collective effort of the all-stars listed below. We deeply appreciate their dedication to improving the "Quality Without a Name" at PLOP! Next year PLOP'99 will be held between August 17 – 20, 1999 at Robert Allerton Park and Conference Center, the University of Illinois at Urbana-Champaign, Monticello, Illinois, USA. See more information at <http://jerry.cs.uiuc.edu/~plop/plop99>.

Organizers

Program Chair: **Steve Berczuk**, Biztravel.com, Inc. (berczuk@acm.org)

Conference Chair: **Joe Yoder**, University of Illinois, Urbana-Champaign (j-yoder@uiuc.edu)

Program Committee

Steve Berczuk (berczuk@acm.org)
Bill Opdyke (wopdyke@lucent.com)
Eugene Wallingford (wallingf@cs.uni.edu)
Joe Yoder (j-yoder@uiuc.edu)
Jens Coldewey (jens_coldewey@acm.org)
Ward Cunningham (ward@c2.com)
Bob Hanmer (r.s.hanmer@lucent.com)
Bobby Woolf (woolf@acm.org)

Shepherds

Aamod Sane (sane@iconcomp.com)
Alan O'Callaghan (100303.2324@compuserve.com)
Alejandra Garrido (garrido@chip.cs.uiuc.edu)
Antonio Rito Silva (Rito.Silva@acm.org)
Bill Opdyke (wopdyke@lucent.com)
Bob Hanmer (r.s.hanmer@lucent.com)
Bobby Woolf (woolf@acm.org)
Brad Appleton (bradapp@enteract.com)
Chris Cleeland (cleeland@cs.wustl.edu)
Christa Schwanninger (christa.schwanninger@mchp.siemens.de)
David DeLano (delanod@agcs.com)
Dennis DeBruier (debruler@lucent.com)
Dirk Riehle (riehle@acm.org)
Don Olson (olsond@agcs.com)
Don Roberts (droberts@cs.uiuc.edu)
Doug Lea (dl@cs.oswego.edu)
Erich Gamma (erich_gamma@acm.org)
Eugene Wallingford (wallingf@cs.uni.edu)

Frank Buschmann (Frank.Buschmann@mchp.siemens.de)
Gerard Mezaros (gerard.mezaros@acm.org)
James Noble (kix@mpce.mq.edu.au)
Jens Coldewey (jens_coldewey@acm.org)
John Goodsen (jgoodsen@saguarosoft.com)
John Vlissides (vlis@watson.ibm.com)
Joshua Kerievsky (joshua@industriallogic.com)
Ken Auer (kauer@rolemodelsoft.com)
Kent Beck (Kent.Beck@cslife.ch)
Kyle Brown (kyle.brown@acm.org)
Linda Rising (risingl@agcs.com)
Lizette Velazquez (lizette@lucent.com)
Mark Bradac (bradac@lucent.com)
Michael A. Beedle (beedlem@fti-consulting.com)
Michael Stal (michael.stal@mchp.siemens.de)
Michel de Champlain (michel@ece.concordia.ca)
Neil Harrison (nbharrison@lucent.com)
Norm Kerth (nkerth@teleport.com)
Peter Sommerlad (sommerlad@ifa.ch)
Ralph Johnson (johnson@cs.uiuc.edu)
Richard Helm (helm@raleigh.ibm.com)
Robert C. Martin (rmartin@oma.com)
Robert Hirschfeld (hirschfeld@acm.org)
Robert Orenstein (rlo@perforce.com)
Steve Berczuk (berczuk@acm.org)
Todd Coram (tcoram@pobox.com)
Ward Cunningham (ward@c2.com)
Wolf Siberski (daasb@rwg.de)
Wolfgang Keller (WofgangWKeller@compuserve.com)

Staff

Brian Foote (bfoote@uiuc.edu)
David Anderson (deandrsn@uiuc.edu)
Don Roberts (droberts@cs.uiuc.edu)
Dragos Manolescu (manolesc@uiuc.edu)
Ed Peters (elpeters@uiuc.edu)
Ian Chai (chai@uiuc.edu)
Lewis Muir (jl-muir@uiuc.edu)
Paul Rubel (rubel@crhc.uiuc.edu)
Peter Hatch (p-hatch@uiuc.edu)
Ping Liu (pingliu@uiuc.edu)
Weerasak Witthawaskul (witthawa@uiuc.edu)



Agenda

Time	Events	Other Activities
Tuesday August 11		
17:00 – 21:00	Registration	
17:00 – 21:00	Pizza	
19:00 – 19:45	Games	
Wednesday August 12		
7:30 – 8:30	Breakfast	
8:35 – 8:55	Reading time	
9:00 – 9:55	Greeting, Demo Writers Workshop	
10:00 – 10:40	Games	
10:45 – 10:55	Break	
11:00 – 11:30	Writers Workshop Organization Session	
11:35 – 11:55	Nap/Reading Time	
12:00 – 13:00	Lunch	
13:05 – 14:15	Writers Workshop 1	
14:20 – 14:55	Games	
15:00 – 16:10	Writers Workshop 2	
16:15 – 16:55	Nap/Reading Time	<i>Patterns Book Shop Open 4 PM to 8 PM</i>
17:00 – 18:25	Dinner	
18:30 – 20:25	BoFs, Working Groups	
20:30 – 22:30	Entertainment	

Time	Events	Other Activities
Thursday August 13		
7:30 – 8:40	Breakfast	
8:45 – 9:55	Writers Workshop 3	
10:00 – 10:25	Games	
10:30 – 10:45	Break	
10:50 – 11:55	Writers Workshop 4	
12:00 – 13:00	Lunch	
13:05 – 13:55	Nap/Reading Time	
14:00 – 15:10	Writers Workshop 5	
15:15 – 15:25	Break	
15:30 – 16:40	Writers Workshop 6	<i>Patterns Book Shop Open 4 PM to 8 PM</i>
16:45 – 17:25	Nap/Reading Time	
17:30 – 18:55	Dinner	
19:00 – 20:25	Misc. Sessions	
20:30 – 22:30	Entertainment	<i>(Tentative) Book Signing</i>
Friday August 14		
7:30 – 8:40	Breakfast	
8:45 – 9:55	Writers Workshop 7	
10:00 – 10:25	Games	
10:30 – 10:55	Break/Checkout time	
11:00 – 11:55	Closing Session & Games	
12:00 – 13:00	Lunch	

Note: See detail on some activities on the next page.

Details on Some Activities

Patterns Book Shop

Borders Book Store will set up a table where you can buy a selection of books on patterns, including some of Christopher Alexander's books.

Book Signing

Bring your books on software patterns for authors to sign. Check the [registration list](#) to see which authors will be at the conference.

PLoP'98 Proceedings by Group

Group 1 Agricultural Valleys

Lead moderator: Wolfgang Keller

- 1.1 Time Patterns
[Manfred Lange](#)
- 1.2 Temporal Patterns
[Andy Carlson](#)
- 1.3 Some Patterns for Insurance Systems
[Wolfgang Keller](#)
- 1.4 Application Scenario - A Pattern Language For Business Process Control
[S. Ramesh](#)
- 1.5 The Emissary Design Pattern
[Ramiro Gonzalez Maciel](#)
- 1.6 A Collection of History Patterns
[Francis Anderson](#)
(Review page 19-32)

Group 3 Zen View

Lead moderator: Linda Rising

- 3.1 Identify the Champion
[Oscar Nierstrasz](#)
- 3.2 Customer Interaction Patterns
[Linda Rising](#)
- 3.3 User Interface Software
[Jens Coldewey](#)
- 3.4 Interaction Patterns
[Jennifer Tidwell](#)
- 3.5 Telecommunications Input and Output Pattern Language
[Bob Hanmer](#) and [Greg Stymfal](#)
- 3.6 Patterns for Interactive Applications
[William C. Wake](#)
- 3.7 Patterns for Dynamic Websites
[Fernando Lyardet](#) and [Gustavo Rossi](#)

Group 2 Network of Learning

Lead moderator: Steve Berczuk

- 2.1 Courier Patterns
[Robert Switzer](#)
(Review page 1-10)
- 2.2 Encourage Piecemeal Growth
[Nicholas Jacobs](#)
- 2.3 Streamed Lines: Branching Patterns for Parallel Software Development
[Brad Appleton](#), et al.
- 2.4 Override Current Processing
[Philip Eskelin](#)
- 2.5 SCRUM: An extension pattern language for hyperproductive software development
[Michael A. Beedle](#), et al.
- 2.6 Patterns for Productivity
[Paul Taylor](#)
(Review pattern 3-6 and 9-11)

Group 4 Four-Story Limit

Lead moderator: Robert Hirschfeld

- 4.1 Interaction Patterns for Communicating Processes
[Ted Faison](#)
- 4.2 Foundation Patterns
[Dwight Deugo](#)
- 4.3 Essence Pattern
[Andy Carlson](#)
- 4.4 Lock Server
[Robert Hirschfeld](#) and [Jeff Eastman](#)
- 4.5 Convenience Methods
[Robert Hirschfeld](#) and [Jeff Eastman](#)
- 4.6 Notification Server
[Robert Hirschfeld](#) and [Jeff Eastman](#)
- 4.7 A Pattern Language of Statecharts
[Sherif Yacoub](#) and [Hany H. Ammar](#)

Group 5 Mosaic of Subcultures

Lead moderator: Joe Yoder

- 5.1 Feature Extraction - A Pattern for Information Retrieval
Dragos Manolescu
- 5.2 A Confederation of Patterns for Resource Management
Rosana T. Vaccare Braga, Fernao S. R. Germano and Paulo Cesar Masiero
- 5.3 Shared Repository
Philippe Lalanda
- 5.4 Using Design Patterns to Develop a Hyper-controllable Medical Image Application
K.Y. Chang and Lih-Shyang Chen
- 5.5 Structured Matcher
Eugene Wallingford
- 5.6 The HiStar Pattern Language for Hierarchical Query
F Nelson Loney
- 5.7 Connecting Business Objects to Relational Databases
Joe Yoder
(Review Introduction and first 5 patterns)

Group 6 Site Repair

Lead moderator: Hans Rohnert

- 6.1 Display Maintenance
Dwayne Towell
- 6.2 Microthread
Joe Hoffert and Kenneth Goldman
- 6.3 Triggered Placeholder - An Object Behavioral Pattern for Delaying Object Creation
Joe Hoffert
- 6.4 A Componentware Development Methodology based on Process Patterns
Klaus Bergner, et al.
- 6.5 Cascading Bridge Design Pattern
Brendan McCarthy
- 6.6 Virtual Singleton
Gerard Mezaros

Group 7 Scattered Work

Lead moderator: Bobby Woolf

- 7.1 Philosophical Patternin Software Development
Ken Auer
- 7.2 The Object Recursion Pattern
Bobby Woolf
- 7.3 An Alternative Solution for the Observation Pattern Problem
Kinh Nguyen
- 7.4 Fundamental Elements of an Extendible Java Framework
Ken Auer
- 7.5 Refining the Observer Pattern: The Middle Observer Pattern
Pablo Iaria and Ulises Chesini
- 7.6 Telephony Data Handling Pattern Language
David DeLano
- 7.7 Loop Patterns
Owen Astrachan and Eugene Wallingford

Group 8 Eccentric Nucleus

Lead moderator: Brian Foote

- 8.1 Composite as Metamodel: a Design Pattern for Implementing SIGOBT-style Object Wrappers for HL7 Messages
Frederick KOH
- 8.2 Tropec: A Pattern Language for Cryptographic Software
Alexandre Melo Braga, et al.
(Review page 1-15)
- 8.3 Composing Multimedia Artifacts for Reuse
Jacob L. Cybulski and Tanya Linden
(Review page 3-13)
- 8.4 The Reflective State Pattern
Luciane Lamour Ferreira and Cecilia M. F. Rubira
(Review page 1-12)
- 8.5 Lingua Franca
Brian Foote and Don Roberts
- 8.6 Metadata and Active Object-Models
Brian Foote and Joe Yoder
(Review page 1-15)

PLoP'98 Proceedings by Author

❖ A ❖

- 1.6 A Collection of History Patterns
Francis Anderson
(Review page 19-32)
- 2.3 Streamed Lines: Branching Patterns for
Parallel Software Development
Brad Appleton, et al.
- 7.7 Loop Patterns
Owen Astrachan and Eugene Wallingford
- 7.1 Philosophical Patterns in Software
Development
Ken Auer
- 7.4 Fundamental Elements of an Extendible
Java Framework
Ken Auer

❖ B ❖

- 2.5 SCRUM: An extension pattern language for
hyperproductive software development
Michael A. Beedle, et al.
- 6.4 A Componentware Development
Methodology based on Process Patterns
Klaus Bergner, et al.
- 5.2 A Confederation of Patterns for Resource
Management
*Rosana T. Vaccare Braga, Fernao S. R.
Germano and Paulo Cesar Masiero*
- 8.2 Tropyc: A Pattern Language for
Cryptographic Software
Alexandre Melo Braga, et al.

❖ C ❖

- 1.2 Temporal Patterns
Andy Carlson
- 4.3 Essence Pattern
Andy Carlson
- 5.4 Using Design Patterns to Develop a Hyper-
controllable Medical Image Application
K.Y. Chang and Lih-Shyang Chen
- 3.3 User Interface Software
Jens Coldewey
- 8.3 Composing Multimedia Artifacts for Reuse
Jacob L. Cybulski and Tanya Linden

❖ D ❖

- 7.6 Telephony Data Handling Pattern Language
David DeLano
- 4.2 Foundation Patterns
Dwight Deugo

❖ E ❖

- 2.4 Override Current Processing
Philip Eskelin

❖ F ❖

- 4.1 Interaction Patterns for Communicating
Processes
Ted Faison
- 8.4 The Reflective State Pattern
*Luciane Lamour Ferreira and Cecilia M. F.
Rubira*
- 8.5 Lingua Franca
Brian Foote and Don Roberts
- 8.6 Metadata and Active Object-Models
Brian Foote and Joe Yoder

❖ H ❖

- 3.5 Telecommunications Input and Output
Pattern Language
Bob Hanmer and Greg Stymfal
- 4.4 Lock Server
Robert Hirschfeld and Jeff Eastman
- 4.5 Convenience Methods
Robert Hirschfeld and Jeff Eastman
- 4.6 Notification Server
Robert Hirschfeld and Jeff Eastman
- 6.2 Microthread
Joe Hoffert and Kenneth Goldman
- 6.3 Triggered Placeholder - An Object
Behavioral Pattern for Delaying Object
Creation
Joe Hoffert

❖ I ❖

- 7.5 Refining the Observer Pattern: The Middle
Observer Pattern
Pablo Iaria and Ulises Chesini

❖ J ❖

- 2.2 Encourage Piecemeal Growth
[Nicholas Jacobs](#)

❖ K ❖

- 1.3 Some Patterns for Insurance Systems
[Wolfgang Keller](#)
- 8.1 Composite as Metamodel: a Design Pattern for Implementing SIGOBT-style Object Wrappers for HL7 Messages
[Frederick KOH](#)

❖ M ❖

- 6.5 Cascading Bridge Design Pattern
[Brendan McCarthy](#)
- 6.6 Virtual Singleton
[Gerard Mezaros](#)

❖ L ❖

- 5.3 Shared Repository
[Philippe Lalanda](#)
- 1.1 Time Patterns
[Manfred Lange](#)
- 5.6 The HiStar Pattern Language for Hierarchical Query
[F Nelson Loney](#)
- 3.7 Patterns for Dynamic Websites
[Fernando Lyardet](#) and [Gustavo Rossi](#)

❖ M ❖

- 1.5 The Emissary Design Pattern
[Ramiro Gonzalez Maciel](#)
- 5.1 Feature Extraction - A Pattern for Information Retrieval
[Dragos Manolescu](#)

❖ N ❖

- 7.3 An Alternative Solution for the Observation Pattern Problem
[Kinh Nguyen](#)
- 3.1 Identify the Champion
[Oscar Nierstrasz](#)

❖ R ❖

- 1.4 Application Scenario - A Pattern Language For Business Process Control
[S. Ramesh](#)
- 3.2 Customer Interaction Patterns
[Linda Rising](#)

❖ S ❖

- 2.1 Courier Patterns
[Robert Switzer](#)

❖ T ❖

- 2.6 Patterns for Productivity
[Paul Taylor](#)
(Review pattern 3-6 and 9-11)
- 3.4 Interaction Patterns
[Jenifer Tidwell](#)
- 6.1 Display Maintenance
[Dwayne Towell](#)

❖ W ❖

- 3.6 Patterns for Interactive Applications
[William C. Wake](#)
- 5.5 Structured Matcher
[Eugene Wallingford](#)
- 7.2 The Object Recursion Pattern
[Bobby Woolf](#)

❖ Y ❖

- 4.7 A Pattern Language of Statecharts
[Sherif Yacoub](#) and [Hany H. Ammar](#)
- 5.7 Connecting Business Objects to Relational Databases
[Joe Yoder](#)

Writers Workshop Guidelines

This document explains how writer workshops should work at PLoP '98. The process is basically the same as previous conferences, but there are a few differences. One difference is the manifestation of the *Strong Moderator* role. The workshops need to balance the exchange of ideas with the ability to fit all the parts of the workshop process into a workshop session, and the ability to fit all the workshop sessions into a conference. In the description of the workshop process below time limits are assigned to each section. The limits are only recommendations, but past experience has demonstrated that sessions which vary too much from these limits tend not to fit everything in, and the author typically gets cheated out of their chance to ask for clarification. Workshop groups are free to continue the workshoping of a paper during some of the free sessions.

For more information on writers' workshops also see these the next section on conducting a writers workshop.

Roles

- **Lead Moderator.** The Lead moderator will be chosen by the program chair, in advance, from among the participants in a workshop group. This person will be someone who has experience with the writers workshop format. The Lead moderator will run the logistics session, and moderate the first paper. The role of session moderator can rotate among the persons in the group. The Lead moderator is responsible for reminding the session moderator to keep a session on track.
- **Session Moderator.** The Session Moderator runs the workshop according to the Workshop Process. The moderator is responsible for making sure that participants adhere to the format (i.e., no criticism during the "positive feedback" session), to make sure that the discussion remains focused (i.e., limiting the number of times and ways the various participants restate essentially the same point), and also making sure that the workshop stays on schedule so that the author has enough time to request feedback.
- **Author.** The author's role is to listen to the other workshop participants.
- **Workshop Participants** Fellow Authors follow the [Workshop Process](#), providing constructive feedback to the author.
- **Visitors** Non-Authors observe, and at the discretion of the author group participant in the [Workshop Process](#). The reason for this participation being discretionary is to create an environment where the authors are subject to feedback, in addition to providing feedback.

Logistics Session

Each workshop session will be slightly more than 1 hour in length. There will be a special 30 minute session at the beginning of the conference so that each workshop group can be introduced, and work out logistics, such as how much they want non authors to participate and in which order the papers will be presented.

The Workshop Process

Each workshop session will be 1 hour long. The writers workshop format has proven to be useful in past PLoP conferences, and should be followed by each group.

- *Introduction/Reading:* Moderator introduces the Author and the Author reads a selection from the paper. This is the last we hear of the author til the end. (allow 5 minutes)
- *Summary*One of the workshop participants summarizes the paper.(5 minutes)
- *Positive Feedback:* Moderator asks for things people liked about the patterns. The comments can be about presentation or content, and at the discretion of the moderator comments about presentation and content can be intermingled, or done separately. (Allow 15 minutes)
- *Constructive Criticism:*Moderator asks for ways in which the paper can be improved, both in content and presentation. (Allow 20-25 minutes)
- *Author Feedback:*The author asks for clarification on comments made during the session. The Author should pick a few of the most important points. (or ones which were made by the most people.) Further clarification can be had during off line discussions. (Allow 10 minutes)
- *Closing:* The workshop participants thank the author.

Remember that while the writers workshop format may seem unfamiliar, it has been shown to be useful for developing an environment where patterns authors can share their ideas.

How to Hold a Writer's Workshop

This document represents the collective work of various people (e.g., Frank Buschmann, Ralph Johnson, Jim Coplien, Linda Rising, David Delano, Erich Gamma, and Doug Schmidt) in the [patterns community](#). Jim Coplien has also written down [patterns](#) for writer's workshops.

Structure of a Writer's Workshop

The writers workshop format is a particularly effective method to review, evaluate, and improve pattern descriptions. The general structure of a writers workshop has a group of "discussants" read the paper carefully before the session. During the workshop the discussants examine the strengths and weaknesses of each paper, accentuating positive aspects and suggesting improvements in content and style.

Although the author is present, he or she remains "invisible" during most of the discussion. The author is expected to take notes and/or have someone take notes for them during the discussion (so they can concentrate on the discussion). Many reviewers also give their marked-up copy of the paper to the authors with further written comments. These comments are intended to help the author improve the paper, but the author is not obliged to follow all the suggestions. The entire process normally takes about an hour per paper.

Within a writers workshop session, papers are discussed in several "rounds" according to the following format:

1. The paper is discussed by a group of people including its author, a moderator, and a group of reviewers who are familiar with the contents of the paper.
2. The author of a paper reads a paragraph of his/her choice. The goal is to let the author express what he or she feels is particularly important about the pattern, as well as to give participants a change to get to know the author a bit.
3. One or two reviewers briefly summarize the paper from their personal viewpoint. The goal is to identify what the reviewer(s) thought were the key points of the pattern. Since the other participants should already have read the work, the summaries should be concise. In particular, it's best to avoid debating any inconsistencies between different reviewer's interpretations of the paper at this point.
4. The group then discusses what they liked about the paper, first in terms of content and then in terms of style. The goal is to identify and praise the strengths of the work.
5. After presenting the positive aspects of the paper, the group discusses how to improve the content and style of the paper. The goal here is *not* to criticize the paper per se, but rather to give the author constructive suggestions on how to make the paper better. In general, the style for critical comments is to first state the problem followed by a suggestion on how to solve the problem.
6. After this discussion, the author of the paper may ask questions of the reviewers to clarify their statements. The goal is to give the author a chance to better understand certain comments, rather than to defend the paper.
7. The session closes with the audience thanking the author for writing the paper.

Note that during the rounds 3, 4, and 5 the author of the paper is only "virtually" present. He or she does not actively participate in the discussion. Moreover, the reviewers do not address him or her directly, i.e., the reviewers discuss the paper as if its author is not present. In particular, the reviewers should refer to "the author" in the third person and should not look at the author when making comments.

Suggestions for a Successful Writer's Workshop

The following are some suggestions for holding a successful writer's workshop. The purpose of these suggestions are to foster creativity and sharing of ideas and insights among the participants and to respect and appreciate the contribution of the authors.

- 1. Ensuring appropriate atmosphere** -- It is very important that workshop participants act professionally and courteously towards each other and towards the author during a workshop. The goal is to ensure that participants feel comfortable sharing their insights and suggestions for improvement. We all feel uneasy when we are being evaluated, and so authors will feel nervous under the best of circumstances. If one of the workshop attendees is very critical then the author will feel so bad he or she will probably never want to come back.
It is the responsibility of the moderators (and particularly the lead moderator) to ensure the atmosphere of the workshop is constructive and conducive to insightful discussions, rather than having people show off their intellect by attacking other people. Moreover, it's very important to stress positive aspects of the paper before presenting (constructive) criticisms.
- 2. Contributions by non-authors** -- Moderators should ask each author whether they would like to include or exclude non-authors from commenting on their paper during their workshop review. In addition, moderators should let authors comment on the paper being reviewed before asking for non-author comments. The goal is to make sure that those who have contributed their effort to write papers for the workshop are recognized accordingly.
We've found that non-authors usually have good things to contribute, and authors would lose a lot if they didn't get their input. However, we realize that most of the authors do not have a lot of writing experience. They are expert developers, not expert writers, so we are sympathetic and try to be as helpful as we can be. The authors are sharing their hard-earned experience with us, and we want to demonstrate our appreciation to them.
- 3. Circle organization** -- When possible, the writer's workshops should be organized with the authors in the workshop sitting in the inner circle, and non-authors sitting in the outer circle. The goal is to recognize the authors, who have contributed their efforts to making the workshop possible, and to encourage the authors to contribute to the discussion. Naturally, if the room isn't large enough to accommodate this, then everyone should sit in a single larger circle.
- 4. Rotating moderators** -- It has been common practice at past PLoPs to rotate moderators among authors in writer's workshops. The goal is to give authors the opportunity to gain experience moderating a writer's workshop. It is the responsibility of the lead moderator for each session to help other moderators if questions arise.
- 5. Workshop size** -- To avoid overcrowding please limit the number of non-authors in a writer's workshop to around 10. The goal is to disperse the non-authors relatively evenly throughout the 7 workshops. Since we have almost 80 authors at PLoP '96 (and about 110 attendees) this should be fairly easy to do since the average number of non-authors will be about 5 per session.

Call for Submissions

Pattern Languages of Program Design 4 (PLoPD 4)¹

The long awaited fourth volume in the Pattern Languages of Program Design series will be published in the spring of 1999. This series of books has provided a large body of patterns to the software community, and is helping to reshape the way software is developed.

Submissions are solicited from works which have been workshopped at one of the PLoP conferences, namely:

- Patterns workshopped at PLoP '98 or EuroPLoP '98.
- Patterns workshopped at PLoP '97 or EuroPLoP '97.
- Patterns workshopped at 1996 and earlier PLoPs or EuroPLoPs which have not been published.

Dates

The due date for submissions is Monday, 5 October 1998. Authors will be notified of acceptance or rejection by Monday, 21 December 1998.

Submission Guidelines

There is no fixed length requirement. Most patterns published in previous PLoPD volumes have been 20 pages or less. Submissions may be in MS Word, HTML, PDF, or PostScript. Final submissions to Addison-Wesley Longman must be in MS Word format, as per their guidelines.

Submission Instructions

We expect to begin accepting submissions shortly after PLoP '98 USA. Detailed submission instructions will be posted here once they are available.

Selection Criteria

Papers will be reviewed by members of an editorial committee, who will submit their recommendations to the editorial board. We also plan to take prior PLoP program committee review materials into account. The editorial board will make the final decisions about what submissions are included in the PLoPD4 book.

Our criteria will include:

- Overall quality of the submission, including technical soundness and literary excellence.
- Groundbreaking nature of the submission. In other words, new patterns will be favored over minor variations on previously published patterns.
- "Fit" with other patterns. A pattern that fits together with the other patterns for the book will tend to be favored over a pattern on a totally different topic.

Because of the large number of submissions expected, we expect that we will not be able to accept all the high quality patterns that are submitted. Submissions that are rejected will remain in the "pattern pool", and might be considered for later PLoPD or topical patterns books. Of course, authors retain their rights to such materials, and are free to seek other outlets for their publication.

The decision of the editorial board is final.

¹ To be published by Addison-Wesley Longman as part of the Software Patterns Series.

Suggestions for Improving Your Chances of Acceptance

We stress two requirements above all others: technical soundness and literary quality. Papers that are not technically sound or are of dubious quality will not be accepted. Also note that the adage "In dubio pro reo"² does not hold here. Your writing must convince the reviewers and the editors of the technical quality of your patterns. Of course even if your paper is technically superior, it may be rejected if it is hard to read. If, on the other hand, your paper is carefully written but technically flawed or shallow, it will be another good candidate for rejection.

If you are a non-native speaker make sure that your English is grammatically correct. Also, do your best to write in a clear and unpretentious style. Of course, avoid bland, dull, soporific presentations as well. If you can, give your paper to a native speaker before submitting it.

One of the best ways to learn how to write successful patterns is to read successful patterns. Therefore, look at the previous PLoP volumes, and the landmark pattern books, such as the Design Patterns (GoF) and Pattern-Oriented Software Architecture (POSA) volumes. Do not limit yourself to technical material. Alexander's A Pattern Language presents patterns from the realm of building in a clear, engaging, erudite fashion.

A fine place to start is A Pattern Language for Pattern Writing by Gerard Meszaros and Jim Doble, which simultaneously presents patterns for writing patterns, and examples of their use. This paper appeared in the PLoPD3 book.

We do not mandate a single "pattern form" for patterns. Authors are left to decide which form best suits their material. See the GoF or POSA books for two examples of how to format your patterns. Some authors prefer a more literary "Alexanderian style", modeled after Christopher Alexander's A Pattern Language. The editors may elect to impose a single form on papers that use similar forms.

Don't neglect the judicious inclusion of pictures and diagrams. They can make a submission much more appealing and comprehensible. Just as a few well-chosen diagrams can make the difference between an arcane work and a widely useful one, they can also make the difference between acceptance and rejection. Of course, too many such ornaments can clutter your work. Also, keep in mind that copyright permissions must be obtained in order to publish materials owned by others.

Note that there will be little time for refinement, and no time for shepherding after you submit your paper. The reviewers will only be asked to give a verdict, not to give you suggestions on how to improve the paper. Rather, we assume that you will have incorporated the feedback you got in the writer's workshop to polish your paper. Consider these suggestions carefully. For instance a suggestion for a local improvement may apply to other parts of your paper as well. If a reader asks for an example to clarify the forces in one of your patterns, this may apply to your other patterns as well.

Finally, a word on the length of your submission. Although, we do not set hard limits on the length of submissions, please be considerate. An often employed threshold for books is 500 pages and your paper will be one of many. So, we will tend to favor shorter papers over longer ones. If you really want to submit a long paper, you had better make sure that it is of outstanding importance, since it might be passed over simply because of its length.

The PLoPD-4 Editors:

Neil Harrison
Brian Foote
Hans Rohnert

Addison-Wesley Longman Software Patterns Series Editor:
John Vlissides

² The benefit of the doubt goes to the accused.

Group 1 Agricultural Valleys

Lead moderator: Wolfgang Keller

- 1.1 Patterns
Manfred Lange
- 1.2 Temporal Patterns
Andy Carlson
- 1.3 Some Patterns for Insurance Systems
Wolfgang Keller
- 1.4 Application Scenario - A Pattern Language For Business Process Control
S. Ramesh
- 1.5 The Emissary Design Pattern
Ramiro Gonzalez Maciel
- 1.6 A Collection of History Patterns
Francis Anderson
(Review page 19-32)

Group 2 Network of Learning

Lead moderator: Steve Berczuk

2.1 Courier Patterns

[Robert Switzer](#)

(Review page 1-10)

2.2 Encourage Piecemeal Growth

[Nicholas Jacobs](#)

2.3 Streamed Lines: Branching Patterns for Parallel Software Development

[Brad Appleton, et al.](#)

2.4 Override Current Processing

[Philip Eskelin](#)

2.5 SCRUM: An extension pattern language for hyperproductive software development

[Michael A. Beedle, et al.](#)

2.6 Patterns for Productivity

[Paul Taylor](#)

(Review pattern 3-6 and 9-11)

Group 3 Zen View

Lead moderator: Linda Rising

- 3.1 Identify the Champion
Oscar Nierstrasz
- 3.2 Customer Interaction Patterns
Linda Rising
- 3.3 User Interface Software
Jens Coldewey
- 3.4 Interaction Patterns
Jenifer Tidwell
- 3.5 Telecommunications Input and Output Pattern Language
Bob Hanmer and Greg Stymfal
- 3.6 Patterns for Interactive Applications
William C. Wake
- 3.7 Patterns for Dynamic Websites
Fernando Lyardet and Gustavo Rossi

Group 4 Four-Story Limit

Lead moderator: Robert Hirschfeld

- 4.1 Interaction Patterns for Communicating Processes
Ted Faison
- 4.2 Foundation Patterns
Dwight Deugo
- 4.3 Essence Pattern
Andy Carlson
- 4.4 Lock Server
Robert Hirschfeld and Jeff Eastman
- 4.5 Convenience Methods
Robert Hirschfeld and Jeff Eastman
- 4.6 Notification Server
Robert Hirschfeld and Jeff Eastman
- 4.7 A Pattern Language of Statecharts
Sherif Yacoub and Hany H. Ammar

Group 5 Mosaic of Subcultures

Lead moderator: Joe Yoder

- 5.1 Feature Extraction - A Pattern for Information Retrieval
Dragos Manolescu
- 5.2 A Confederation of Patterns for Resource Management
Rosana T. Vaccare Braga, Fernao S. R. Germano and Paulo Cesar Masiero
- 5.3 Shared Repository
Philippe Lalanda
- 5.4 Using Design Patterns to Develop a Hyper-controllable Medical Image Application
K.Y. Chang and Lih-Shyang Chen
- 5.5 Structured Matcher
Eugene Wallingford
- 5.6 The HiStar Pattern Language for Hierarchical Query
F Nelson Loney
- 5.7 Connecting Business Objects to Relational Databases
Joe Yoder
(Review Introduction and first 5 patterns)

Group 6 Site Repair

Lead moderator: Hans Rohnert

- 6.1 Display Maintenance
Dwayne Towell
- 6.2 Microthread
Joe Hoffert and Kenneth Goldman
- 6.3 Triggered Placeholder - An Object Behavioral Pattern for Delaying Object Creation
Joe Hoffert
- 6.4 A Componentware Development Methodology based on Process Patterns
Klaus Bergner, et al.
- 6.5 Cascading Bridge Design Pattern
Brendan McCarthy
- 6.6 Virtual Singleton
Gerard Mezaros

Group 7 Scattered Work

Lead moderator: Bobby Woolf

- 7.1 Philosophical Patterns in Software Development
Ken Auer
- 7.2 The Object Recursion Pattern
Bobby Woolf
- 7.3 An Alternative Solution for the Observation Pattern Problem
Kinh Nguyen
- 7.4 Fundamental Elements of an Extendible Java Framework
Ken Auer
- 7.5 Refining the Observer Pattern: The Middle Observer Pattern
Pablo Iaria and Ulises Chesini
- 7.6 Telephony Data Handling Pattern Language
David DeLano
- 7.7 Loop Patterns
Owen Astrachan and Eugene Wallingford

Group 8 Eccentric Nucleus

Lead moderator: Brian Foote

- 8.1 Composite as Metamodel: a Design Pattern for Implementing SIGOBT-style Object Wrappers for HL7 Messages
Frederick KOH
- 8.2 Tropyc: A Pattern Language for Cryptographic Software
Alexandre Melo Braga, et al.
(Review page 1-15)
- 8.3 Composing Multimedia Artifacts for Reuse
Jacob L. Cybulski and Tanya Linden
(Review page 3-13)
- 8.4 The Reflective State Pattern
Luciane Lamour Ferreira and Cecilia M. F. Rubira
(Review page 1-12)
- 8.5 Lingua Franca
Brian Foote and Don Roberts
- 8.6 Metadata and Active Object-Models
Brian Foote and Joe Yoder
(Review page 1-15)