Workshop on Design Patterns for Concurrent, Parallel and Distributed Object-Oriented Systems

Organized By:

Dennis Kafura
Virginia Tech
kafura@cs.vt.edu

Greg Lavender
ISODE Consortium
G.Lavender@isode.com

Doug Schmidt
Washington University
schmidt@cs.wustl.edu

1. Overview

The workshop was organized to stimulate and focus research on design patterns for concurrent, parallel, and distributed (CPD) object-oriented systems. Workshop participants spent a day examining design patterns that capture the static and dynamic structures of successful solutions to problems known to arise when building CPD systems. The goal of the workshop was to document and collect common design patterns to facilitate the practical construction of CPD systems.

The emerging focus on design patterns in the object-oriented community offers developers of CPD systems both a language of discourse and a conceptual framework for capturing the essence of successful architectures, components, policies, services, and programming mechanisms. Once expressed in the pattern form, CPD solutions may be recast in new contexts to facilitate the widespread reuse of (micro-)architectures, detailed designs, algorithms, and implementations.

Despite dramatic increases in network speeds and host processing power, the design and implementation of CPD systems remains a challenging problem. Moreover, the growing heterogeneity of hardware/software architectures and diversity of operating system platforms make it increasingly difficult to directly reuse existing algorithms, detailed designs, interfaces, or implementations.

1.1. Suggested Paper Topics

The suggested topics for papers included:

- Identifying reusable patterns of design for constructing flexible synchronization mechanisms.
- Identifying reusable patterns of design for constructing robust, extensible, and efficient communication protocols and services.
- Identifying reusable patterns of design for constructing reliable distributed objects.
- Identifying reusable patterns of design for constructing groups of collaborating objects.
- Identifying patterns of design that enable the construction of higher order language mechanisms for concurrent, distributed, and parallel programming (e.g., active objects, message passing, actors, meta-object protocols, etc).

1.2. Focus Issues

Specific issues that authors were asked to focus on included:

- What types of patterns emerge when separating policy from mechanism? Can patterns of design be captured in mechanisms such that the mechanisms are parameterized by particular policies?
- What patterns arise when defining taxonomies for synchronization mechanisms? Can a standard library of reusable components be defined from which a wide-class of synchronization mechanisms can be constructed?
- What types of patterns emerge in CPD systems, where efficiency concerns are often paramount, but

where abstraction and efficiency are often in conflict?

- What types of patterns emerge to support micro-architectures for common services and mechanisms required in distributed OO systems; for example: name services, exceptions, and event loops.

Participation in the workshop was open to anyone with an interest in advancing the scope of research in concurrent, parallel, and distributed computing. Participants were asked to submit an original paper that identified one or more design patterns based upon their experience building CPD systems.

2. Writers Workshop Format

The workshop was organized in the style of a writers workshop, popularized in the design patterns community by the Pattern Languages of Programming (PLoP) conference and in other venues. Unlike the typical conference workshop, which is like a mini-conference with authors presenting their papers to relatively passive listeners, the writers workshop requires active participation by the audience, and mostly passive participation by the author.

The writers workshop format was conducted as follows:

- A mediator introduces the author and the title of their paper.
- The author provides a short summary of the paper by reading a few paragraphs of his or her choosing. After this summary, the author is only allowed to make notes and cannot comment further during the review process.
- The mediator asks two or three people to provide a brief summary of the key ideas that the paper is attempting to communicate to the reader.
- Positive comments are then solicited about the contents of the paper, the organization, and the writing style of the paper.
- Constructive criticisms are then solicited on the content, organization, and style.
- Once all comments have been made, the author is reintroduced and allowed to ask for clarification of comments made on the papers by workshop participants. The author is then give an opportunity to make a few closing remarks about their paper.

A key feature of this format is that once the author goes silent, any comments that are made regarding the paper are addressed to the group as a whole, or the moderator, without direct reference to the author. The purpose of this format is to allow the participants to make reasonably open and unconstrained remarks about the author's paper. The opportunity exists for the participants to harshly critique a particular paper; however, the heat and rhetoric of the discussion is controlled by the mediator, who acts to focus the discussion on those aspects of the paper that will ultimately lead the author to realize where the paper might be weak. The purpose is to convey to the writer how others interpret what is written and to help the writer make substantive improvements so that the ideas in the paper are clearly communicated.

3. Workshop Content

Fourteen papers were accepted to the workshop. The papers were made available on a web server so that workshop participants could have access to all papers prior to the official workshop. The key to a successful writers workshop is that the participants read some, if not all, of the papers. The organizers anticipated that most participants would not be familiar with the writer's workshop format, so the workshop was organized into two separate sessions. The morning session consisted of a group review of three papers selected by the organizers as representative of the common styles of design pattern papers. In the afternoon section the participants split up into three parallel sessions, each headed by one of the workshop organizers.

In the morning session, each of the organizers took turns moderating the critique of a paper so that participants could learn how a writers workshop is conducted. As expected, the review of the first paper was dominated by a small number of people who had experience with the writers workshop format, and who had read the paper beforehand. By the review of the second paper, more participants started contributing constructive comments and criticisms (possibly having had time to skim the second paper while the first was being reviewed!). By the review of the third paper, most participants had caught on to the style of the workshop and were making substantive contributions. The groups and papers discussed during the workshop are outlined below.[1]

3.1. Group Discussion Papers

Composite Messages: A Structural Pattern for Communication between Components, by Aamod Sane and Roy Campbell, University of Illinois, Urbana-Champaign.

The Broker Architectural Framework, by Michael Stal, SIEMENS AG Corporate Research and Development.

Object Group: an Object Behavioral Pattern for Fault-Tolerant and Group Communication in Distributed Systems, by Silvano Maffeis, Department of Computer Science, Cornell University.

3.2. Group I (leader: Dennis Kafura)

Using Replication for Distribution: Patterns for Efficient Updating, by Charles Weir, Object Designers Ltd.

Warden: A Pattern for Object Distribution, by Fernando Das Neves and Alejandra Garrido, LIFIA, La Plata, Buenos Aires.

Recoverable Distributor: A Design Pattern for Fault-Tolerant Sharing in Distributed Computing Nayeem Islam and Murphy Devarokonda, IBM T. J. Watson Labs.

Managing Continuous Data Feed with Subscriber/ Publisher Pattern, by Raman Kannan, Monmouth University.

3.3. Group II (leader: Greg Lavender)

Design Patterns for Binding in Distributed Systems, by Steve Crane, Jeff Magee, Nat Pryce, Department of Computing, Imperial College, London.

The Pipeline Design Pattern, by Allan Vermeulen, Gabe Beged-Dov and Patrick Thompson, Rogue Wave Inc.

Identity Indirection, Chris Tarr, ObjectSpace Inc.

3.4. Group III (leader: Doug Schmidt)

Thread-Specific Storage: A Pattern for Reducing Locking Overhead in Concurrent Programs, by Tim Harrison and Douglas C. Schmidt, Washington University, St. Louis.

Local Serialization Pattern, by Antonio Rito Silva, Joao Pereira and Pedro Sousa, INESC/IST Technical University of Lisbon.

Buffered Collection and Buffered Iterator Patterns, by Phil Brooks, Mentor Graphics Corporation.

Private Thread: A Software Pattern for the Implementation of Autonomic Object Behavior, John Gilbert, Objective Software.

3.5. Final Discussion and Closing Remarks

One sign of a successful workshop is when the participants do not all rush for the door as the workshop is winding down. Almost all of the participants were on hand for the final group discussion and closing remarks. The organizer's asked the participants to turn the table around and apply the writer's workshop model to the way that the workshop was organized. They were asked to make constructive comments about the organization, structure and content of the day's proceedings. Two key observations are:

- Many participants remarked that they had not participated in a writers workshop before, and were really quite surprised at how much they had benefited and learned more about the author's papers being able to have a group discussion, even though they may not have read the paper in detail.

[1] All papers are available on the web using the URL `http://www.cs.wustl.edu/~schmidt/OOPSLA-95/`

- Some participants commented that they plan to adopt the writers workshop format at their organizations as the preferred way to review and discuss papers.
- Some authors commented that although it was perhaps initially difficult for them to listen to and accept criticisms of their paper and ideas, they came to understand that many of the comments were in fact constructive and would enhance the quality of their paper.

The following summarizes some observations and issues that arose throughout the group sessions and during the closing discussion:

- The forces that motivate an author to write a paper are often not the same as those that motivate someone to read the paper. For most authors, this is a not-so-obvious realization. The motivation section of a paper is often the most criticized as it either sets the correct framework and expectation for the rest of the paper, or leads to numerous misunderstandings about the ideas and intents of the paper.
- There are different opinions on the degree of real code versus pseudo-code that should be in a patterns paper. Some participants felt that real code was too detailed and detracted from the flow of the paper. Other participants felt that papers lacking real code were incomplete and vague. There was a fair amount of consensus that patterns should be written in a modular way, so that code was localized to the implementation section. This strategy allows readers with different levels of interest in the details to benefit from the paper without being unduly distracted or disappointed in the level of detail.
- Performance is a dominant issue in CPD systems. Many patterns do not seem to convey this adequately since they tend to focus on structural and behavioral aspects, rather than performance aspects. There was general agreement that "optimization patterns" are a fruitful direction for CPD patterns research.
- Many CPD patterns present high-level architectural relationships while trying to remain implementation neutral. It is difficult to totally avoid implementation bias in describing a general pattern applicable to CPD systems because implementation issues such as location, policies, mechanisms, and algorithms quickly manifest themselves. Architectural CPD patterns are perhaps best described in general terms, followed by a discussion of the related patterns supporting an implementation and the forces that lead one to make specific implementation choices, or perhaps more importantly, indicate which choices to avoid.
- A final observation is that there were many instances of patterns for concurrent and distributed systems, but no contributions from the parallel computing domain. Effort should be made to communicate with researchers and practitioners in the parallel computing domain to solicit their contributions to the area of CPD patterns.

4. Participants

Phil Brooks, Mentor Graphics Inc.
Murphy Devarokonda, IBM T. J. Watson Labs
Jim Doble, BNR
Gabe Beged-Dov, Rogue Wave Inc.
Il-Hyung Cho, Clemson University
Alejandra Garrido, University Nacional dela Plata
John Gilbert, Objective Software Inc.
Hermann Huni, GLUE Software Engineering
Nayeem Islam, IBM T. J. Watson Labs
Raman Kannan, Monmouth University
Doug Lea, SUNY Oswego
Donald Liib, Royal Institute of Technology, Sweden
Fernando Das Neves, University Nacional dela Plata
Ed Posnak, University of Texas at Austin
Nat Pryce, Imperial College
Aamod Sane, University of Illinois at Urbana
Kevin Shank, Rochester Institute of Technology
Antonio Rito Silva, INESC Technical University of Lisbon
Chris Tarr, ObjectSpace Inc.
Patrick Thompson, Rogue Wave Inc.
Anand Tripathi, National Science Foundation
Allan Vermeulen, Rogue Wave Inc.
Charles Weir, Object Designers Limited
Bob Wilhelm, Objective Control Inc.