

Reflective and Adaptive Systems

Fall 2004: CS 622/722

Time: 2:00pm - 3:15pm Tuesday and Thursday
Classroom: CH 396

Instructor: Jeff Gray, Ph.D.
Office: CH 126
Phone: 934-8643
Email: gray (at) cis.uab.edu
Web: <http://www.gray-area.org>

Course URL: <http://www.cis.uab.edu/cs622>

Office Hours: Tu/Thu 3:30pm-5:00pm, or by appointment

Course Goals

... program structure should be such as to anticipate its adaptations and modifications. Our program should not only reflect (by structure) our understanding of it, but it should also be clear from its structure what sort of adaptations can be catered for smoothly. [Dijkstra]

A longstanding goal in software development is to construct programs that are easily modified and extended. The desired result is to achieve modularization such that a change in a design decision is isolated to one location of a program. The proliferation of software in everyday life (e.g., embedded systems found in automobiles, mobile phones, and television sets) has increased the conformity and invisibility of software. As demands for such software increase, future requirements will necessitate new strategies for improved modularization in order to support the requisite adaptations.

Software's ability to adapt is actually partitioned among two stages: modifiability during development, and adaptation during execution. The first type of adaptation is concerned with design-time, or compile-time, techniques that permit the modification of the structure and function of a software representation in order to address changing stakeholder requirements. To support such evolution, techniques such as aspect-oriented programming, and object-oriented frameworks, are but a few of the ideas that have shown promise in assisting a developer in the isolation of points of variation and configurability.

The second type of adaptation occurs at run-time during the execution of the program. This type of adaptation refers to a system's ability to modify itself and to respond to changing conditions in its external environment. To accommodate such changes, research

in meta-programming and reflection have offered some recourse, especially in the area of adaptive middleware.

Reflection and metaprogramming are useful techniques that provide support for adaptive systems. Reflection permits a program to inquire about its own state at run-time (called introspection), and, in some cases, permits the modification of the semantics of the run-time system itself (called intercession).

This course will examine the principles of reflection and metaprogramming in several contexts. The general topics of the course will be:

- An introduction to the history and literature of reflection and metaprogramming
- A review of the types of reflection supported in conventional languages, and some of the implementation drawbacks (e.g., Java)
- Experimentation with several research prototypes that support novel ideas in reflection
- Applications of these techniques to distributed object computing (e.g., literature on reflective middleware)
- Issues and tool exposure to metamodeling and domain modeling using Model-Integrated Computing (MIC)
- Topics within the area of Aspect-Oriented Software Development (AOSD) will be studied in the course

Textbook

Although there is a book that will be published in this area in December, currently there is no formal textbook for this course. This syllabus contains a list of the required readings, as well as other related papers/books. You will be given a copy of each paper well in advance of the designated class to which it is assigned.

Grade Determination

Your final grade for this course will be determined by the following items:

Quizzes:	20%	Homework:	30%
Midterm Exam:	20%	Final Exam:	30%

There will be a total of 10 quizzes. You will be able to drop the lowest score (that is, only the top 9 scores will count toward your overall quiz grade). There will be NO makeup quizzes, unless you *give me over a weeks advanced notice* with a good reason (e.g., out of town for a conference). Requests for make-up quizzes, after the fact, will be denied.

There will be NO makeup exams.
Homework is due at the beginning of class.

Honor Code

Each student is to do his or her own work. This means that you are not to seek out the help of other students (or give help, if asked) in order to solve specific problems of your homework assignments. It also means that you should not sign up for mailing lists and ask for detailed help from others on the net. Of course, you may discuss generalities about an assignment with your fellow students. If you are unsure of what is permitted, in terms of discussing an assignment problem, please ask me for clarification.

Please turn off all cell phones and pagers while you are in lecture. If we are in the lab, do not surf the web or write email during the lecture.

Please arrive on time so that you do not cause a disruption in the middle of class.

Disabilities

If you have any disability that would put you at a disadvantage in performing an assignment, or in taking an exam, please meet with me privately to discuss ways in which I can assist you as you perform the required work in this course.

Tentative Schedule

The following is an outline of the schedule for this course. The schedule itself may need to be somewhat “adaptive” (i.e., some papers may be dropped from the list at a later time due to lack of time, or other papers may be added in the event new papers are published during the course of the semester).

Date	Topic	Readings	Remarks
August 19	Course Introduction		
	<u>Reflection Quarter</u>		
August 24	Computational and Object Reflection	[Maes, 1987]	
August 26	Java Reflection	[Liang and Bracha, 1998] Java Reflection API	Quiz 1
August 31	OpenC++	[Chiba, 1995]	
September 2	OpenJava	[Tatsubori et al., 2000]	Quiz 2
September 7	Bytecode Manipulation JMangler	[Kniesel et al., 2001]	
September 9	Javassist	[Chiba, 2000]	Quiz 3 HW 1 Assigned
September 14	Javassist Lab Exercise		
September 16	Class cancelled		Hurricane Ivan
September 21 <i>At EDOC In Monterey</i>	Early AOP		Gregor Kiczales AOP Video
September 23 <i>At EDOC In Monterey</i>	Industry examples of reflection usage		Guest Lecture: Arvind Chavar (SunGard) Quiz 4 (free)
September 28	JMangler Lab HW 1 help session		
September 30	Template Metaprogramming	[Veldhuizen, 1995]	Quiz 5 HW 1 Due

	<u>AOSD Quarter</u>		
October 5	Separation of Concerns AOSD	[Kiczales et al., 1997] [Filman/Friedman, 2000]	[Parnas, 1972]
October 7	Benefits of AOSD	[Lippert/Lopes, 2000] [Coady/Kiczales, 2003]	Quiz 6
October 12 <i>At OMG D.C.</i>	Open Implementation	[Kiczales, 1996]	Another Kiczales Video
October 14 <i>At OMG D.C.</i>	HyperJ/CME	[Tarr et al., 1999] [Ossher/Tarr, 2001]	<u>Carl Wu</u>
October 19	AspectJ Intro	[Kiczales et al., 2001]	[Walker et al., 1999] [Murphy et al., 2001]
October 21	AspectJ		Quiz 7 HW 2 Assigned
October 26 <i>At OOPSLA Vancouver</i>	Aspects and MPI	[Harbulot/Gurd, 2004]	<u>HPCL Members</u>
October 28 <i>At OOPSLA Vancouver</i>	<u>Mid-term Exam</u>		
November 2	AspectJ Lab Exercises		[Bergmans/Aksit, 01]
November 4	Refactoring and Aspects	[Hanenberg, 2003]	
	<u>MIC Quarter</u>		
November 9	MIC Motivation GME Intro	[Karsai, 1995] [Long et al., 1998] [Sztipanovits/Karsai, 97] [Sztipanovits et al., 1998]	HW 2 Due
November 11	MIC AODM	[Gray et al., 2001] [Batory, 2003]	Quiz 8 HW 3 Assigned
November 16	GME Metamodeling	[Lédeczi et al., 2001] [Nordstrom et al., 1999]	

November 18	GME Metamodeling (Lab)		
November 23	GME Interpreter Writing		Quiz 9
November 25	<u>Thanksgiving</u>		No class
November 30	GME Interpreter Writing (Lab)		
	<u>Adaptive Middleware Quarter</u>		
December 2	Aspect-Oriented Middleware	[Colyer, 2004]	HW 3 Due Quiz 10 [Zhang/Jacobsen, 03] Zen [Wang et al., 2000] <u>CACM special issue</u> [Schmidt et al., 2001] [Karr et al., 2001] [Loyall et al., 2002]
December 9-15	Final Exam		

Primary Course Readings

[Batory2003] Don Batory, Jacob Neal Sarvela, and Axel Rauschmeyer, "Scaling Step-Wise Refinement," *International Conference on Software Engineering*, Portland, Oregon, May 2003, pp. 187-197.

[Bergmans and Aksit, 2001] Lodewijk Bergmans and Mehmet Aksit, "Composing Crosscutting Concerns using Composition Filters," *Communications of the ACM*, October 2001, pp. 51-57.

[Chiba, 1995] Shigeru Chiba, "A Metaobject Protocol for C++," *Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, Austin, Texas, October 1995, pp. 285-299.

[Chiba, 2000] Shigeru Chiba, "Load-time Structural Reflection in Java," *European Conference on Object-Oriented Programming (ECOOP)*, LNCS 1850, Springer-Verlag, Cannes, France, June 2000, pp. 313-336.

[Clarke and Walker, 2001] Siobhán Clarke and Robert J. Walker, "Composition Patterns: An Approach to Designing Reusable Aspects," *International Conference on Software Engineering (ICSE)*, Toronto, Ontario, Canada, May 2001, pp. 5-14.

[Coady and Kiczales, 2003] Yvonne Coady and Gregor Kiczales, "Back to the Future: A Retroactive Study of Aspect Evolution in Operating System Code," *Proceedings of the International Conference on Aspect-Oriented Software Development*, Boston, MA, March 2003, pp. 50-59.

[Filman and Friedman, 2000] Robert Filman and Dan Friedman, "Aspect-Oriented Programming is Quantification and Obliviousness," *OOPSLA Workshop on Advanced Separation of Concerns*, Minneapolis, Minnesota, October 2000.

[Gray et al., 2001] Jeff Gray, Ted Bapty, Sandeep Neema, and James Tuck, "Handling Crosscutting Constraints in Domain-Specific Modeling," *Communications of the ACM*, October 2001, pp. 87-93.

[Karr et al., 2001] David Karr, Craig Rodrigues, Joseph Loyall, Richard Schantz, Yamuna Krishnamurthy, Irfan Pyarali, and Douglas Schmidt, "Application of the QuO Quality-of-Service Framework to a Distributed Video Application," *International Symposium on Distributed Objects and Applications*, Rome, Italy, September 2001.

[Karsai, 1995] Gábor Karsai, "A Configurable Visual Programming Environment: A Tool for Domain-Specific Programming," *IEEE Computer*, March 1995, pp. 36-44.

[Kiczales et al., 1993] Gregor Kiczales, J. Michael Ashley, Luis Rodriguez, Amin Vahdat, and Daniel G. Bobrow, "Metaobject Protocols: Why We Want Them and What Else Can They Do?" A. Paepcke, editor, *Object-Oriented Programming: The CLOS Perspective*, 1993, pp. 101-118.

[Kiczales, 1996] Gregor Kiczales, "Beyond the Black Box: Open Implementation," *IEEE Software*, January 1996, pp. 8-11.

[Kiczales et al., 1997] Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Videira Lopes, Jean-Marc Loingtier, and John Irwin, "Aspect-Oriented Programming," *European Conference on Object-Oriented Programming (ECOOP)*, LNCS 1241, Springer-Verlag, Jyväskylä, Finland, June 1997, pp. 220-242.

[Kiczales et al., 2001] Gregor Kiczales, Eric Hilsdale, Jim Hugunin, Mik Kersten, Jeffrey Palm, and William Griswold, "Getting Started with AspectJ," *Communications of the ACM*, October 2001, pp. 59-65.

[Kniesel et al., 2001] Günter Kniesel, Pascal Costanza, Michael Austermann, "JMangler - A Framework for Load-Time Transformation of Java Class Files," *IEEE Workshop on Source Code Analysis and Manipulation (SCAM)*, November 2001.

[Lafferty and Cahill, 2003] Donal Lafferty and Vinny Cahill, "Language-Independent Aspect-Oriented Programming," *Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, Anaheim, CA, October 2003.

[Lédeczi et al., 2001] Ákos Lédeczi, Arpad Bakay, Miklos Maroti, Peter Volgyesi, Greg Nordstrom, Jonathan Sprinkle, and Gábor Karsai, "Composing Domain-Specific Design Environments," *IEEE Computer*, November 2001, pp. 44-51.

[Liang and Bracha, 1998] Sheng Liang and Gilad Bracha, "Dynamic Class Loading in the Java Virtual Machine," *Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, Vancouver, B.C., Canada, October 1998, pp. 36-44.

[Lieberherr et al., 2001] Karl Lieberherr, Doug Orleans, and Johan Ovlinger, "Aspect-Oriented Programming with Adaptive Methods," *Communications of the ACM*, October 2001, pp. 39-41.

[Lippert and Lopes, 2000] Martin Lippert and Cristina V. Lopes, "A Study on Exception Detection and Handling Using Aspect-Oriented Programming," *International Conference on Software Engineering (ICSE)*, Limerick, Ireland, June 2000, pp. 418-427.

[Long et al., 1998] Earl Long, Amit Misra, and Janos Sztipanovits, "Increasing Productivity at Saturn," *IEEE Computer*, August 1998, pp. 35-43.

[Loyall et al., 2002] Joseph Loyall, Rick Schantz, Michael Atighetchi, and Partha Pal, "Packaging Quality of Service Control Behaviors for Reuse," *5th IEEE International Symposium on Object-Oriented Real-time Distributed Computing (ISORC)*, Washington, DC, April 29 - May 1, 2002.

[Maes, 1987] Pattie Maes, "Concepts and Experiments in Computational Reflection," *Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, Orlando, Florida, December 1987, pp. 147-155.

[Masuhara and Kiczales, 2003] Hidehiko Masuhara and Gregor Kiczales, "Modeling Crosscutting in Aspect-Oriented Mechanisms," *European Conference on Object-Oriented Programming (ECOOP)*, Darmstadt, Germany, July 2003.

[Nordstrom et al., 1999] Greg Nordstrom, Janos Sztipanovits, Gábor Karsai, and Ákos Lédeczi, "Metamodeling - Rapid Design and Evolution of Domain-Specific Modeling Environments," *International Conference on Engineering of Computer-Based Systems (ECBS)*, Nashville, Tennessee, April 1999, pp. 68-74.

[Ossher and Tarr, 2001] Harold Ossher and Peri Tarr, "Using Multidimensional Separation of Concerns to (Re)Shape Evolving Software," *Communications of the ACM*, October 2001, pp. 43-50.

[Popovici et al., 2003] Andrei Popovici, Gustavo Alonso, and Thomas Gross, "Just-in-time Aspects: Efficient Dynamic Weaving for Java," *Proceedings of the International Conference on Aspect-Oriented Software Development*, Boston, MA, March 2003, pp. 100-109.

[Pratap and Cytron, 2003] Ravi Pratap and Ron Cytron, "Transport Layer Abstraction in Event Channels for Embedded Systems," *Languages, Compilers, and Tools for Embedded Systems (LCTES 03)*, San Diego, CA, June 2003.

[Schmidt et al., 2001] Schmidt DC, Schantz RE, Masters MW, Cross JK, Sharp DC, DiPalma LP, "Toward Adaptive and Reflective Middleware for Network-Centric Combat Systems," *Crosstalk: The Journal of Defense Software Engineering*, November 2001, pp. 10-16.

[Sharp, 1998] David Sharp, "Reducing Avionics Software Cost Through Component Based Product-Line Development," *Software Technology Conference*, Salt Lake City, Utah, April 1998.

[Shonle et al., 2003] Macneil Shonle, Karl Lieberherr, Ankit Shah, "XAspects: An Extensible System for Domain-Specific Aspect Languages," *Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, Anaheim, CA, October 2003.

[Simonyi, 1996] Charles Simonyi, "Intentional Programming: Innovation in the Legacy Age," Presented at *IFIP WG 2.1*, June 1996.

[Smith, 1984] Brian Cantwell Smith, "Reflection and Semantics in Lisp," *Annual Symposium on Principles of Programming Languages*, Salt Lake City, Utah, 1984, pp. 23-35.

[Sztipanovits and Karsai, 1997] Janos Sztipanovits and Gábor Karsai, "Model-Integrated Computing," *IEEE Computer*, April 1997, pp. 10-12.

[Sztipanovits et al., 1998] Janos Sztipanovits, Gábor Karsai, and Ted Bapty, "Self-Adaptive Software for Signal Processing," *Communications of the ACM*, May 1998, pp. 66-73.

[Tarr et al., 1999] Peri Tarr, Harold Ossher, William Harrison, and Stanley Sutton, "N Degrees of Separation: Multi-Dimensional Separation of Concerns," *International Conference on Software Engineering (ICSE)*, Los Angeles, California, May 1999, pp. 107-119.

[Tatsubori et al., 2000] Tatsubori, M., S. Chiba, M.-O. Killijian, and K. Itano, "OpenJava: A Class-based Macro System for Java," in *Reflection and Software Engineering* (W. Cazzola, R. J. Stroud, and F. Tisato, eds.), LNCS 1826, Springer Verlag, 2000, pp. 117-133..

[Veldhuizen, 1995] Todd Veldhuizen, "Using C++ Template Metaprograms," *C++ Report* Vol. 7 No. 4 (May 1995), pp. 36-43.

[Wang et al., 2000] Nanbor Wang, Douglas C. Schmidt, Ossama Othman, and Kirthika Parameswaran, "Evaluating Meta-Programming Mechanisms for ORB Middleware," *IEEE Communication Magazine, special issue on Evolving Communications Software: Techniques and Technologies* (Bill Opdyke and Algirdas Pakstas, eds.), October 2000.

[Wohlstadter et al., 2003] Eric Wohlstadter, Stoney Jackson, Premkumar T. Devanbu, "DADO: Enhancing Middleware to Support Crosscutting Features in Distributed, Heterogeneous Systems," *International Conference on Software Engineering*, Portland, Oregon, pp. 174-186.

Other Books/Papers of Interest

[Aksit et al., 1992] Mehmet Aksit, Lodewijk Bergmans, and S. Vural, "An Object-Oriented Language-Database Integration Model: The Composition Filters Approach," *European Conference on Object-Oriented Programming (ECOOP)*, LNCS 615, Springer-Verlag, Utrecht, The Netherlands, June/July 1992, pp. 372-395.

[Aßmann, 2003] Uwe Aßmann, *Invasive Software Composition*, Springer-Verlag, 2003.

[Astley et al., 2001] Mark Astley, Daniel Sturman, and Gul Agha, "Customizable Middleware for Modular Distributed Software," *Communications of the ACM*, May 2001, pp. 99-107.

[Batory and Geraci, 1997] Don Batory and Bart J. Geraci, "Composition Validation and Subjectivity in GenVoca Generators," *IEEE Transactions on Software Engineering*, February 1997, pp. 67-82.

[Batory et al., 1998] Don Batory, Bernie Lofaso, and Yannis Smaragdakis, "JTS: Tools for Implementing Domain-Specific Languages," *Fifth International Conference on Software Reuse*, Victoria, Canada, June 1998, pp. 143-153.

[Bergmans and Aksit, 2001] Lodewijk Bergmans and Mehmet Aksit, "Composing Crosscutting Concerns using Composition Filters," *Communications of the ACM*, October 2001, pp. 51-57.

[Bobrow et al., 1993] Daniel G. Bobrow, Richard Gabriel, and Jon L. White, "CLOS in Context: The Shape of the Design Space," A. Paepcke, editor, *Object-Oriented Programming: The CLOS Perspective*, 1993, pp. 29-61.

[Booch et al., 1998] Grady Booch, Ivar Jacobson, James Rumbaugh, *The Unified Modeling Language User Guide*, Addison-Wesley, 1998.

[Booch, 2001] Grady Booch, "Through the Looking Glass," *Software Development Magazine*, July 2001, pp. 49-51.

[Cardone, 1999] Richard Cardone, "On the Relationship of Aspect-Oriented Programming and GenVoca," *Workshop on Institutionalizing Software Reuse*, Austin, Texas, January 1999.

[Chavez and de Lucena, 2001] Christina von Flach G. Chavez and Carlos J. P. de Lucena, "Design-level Support for Aspect-Oriented Software Development," *OOPSLA Workshop on Advanced Separation of Concerns*, Minneapolis, Minnesota, October 2001.

[Chiba and Masuda, 1993] Shigeru Chiba and Takashi Masuda, "Designing an Extensible Distributed Language with a Metalevel Architecture," *European Conference on Object-*

Oriented Programming (ECOOP), LNCS 707, Springer-Verlag, Kaiserslautern, Germany, July 1993, pp. 482-501.

[Clarke et al., 1999] Siobhán Clarke, William Harrison, Harold Ossher, and Peri Tarr, “Subject-Oriented Design: Towards Improved Alignment of Requirements, Design, and Code,” *Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, Denver, Colorado, November 1999, pp. 325-339.

[Clarke, 2002] Siobhán Clarke, “Extending Standard UML with Model Composition Semantics,” *Science of Computer Programming*, May 2002.

[Clavel, 2000] Manuel Clavel, *Reflection in Rewriting Logic: Metalogical Foundations and Metaprogramming Applications*, CSLI Publications, 2000.

[Coady et al., 2001a] Yvonne Coady, Gregor Kiczales, Mike Feeley, and Greg Smolyn, “Using AspectC to Improve the Modularity of Path-Specific Customization in Operating System Code,” *Proceedings of the Joint European Software Engineering Conference (ESEC) and 9th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE-9)*, Vienna, Austria, September 2001, pp. 78-88.

[Coady et al., 2001b] Yvonne Coady, Gregor Kiczales, Mike Feeley, Norm Hutchinson, and Joon Suan Ong, “Structuring Operating System Aspects,” *Communications of the ACM*, October 2001, pp. 79-82.

[Constantinides et al., 2000] Constantinos Constantinides, Atef Bader, Tzilla Elrad, P. Netinant, and Mohamed Fayad, “Designing an Aspect-Oriented Framework in an Object-Oriented Environment,” *ACM Computing Surveys*, March 2000.

[Craig, 2000] Iain Craig, *The Interpretation of Object-Oriented Programming Languages*, Springer-Verlag, 2000.

[Czarnecki and Eisenecker, 2000] Krzysztof Czarnecki and Ulrich Eisenecker, *Generative Programming: Methods, Tools, and Applications*, Addison-Wesley, 2000.

[de Moor et al., 1999] Oege de Moor, Simon Peyton-Jones, and Eric Van Wyk, “Aspect-Oriented Compilers,” *In First International Symposium on Generative and Component-Based Software Engineering*, Erfurt, Germany, September 1999, pp. 121-133.

[De Volder and D’Hondt, 1999] Kris De Volder and Theo D’Hondt, “Aspect-Oriented Logic Meta Programming,” *Proceedings of Reflection ’99: Metalevel Architectures and Reflection*, LNCS 1616, Springer-Verlag, Saint-Malo, France, July 1999, pp. 250-272.

[Elrad et al., 2001] Tzilla Elrad, Mehmet Aksit, Gregor Kiczales, Karl Lieberherr, and Harold Ossher, “Discussing Aspects of AOP,” *Communications of the ACM*, October 2001, pp. 33-38.

[Fayad et al., 1999] Mohamed Fayad, Douglas Schmidt, and Ralph Johnson, *Building Application Frameworks: Object-Oriented Foundations of Framework Design*, John Wiley and Sons, 1999.

[Filman, 2001] Robert Filman, "What is Aspect-Oriented Programming, Revisited," *ICSE Workshop on Advanced Separation of Concerns*, Toronto, Ontario, Canada, May 2001.

[Filman et al., 2002] Robert Fillman, Stuart Barrett, Diana Lee, and Ted Linden, "Inserting Ilities by Controlling Communications," *Communications of the ACM*, January 2002, pp. 116-122.

[Forman and Danforth, 1999] Ira R. Forman and Scott H. Danforth, *Putting Metaclasses to Work*, Addison-Wesley, 1999.

[Griswold et al., 2001] William G. Griswold, Jimmy J. Yuan, and Yoshikiyo Kato, "Exploiting the Map Metaphor in a Tool for Software Evolution," *International Conference on Software Engineering (ICSE)*, Toronto, Ontario, Canada, May 2001, pp. 265-274.

[Grosso, 2002] William Grosso, "Aspect-Oriented Programming and AspectJ," *Dr. Dobb's Journal*, August 2002.

[Grundy, 2000] John Grundy, "Multi-Perspective Specification, Design and Implementation of Software Components Using Aspects," *International Journal of Software and Knowledge Engineering*, December 2000, pp. 713-734.

[Hannemann and Kiczales, 2001] Jan Hannemann and Gregor Kiczales, "Overcoming the Prevalent Decomposition in Legacy Code," *ICSE Workshop on Advanced Separation of Concerns*, Toronto, Ontario, Canada, May 2001.

[Harrison and Ossher, 1990] William Harrison and Harold Ossher, "Subdivided Procedures: A Language Extension Supporting Extensible Programming," *International Conference on Computer Languages*, New Orleans, Louisiana, March 1990, pp. 190-197.

[Harrison et al., 1997] Timothy Harrison, David Levine, and Douglas C. Schmidt, "The Design and Performance of a Real-Time CORBA Event Service," *Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, Atlanta, Georgia, October 1997, pp. 184-200.

[Hofstadter, 1979] Douglas R. Hofstadter, *Gödel, Escher, Bach*, Random House, 1979.

[Hunleth et al., 2001] Frank Hunleth, Ron Cytron, and Chris Gill, "Building Customized Middleware Using Aspect-Oriented Programming," *OOPSLA Workshop on Advanced Separation of Concerns*, Tampa, Florida, October 2001.

[Johnson, 1997] Ralph E. Johnson, "Frameworks = (Components + Patterns)," *Communications of the ACM*, October 1997, pp. 39-42.

[Karsai and Gray, 2000] Gábor Karsai and Jeff Gray, "Component Generation Technology for Semantic Tool Integration," *IEEE Aerospace Conference*, Big Sky, Montana, March 2000.

[Katz and Gil, 1999] Shmuel Katz and Joseph Gil, "Aspects and Superimpositions," *ECOOP Workshop on Aspect-Oriented Programming*, Lisbon, Portugal, June 1999.

[Kersten and Murphy, 1999] Mik Kersten and Gail C. Murphy, "Atlas: A Case Study in Building a Web-based Learning Environment Using Aspect-Oriented Programming," *Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, Denver, Colorado, November 1999, pp. 340-352.

[Kiczales et al., 1991] Gregor Kiczales, Jim des Rivières, and Daniel G. Bobrow, *The Art of the Metaobject Protocol*, MIT Press, 1991.

[Kiczales, 1992] Gregor Kiczales, "Towards a New Model of Abstraction in the Engineering of Software," *Proceedings of the International Workshop on New Models for Software Architectures (IMSA): Reflection and Metalevel Architecture*, Tokyo, Japan, November 1992, pp. 1-11.

[Kiczales et al., 1992] Gregor Kiczales, John Lamping, Luis H. Rodriguez Jr., and Erik Ruf, "Macros that Reach Out and Touch Somewhere," Internal Technical Report, Embedded Computation Area, Xerox PARC, 1992.

[Kiczales et al., 2001] Gregor Kiczales, Eric Hilsdale, Jim Hugunin, Mik Kersten, Jeffrey Palm, and William Griswold, "An Overview of AspectJ," *European Conference on Object-Oriented Programming (ECOOP)*, LNCS 2072, Springer-Verlag, Budapest, Hungary, June 2001, pp. 327-353.

[Kon et al., 2000] Fabio Kon, Manuel Román, Ping Liu, Jina Mao, Tomonori Yamane, Luiz Claudio Magalhães, and Roy H. Campbell, "Monitoring, Security, and Dynamic Configuration with the dynamicTAO Reflective ORB," *IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing (Middleware'2000)*. New York. April 3-7, 2000.

[Laddad, 2002] Ramnivas Laddad, "I Want My AOP! Parts1-3," *Java World*, January through April, 2002, (<http://www.javaworld.com/javaworld/jw-01-2002/jw-0118-aspect.html>)

[Lee and Zachary, 1995] Arthur H. Lee and Joseph L. Zachary, "Reflections on Metaprogramming," *IEEE Transactions on Software Engineering*, November 1995, pp. 883-893.

[Lesiecki, 2002] Nicholas Lesiecki, "Test Flexibly with AspectJ and Mock Objects," IBM DeveloperWorks, (<http://www-106.ibm.com/developerworks/java/library/j-aspectj2/?open&l=007,t=gr>)

[Lieberherr, 1996] Karl Lieberherr, *Adaptive Object-Oriented Software*, International Thomson Publishing, 1996.

[Lieberherr and Holland, 1989] Karl Lieberherr and Ian Holland, "Assuring Good Style for Object-Oriented Programs," *IEEE Software*, September 1989, pp. 38-48.

[Lopes, 1997] Cristina Lopes, *D: A Language Framework for Distributed Programming*, Ph.D. Dissertation, College of Computer Science, Northeastern University, November 1997.

[Milicev, 2002] Dragan Milicev, "Automatic Model Transformation Using Extended UML Object Diagrams in Modeling Environments," *IEEE Transactions on Software Engineering*, April 2002, pp. 413-430.

[Milicev, 2002] Dragan Milicev, "Domain Mapping Using Extended UML Object Diagrams," *IEEE Software*, March 2002, pp. 90-97.

[Murphy et al., 1999] Gail C. Murphy, Robert J. Walker, and Elisa L.A. Baniassad, "Evaluating Emerging Software Development Technologies: Lessons Learned from Assessing Aspect-Oriented Programming," *IEEE Transactions on Software Engineering*, July/August 1999, pp. 438-455.

[Murphy et al., 2001] Gail C. Murphy, Albert Lai, Robert J. Walker, and Martin P. Robillard, "Separating Features in Source Code: An Exploratory Study," *International Conference on Software Engineering (ICSE)*, Toronto, Ontario, Canada, May 2001, pp. 275-284.

[Nelson et al., 2001] Torsten Nelson, Donald Cowan, and Paulo Alencar, "Supporting Formal Verification of Crosscutting Concerns," *Reflection 2001: The Third International Conference on Metalevel Architectures and Separation of Crosscutting Concerns*, LNCS 2192, Springer-Verlag, Kyoto, Japan, September 2001, pp. 153-169.

[Nordberg, 2001] Martin Nordberg, "Aspect-Oriented Dependency Inversion," *OOPSLA Workshop on Advanced Separation of Concerns*, Tampa, Florida, October 2001.

[Ossher et al., 1996] Harold Ossher, Matthew Kaplan, A. Katz, William Harrison, and Vincent Kruskal, "Specifying Subject-Oriented Composition," *Theory and Practice of Object Systems*, vol. 2(3), 1996, pp. 179-202.

[Ovlinger and Wand, 1999] Johan Ovlinger and Mitchell Wand, "A Language for Specifying Recursive Traversals of Object Structures," *Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, Denver, Colorado, November 1999, pp. 70-81.

[Parnas, 1972] David Parnas, "On the Criteria To Be Used in Decomposing Systems into Modules," *Communications of the ACM*, December 1972, pp. 1053-1058.

[Pohjonen and Kelly, 2002] Risto Pohjonen and Steve Kelly, "Domain-Specific Modeling," *Dr. Dobb's Journal*, August 2002.

[Rashid, 2001] Awais Rashid, "A Hybrid Approach to Separation of Concerns: The Story of SADES," *Reflection 2001: The Third International Conference on Metalevel Architectures and Separation of Crosscutting Concerns*, LNCS 2192, Springer-Verlag, Kyoto, Japan, September 2001, pp. 231-249.

[Robillard and Murphy, 2002] Martin Robillard and Gail Murphy, "Concern Graphs: Finding and Describing Concerns Using Structural Program Dependencies," *International Conference on Software Engineering (ICSE)*, Buenos Aires, Argentina, May 2002.

[Schonger et al., 2002] Stefan Schonger, Elke Pulvermueller, and Stefan Sarstedt, "Aspect-Oriented Programming and Component Weaving: Using XML Representations of Abstract Syntax Trees," *Second German Workshop on Aspect-Oriented Software Development*, Bonn, Germany, February 2002.

[Shukla et al., 2002] Dharma Shukla, Simon Fell, and Chris Sells, "Aspect-Oriented Programming Enables Better Code Encapsulation and Reuse," *MSDN Magazine*, March 2002, pp. 60-68.

[Sobel and Friedman, 1996] Jonathan M. Sobel and Daniel P. Friedman, "An Introduction to Reflection-Oriented Programming," *Reflection '96*, San Francisco, California, April 1996.

[Sullivan, 2001] Gregory T. Sullivan, "Aspect-Oriented Programming using Reflection and Metaobject Protocols," *Communications of the ACM*, October 2001, pp. 95-97.

[Tristram, 2001] Claire Tristram, "The Technology Review Ten: Untangling Code," *MIT Technology Review*, January 2001.

[Viega and Voas, 2000] John Viega and Jeffrey Voas, "Can Aspect-Oriented Programming Lead to More Reliable Software?" *IEEE Software*, November/December 2000, pp. 19-21.

[Walker et al., 1999] Robert J. Walker, Elisa L.A. Baniassad, and Gail C Murphy, "An Initial Assessment of Aspect-Oriented Programming," *International Conference on Software Engineering (ICSE)*, Los Angeles, California, May 1999, pp. 120-130.

[Zhang and Jacobsen, 2003] Charles Zhang, Hans-Arno. Jacobsen, “Quantifying Aspects in Middleware Platforms,” *Proceedings of the International Conference on Aspect-Oriented Software Development*, Boston, MA, March 2003, pp. 130-139.

Other Related Resources

The following is just an initial sample of related resources. Please check the course web site frequently for updates to this list.

Web Sites

All things Aspect-Oriented

Note: Our library has several books on AspectJ.

<http://aosd.net>

<http://aspectj.org>

<http://www.research.ibm.com/hyperspace/HyperJ/HyperJ.htm>

<http://www.ccs.neu.edu/research/demeter/DJ/>

Reflection

<http://java.sun.com/docs/books/tutorial/reflect/>

<http://cliki.tunes.org/Methods%20of%20Reflection>

<http://www.csg.is.titech.ac.jp/~chiba/openc++.html>

Model-Integrated Computing

<http://www.isis.vanderbilt.edu>

Adaptive Reflective Middleware

<http://quo.bbn.com/>

<http://www.comp.lancs.ac.uk/computing/research/mpg/reflection/>

<http://dsonline.computer.org/middleware/index.htm>

<http://choices.cs.uiuc.edu/2k/dynamicTAO>

Java Virtual Machine/Bytecode Information

<http://java.sun.com/docs/books/vmspec/2nd-edition/html/VMSpecTOC.doc.html>

<http://jakarta.apache.org/bcel/>

Related Conferences

<http://www.cis.uab.edu/info/HICSS-AESS/>

<http://www.openjit.org/reflection2001/>

<http://www.csg.is.titech.ac.jp/~chiba/reflection.html>

<http://www.comp.lancs.ac.uk/computing/rm2000/>

<http://www.cs.wustl.edu/~corsaro/RM2003/index.html>

<http://www.csg.is.titech.ac.jp/~chiba/oopsla98ws.html>