

University of Groningen

## Architectural design decisions

Jansen, Antonius Gradus Johannes

**IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.**

*Document Version*

Publisher's PDF, also known as Version of record

*Publication date:*

2008

[Link to publication in University of Groningen/UMCG research database](#)

*Citation for published version (APA):*

Jansen, A. G. J. (2008). *Architectural design decisions*. s.n.

**Copyright**

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

**Take-down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

*Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.*

---

## REFERENCES

---

- [1] G. Agha. *Actors: a model of concurrent computation in distributed systems*. MIT Press, 1986.
- [2] J. Aldrich, C. Chambers, and D. Notkin. Archjava: connecting software architecture to implementation. In *Proceedings of the 24th international conference on Software engineering*, pages 187–197. ACM Press, 2002.
- [3] J. Aldrich, V. Sazawal, C. Chambers, and D. Notkin. Language support for connector abstractions. In *ECOOP 2003 – Object-Oriented Programming: 17th European Conference*, volume 2743 of *Lecture Notes in Computer Science*, pages 74–102. Springer-Verlag, July 2003.
- [4] G. Arango, L. Bruneau, J. F. Cloarec, and A. Feroldi. A tool shell for tracking design decisions. *IEEE Software*, 8(2):75–83, March 1991.
- [5] The Archium website, <http://www.archium.net>.
- [6] M. Babar, I. Gorton, and B. Kitchenham. A framework for supporting architecture knowledge and rationale management. In A. H. Dutoit, R. McCall, I. Mistrík, and B. Paech, editors, *Rationale Management in Software Engineering*, chapter 11, pages 237–254. Springer-Verlag, March 2006.
- [7] M. A. Babar, R. C. de Boer, T. Dingsøyr, and R. Farenhorstir. Architectural knowledge management strategies: approaches in research and industry. In *Proceedings of the 2nd Workshop on SHARing and Reusing architectural Knowledge - Architecture, rationale, and Design Intent (SHARK/ADI 2007)*, May 2007.
- [8] M. Bachler, S. Buckingham Shum, D. D. Roure, D. Michaelides, and K. Page. Ontological mediation of meeting structure: Argumentation, annotation, and navigation. In *1st International Workshop on Hypermedia and the Semantic Web*, 2003.
- [9] E. L. A. Baniassad, G. C. Murphy, and C. Schwanninger. Design pattern rationale graphs: Linking design to source. In *Proceedings of the 25th ICSE*, pages 352–362, May 2003.
- [10] L. Bass, P. Clements, and R. Kazman. *Software architecture in practice*. Addison Wesley, 1998.
- [11] L. Bass, P. Clements, and R. Kazman. *Software architecture in practice 2nd ed*. Addison Wesley, 2003.
- [12] L. Bass, P. Clements, R. L. Nord, and J. Stafford. Capturing and using rationale for a software architecture. In A. H. Dutoit, R. McCall, I. Mistrík, and B. Paech,

- editors, *Rationale Management in Software Engineering*, chapter 12, pages 255–272. Springer-Verlag, March 2006.
- [13] D. Batory, J. Liu, and J. N. Sarvela. Refinements and multi-dimensional separation of concerns. In *Proceedings of the 9th European software engineering conference*, pages 48–57. ACM Press, 2003.
- [14] K. H. Bennett and V. T. Rajlich. Software maintenance and evolution: a roadmap. In *Proceedings of the conference on The future of Software engineering*, pages 73–87. ACM Press, 2000.
- [15] B. W. Boehm, E. Horowitz, R. Madachy, D. Reifer, B. K. Clark, B. Steece, A. W. Brown, S. Chulani, and C. Abts. *Software Cost Estimation with Cocomo II*. Prentice Hall, January 2000.
- [16] C. Boekhoudt. The big bang theory of ides. *Queue*, 1(7):74–82, 2003.
- [17] G. Booch, J. RumBaugh, and I. Jacobson. *The unified modeling language user guide*. Addison Wesley, 1998.
- [18] J. Bosch. Superimposition: A component adaptation technique. *Information and Software Technology*, 41(5):257–273, 25 March 1999.
- [19] J. Bosch. *Design & Use of Software Architectures, Adopting and evolving a product-line approach*. ACM Press/Addison Wesley, 2000.
- [20] J. Bosch. Maturity and evolution in software product lines: approaches, artefacts and organization. In *Proceedings of the 2nd Software Product Line Conference (SPLC 2002)*, August 2002.
- [21] J. Bosch. Software architecture: The next step. In *Software Architecture, First European Workshop (EWSA)*, volume 3047 of *LNCS*, pages 194–199. Springer, May 2004.
- [22] L. Bratthall, E. Johansson, and B. Regnell. Is a design rationale vital when predicting change impact? a controlled experiment on software architecture evolution. In *Second International Conference on Product Focused Software Process Improvement (PROFES)*, volume 1840 of *LNCS*, pages 126–139. Springer, 2000.
- [23] M. Broy. Automotive software and systems engineering. *memocode*, 0:143–149, 2005.
- [24] J. E. Burge and D. C. Brown. An integrated approach for software design checking using design rationale. In *1st International Conference on Design Computing and Cognition (DCC '04)*, pages 557–576, July 2004.
- [25] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal. *A system of patterns*. John Wiley & Sons, Inc., 1996.
- [26] R. Capilla, F. Nava, S. Pérez, and J. C. Dueñas. A web-based tool for managing architectural design decisions. *SIGSOFT Software Engineering Notes*, 31(5), 2006.
- [27] N. Chapin, J. E. Hale, K. M. Khan, J. F. Ramil, and Wui-Gee. Types of software evolution and software maintenance. *Journal of Software Maintenance and Evolution: Research and Practice*, 13(1):3–30, 2001.
- [28] A. Cimitile, A. De Lucia, G. A. Di Lucca, and A. R. Fasolino. Identifying objects in legacy systems using design metrics. *Journal of Systems and Software*, 44(3):199–211, 1999.

- [29] P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, and J. Stafford. *Documenting Software Architectures, Views and Beyond*. Addison Wesley, 2002.
- [30] E. J. Conklin and K. B. Yakemovic. A process-oriented approach to design rationale. *Human-Computer Interaction*, 6(3/4), 1991.
- [31] J. Conklin and M. L. Begeman. gibis: a hypertext tool for exploratory policy discussion. *ACM Transactions on Information Systems (TOIS)*, 6(4):303–331, 1988.
- [32] J. Conklin and M. L. Begeman. gibis: a tool for all reasons. *Journal of the American Society for Information Science*, 40(3):200–213, 1989.
- [33] M. E. Conway. How do committees invent? *Datamation*, 14(4):28–31, April 1968.
- [34] J. O. Coplien and N. B. Harrison. *Organizational Patterns of Agile Software Development*. Pearson Prentice Hall, 1995.
- [35] K. Czarnecki. Overview of generative software development. In *Proceeding of the Unconventional Programming Paradigms, International Workshop (UPP 2004)*, volume 3566 of *Lecture Notes in Computer Science*, pages 326–341. Springer, September 2004.
- [36] K. Czarnecki and U. Eisenecker. *Generative Programming: Methods, Tools, and Applications*. Addison-Wesley, June 2000.
- [37] E. M. Dashofy, A. van der Hoek, and R. N. Taylor. An infrastructure for the rapid development of xml-based architecture description languages. In *Proceedings of the 24th international conference on Software engineering*, pages 266–276. ACM Press, 2002.
- [38] R. C. de Boer and R. Farenhorst. In search of ‘architectural knowledge’. In *SHARK ’08: Proceedings of the 3rd international workshop on Sharing and reusing architectural knowledge*, pages 71–78, New York, NY, USA, 2008. ACM.
- [39] R. C. de Boer, R. Farenhorst, P. Lago, H. van Vliet, and A. G. J. Jansen. Architectural knowledge: Getting to the core. In *Proceedings of the Third International Conference on the Quality of Software Architectures (QoSA 2007)*, volume 4880 of *LNCIS*, pages 197–214, July 2007.
- [40] V. Dhar and M. Jarke. Dependency directed reasoning and learning in systems maintenance support. *IEEE Transactions on Software Engineering*, 14(2):211–227, 1988.
- [41] F. P. Dusan Bålek. Software connectors and their role in component deployment. In K. Zielinski, K. Geihs, and A. Laurentowski, editors, *Third International Working Conference on Distributed Applications and Interoperable Systems (DAIS)*, volume 198 of *IFIP Conference Proceedings*. Kluwer, 2001.
- [42] A. H. Dutoit, R. McCall, I. Mistrik, and B. Paech, editors. *Rationale Management in Software Engineering*. Springer-Verlag, March 2006.
- [43] T. Eisenbarth, R. Koschke, and D. Simon. Aiding program comprehension by static and dynamic feature analysis. In *Proceedings of the International Conference on Software Maintenance (ICSM’01)*, pages 602–611. IEEE Computer Society, November 2001.
- [44] D. Falessi, G. Cantone, and M. Becker. Documenting design decision rationale to improve individual and team design decision making: an experimental evaluation. In

- Proceedings of the 2006 ACM/IEEE international symposium on International symposium on empirical software engineering (ISESE '06)*, pages 134–143, New York, NY, USA, 2006. ACM Press.
- [45] R. Farenhorst, R. C. de Boer, R. Deckers, P. Lago, and H. van Vliet. What's in constructing a domain model for architectural knowledge? In *Proceedings of the 18th International Conference on Software Engineering and Knowledge Engineering (SEKE2006)*, July 2006.
- [46] R. Farenhorst, P. Lago, and H. van Vliet. Effective tool support for architectural knowledge sharing. In *Proceedings of the First European Conference on Software Architecture (ECSA 2007)*, volume 4758 of *LNCS*, pages 123–138, September 2007.
- [47] L. Feijs, R. Krikhaar, and R. van Ommering. A relational approach to support software architecture analysis. *Software - Practice and Experience*, 28(4):371–400, April 1998.
- [48] M. Fowler. Dealing with roles. In *Proceedings of the 4th Annual Conference on the Pattern Languages of Programs (PLoP)*, September 2-5 1997.
- [49] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns - Elements of Reusable Object-Oriented Software*. Addison Wesley, 1994.
- [50] D. Garlan, R. T. Monroe, and D. Wile. Acme: Architectural description of component-based systems. In G. T. Leavens and M. Sitaraman, editors, *Foundations of Component-Based Systems*, pages 47–68. Cambridge University Press, 2000.
- [51] C. Ghezzi, M. Jazayeri, and D. Mandrioli. *Fundamentals of software engineering*. Prentice-Hall, Inc., 1991.
- [52] P. Gibson. Feature requirements models: Understanding interactions. In L. L. P. Dini, R. Boutaba, editor, *Feature Interactions in Telecommunications Networks IV*, pages 46–60. IOS Press, June 1997.
- [53] R. L. Glass, I. Vessey, and V. Ramesh. Research in software engineering: an analysis of the literature. *Information & Software Technology*, 44(8):491–506, 2002.
- [54] R. Godin, G. Mineau, R. Missaoui, M. St-Germain, and N. Faraj. Applying concept formation to software reuse. *International Journal of Software Engineering and Knowledge Engineering*, 5(1):119–142, 1995.
- [55] D. G. Gregg, U. R. Kulkarni, and A. S. Vinzé. Understanding the philosophical underpinnings of software engineering research in information systems. *Information Systems Frontiers*, 3(2):169–183, 2001.
- [56] Griffin project website, <http://griffin.cs.vu.nl>.
- [57] M. L. Griss. Implementing product-line features by composing aspects. In *Proceedings of the first conference on Software product lines : experience and research directions*, pages 271–288, Norwell, MA, USA, 2000. Kluwer Academic Publishers.
- [58] M. L. Griss. Implementing product-line features with component reuse. In W. B. Frakes, editor, *Software Reuse: Advances in Software Reusability, 6th International Conference, ICSR-6*, volume 1844 of *Lecture Notes in Computer Science*. Springer, June 2000.
- [59] M. L. Griss, J. Favaro, and M. d' Alessandro. Integrating feature modeling with the rseb. In *ICSR '98: Proceedings of the 5th International Conference on Software*

- Reuse*, page 76, Washington, DC, USA, 1998. IEEE Computer Society.
- [60] Y.-G. Gueheneuc. A systematic study of uml class diagram constituents for their abstract and precise recovery. In *Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC'04)*, pages 265–274. IEEE Computer Society, 2004.
- [61] J. V. Gurp, J. Bosch, and M. Svahnberg. On the notion of variability in software product lines. In *WICSA '01: Proceedings of the Working IEEE/IFIP Conference on Software Architecture (WICSA'01)*, page 45, Washington, DC, USA, 2001. IEEE Computer Society.
- [62] I. Habli and T. Kelly. Capturing and replaying architectural knowledge through derivational analogy. In *SHARK-ADI '07: Proceedings of the Second Workshop on SHaring and Reusing architectural Knowledge Architecture, Rationale, and Design Intent*, page 4, Washington, DC, USA, 2007. IEEE Computer Society.
- [63] M. T. Hansen, N. Nohria, and T. Tierney. What's your strategy for managing knowledge? *Havard Business Review*, 77(2):106–116, March–April 1999.
- [64] N. B. Harrison, P. Avgeriou, and U. Zdun. Architecture patterns as mechanisms for capturing architectural decisions. *IEEE Software*, 24(4):38–45, 2007.
- [65] W. Harrison and H. Ossher. Subject-oriented programming: a critique of pure objects. In *Proceedings of the eighth annual conference on Object-oriented programming systems, languages, and applications*, pages 411–428. ACM Press, 1993.
- [66] C. Hofmeister, P. Kruchten, R. L. Nord, H. Obbink, A. Ran, and P. America. Generalizing a model of software architecture design from five industrial approaches. In *Proceedings of the 5th IEEE/IFIP Working Conference on Software Architecture (WICSA 2005)*, pages 77–88. IEEE Computer Society, 2005.
- [67] C. Hofmeister, R. Nord, and D. Soni. *Applied software architecture*. Addison Wesley, 2000.
- [68] C. Hofmeister, R. L. Nord, and D. Soni. Global analysis: moving from software requirements specification to structural views of the software architecture. *IEE Proceedings Software*, (4):187–197, August 2005.
- [69] H. J. Holz, A. Applin, B. Haberman, D. Joyce, H. Purchase, and C. Reed. Research methods in computing: what are they, and how should we teach them? In *ITiCSE-WGR '06: Working group reports on ITiCSE on Innovation and technology in computer science education*, pages 96–114, New York, NY, USA, 2006. ACM Press.
- [70] IEEE/ANSI. *Recommended Practice for Architectural Description of Software-Intensive Systems*, 2000. IEEE Standard No. 1471-2000, Product No. SH94869-TBR.
- [71] V. Jakobac, A. Egyed, and N. Medvidovic. Improving system understanding via interactive, tailorable, source code analysis. In M. Cerioli, editor, *FASE*, volume 3442 of *Lecture Notes in Computer Science*, pages 253–268. Springer, 2005.
- [72] C. B. Jaktman, J. Leaney, and M. Liu. Structural analysis of the software architecture - a maintenance assessment case study. In P. Donohoe, editor, *Software Architecture (WICSA1)*, volume 140 of *IFIP Conference Proceedings*, pages 455–470. Kluwer, Februari 1999.
- [73] A. Jansen. Feature based composition. Master's thesis, University of Groningen, September 2002.

- [74] A. Jansen, R. Smedinga, J. van Gorp, and J. Bosch. First class feature abstractions for product derivation. *IEE Proceedings Software*, 151(4):187–197, August 2004.
- [75] A. G. J. Jansen. Athena, a large scale programming lab support tool. In *Proceedings of the Dutch National Computer Science Education Congress (NIOC)*, pages 83–89. Uitgeverij Passage, 2004.
- [76] A. G. J. Jansen and J. Bosch. Evaluation of tool support for architectural evolution. In *Proceedings of the 19th IEEE International Conference on Automated Software Engineering (ASE 2004)*, pages 375–378. IEEE, September 2004.
- [77] A. G. J. Jansen and J. Bosch. Software architecture as a set of architectural design decisions. In *Proceedings of the 5th IEEE/IFIP Working Conference on Software Architecture (WICSA 2005)*, pages 109–119, November 2005.
- [78] A. G. J. Jansen, J. Bosch, and P. Avergiou. Documenting after the fact: recovering architectural design decisions. *Journal of Systems and Software*, 81(4):536–557, April 2008.
- [79] A. G. J. Jansen, J. van der Ven, P. Avgeriou, and D. K. Hammer. Tool support for architectural decisions. In *Proceedings of the 6th IEEE/IFIP Working Conference on Software Architecture (WICSA 2007)*, page 4, Januari 2007.
- [80] A. G. J. Jansen, J. van Gorp, and J. Bosch. Reconstructing architectural design decisions: A case study. Technical Report IWI preprint 2003-7-02, Department of Mathematics and Computing Science, University of Groningen, PO Box 800, 9700 AV The Netherlands, December 2003.
- [81] JavaCC website, <http://javacc.dev.java.net/>.
- [82] JGraph website, <http://www.jgraph.org>.
- [83] Chris johnson’s website on research in computing science. [http://www.dcs.gla.ac.uk/~johnson/teaching/research\\_skills/research.html](http://www.dcs.gla.ac.uk/~johnson/teaching/research_skills/research.html).
- [84] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson. Feature-oriented domain analysis (foda) feasibility study. Technical Report CMU/SEI-90-TR-21, ADA 235785, Software Engineering Institute, Carnegie Mellon University, 1990.
- [85] K. C. Kang, S. Kim, J. Lee, K. Kim, E. Shin, and M. Huh. Form: A feature-oriented reuse method with domain-specific reference architectures. *Ann. Softw. Eng.*, 5:143–168, 1998.
- [86] R. K. Keller, R. Schauer, S. Robitaille, and P. Page. Pattern-based reverse-engineering of design components. In *Proceedings of the 21st International Conference on Software Engineering (ICSE 1999)*, pages 226–235. IEEE Computer Society, May 1999.
- [87] G. Kiczales, E. Hilsdale, J. Hugunin, M. Kersten, J. Palm, and W. G. Griswold. An overview of AspectJ. *Lecture Notes in Computer Science*, 2072:327–355, 2001.
- [88] G. Kiczales, J. Lamping, A. Menhdhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin. Aspect-oriented programming. In M. Akşit and S. Matsuoka, editors, *Proceedings ECOOP*, volume 1241, pages 220–242. Springer-Verlag, 1997.
- [89] M. Klein. Capturing design rationale in concurrent engineering teams. *IEEE Computer*, 26(1):39–47, 1993.

- [90] A. G. Kleppe, J. B. Warmer, and W. Bast. *MDA Explained: The Model Driven Architecture : Practice and Promise*. Addison-Wesley, 2003.
- [91] R. L. Krikhaar, A. Postma, A. Sellink, M. Stroucken, and C. Verhoef. A two-phase process for software architecture improvement. In *International Conference on Software Maintenance (ICSM99)*, pages 371–380, september 1999.
- [92] P. Kruchten. The 4+1 view model of architecture. *IEEE Software*, 12(6):42–50, November 1995.
- [93] P. Kruchten. An ontology of architectural design decisions in software intensive systems. In *2nd Groningen Workshop on Software Variability*, pages 54–61, December 2004.
- [94] P. Kruchten. Casting software design in the function-behavior-structure framework. *IEEE Softw.*, 22(2):52–58, 2005.
- [95] P. Kruchten, P. Lago, and H. van Vliet. Building up and reasoning about architectural knowledge. In *Proceedings of the Second International Conference on the Quality of Software Architectures (QoSA 2006)*, June 2006.
- [96] P. Kruchten, P. Lago, H. van Vliet, and T. Wolf. Building up and exploiting architectural knowledge. In *WICSA 5*, November 2005.
- [97] W. Kunz and H. W. J. Rittel. Issues as elements of information systems. Technical Report Working paper 131, Institut fur Grundlagen der Planung, Universitat Stuttgart, July 1970.
- [98] P. Lago and P. Avgeriou. First workshop on sharing and reusing architectural knowledge. *SIGSOFT Software Engineering Notes*, 31(5):32–36, 2006.
- [99] P. Lago and H. van Vliet. Explicit assumptions enrich architectural models. In *ICSE '05: Proceedings of the 27th international conference on Software engineering*, pages 206–214, New York, NY, USA, 2005. ACM Press.
- [100] A. Lakhoria. A unified framework for expressing software subsystem classification techniques. *Journal of Systems and Software*, 36(3):211–231, 1997.
- [101] J. Lee. Extending the pots and bruns model for recording design rationale. In *Proceedings of the 13th International Conference on Software Engineering (ICSE 1991)*, pages 114–125. IEEE, 1991.
- [102] A. MacLean, R. M. Young, V. M. Bellotti, and T. P. Moran. Questions, options, and criteria: Elements of design space analysis. *Human-Computer Interaction*, 6(3&4):201–250, 1991.
- [103] G. Malpohl. Jplag website. <http://www.jplag.de/>.
- [104] E. Marcos. Software engineering research versus software development. *SIGSOFT Softw. Eng. Notes*, 30(4):1–7, 2005.
- [105] C. McNamara. *Field Guide to Consulting and Organizational Development: A Collaborative and Systems Approach to Performance, Change and Learning*. Authenticity Consulting, LLC, 2006.
- [106] N. Medvidovic, D. S. Rosenblum, and R. N. Taylor. A language and environment for architecture-based software development and evolution. In *Proceedings of the 21st International Conference on Software Engineering (ICSE 1999)*, pages 44–53. IEEE Computer Society Press, 1999.



- [107] N. Medvidovic and R. N. Taylor. A classification and comparison framework for software architecture description languages. *IEEE Transactions on Software Engineering*, 26(1):70–93, 2000.
- [108] N. C. Mendonça and J. Kramer. Developing an approach for the recovery of distributed software architectures. In *6th IEEE International Workshop on Program Comprehension*, pages 28–36, Ischia, Italy, June 1998. IEEE. The paper describes the initial work on the X-RAY architecture recovery approach and tools.
- [109] T. Mens, J. Buckley, M. Zenger, and A. Rashid. Towards a taxonomy of software evolution. In *Proceedings of the Second International Workshop on Unanticipated Software Evolution (USE 2003)*, April 2003.
- [110] I. G. Muhammad Ali Babar and R. Jeffery. Toward a framework for capturing and using architecture design knowledge. Technical Report UNSW-CSE-TR-0513, University of New South Wales, Australia and National ICT Australia Ltd., June 2005.
- [111] I. Nonaka. A dynamic theory of organizational knowledge creation. *Organization Science*, 5(1):14–37, February 1994.
- [112] I. Nonaka and H. Takeuchi. *The Knowledge-creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press Inc, USA, 1995.
- [113] D. Ohst, M. Welle, and U. Kelter. Differences between versions of uml diagrams. In *Proceedings of the 9th European software engineering conference held jointly with 10th ACM SIGSOFT international symposium on Foundations of software engineering*, pages 227–236. ACM Press, 2003.
- [114] OMG. Common object request broker architecture (corba/iiop). version 3.0.3. Technical Report formal/2004-03-12, Object Management Group, 2004.
- [115] P. Oreizy, N. Medvidovic, and R. N. Taylor. Architecture-based runtime software evolution. In *Proceedings of the 20th International Conference on Software Engineering (ICSE 1998)*, pages 177–186. IEEE, 1998.
- [116] H. Ossher and P. Tarr. Multi-dimensional separation of concerns and the hyperspace approach. In *Proceedings of the Symposium on Software Architectures and Component Technology: The State of the Art in Software Development*. Kluwer, 2000.
- [117] D. L. Parnas. On the criteria to be used in decomposing systems into modules. *Commun. ACM*, 15(12):1053–1058, 1972.
- [118] D. L. Parnas and P. C. Clements. A rational design process: How and why to fake it. *IEEE Transactions on Software Engineering*, 12(2):251–257, 1986.
- [119] D. E. Perry and A. L. Wolf. Foundations for the study of software architecture. *ACM SIGSOFT Software Engineering Notes*, 17(4):40–52, 1992.
- [120] C. Petrie. Constrained decision revision. In *Proceedings of the Tenth AAAI Conference*, pages 393–400, 1992.
- [121] F. Plášil, D. Bálek, and R. Janecek. Sofa/dcup: Architecture for component trading and dynamic updating. In *Proceedings of the International Conference on Configurable Distributed Systems*, page 43. IEEE Computer Society, 1998.
- [122] C. Potts and G. Bruns. Recording the reasons for design decisions. In *Proceedings of the 10th International Conference on Software Engineering (ICSE 1988)*, pages 418–427. IEEE, 1988.

- [123] C. Prehofer. Feature-oriented programming: A fresh look at objects. In *ECOOP*, pages 419–443, 1997.
- [124] C. Prehofer. An object-oriented approach to feature interaction. In *Feature Interactions in Telecommunications Networks IV*, pages 313–325. IOS Press, June 1997.
- [125] T. Qin, L. Zhang, Z. Zhou, D. Hao, and J. Sun. Discovering use cases from source code using the branch-reserving call graph. In *Proceedings of the Tenth Asia-Pacific Software Engineering Conference Software Engineering Conference (APSEC)*, page 60, Washington, DC, USA, 2003. IEEE Computer Society.
- [126] B. Ramesh and V. Dhar. Supporting systems development by capturing deliberations during requirements engineering. *IEEE Transactions on Software Engineering*, 18(6):498–510, June 1992.
- [127] A. Rashid, P. Sawyer, A. M. D. Moreira, and J. Araújo. Early aspects: A model for aspect-oriented requirements engineering. In *Proceedings of the 10th IEEE Joint International Conference on Requirements Engineering (RE 2002)*, pages 199–202. IEEE Computer Society, September 2002.
- [128] W. Regli, X. Hu, M. Atwood, and W. Sun. A survey of design rationale systems: Approaches, representation, capture and retrieval. *Engineering with Computers*, 16(3-4):209–235, December 2000.
- [129] D. Riehle and T. Gross. Role model based framework design and integration. In *Proceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications (OOPSLA)*, pages 117–133, New York, NY, USA, 1998. ACM Press.
- [130] M. P. Robillard and G. C. Murphy. Concern graphs: finding and describing concerns using structural program dependencies. In *ICSE '02: Proceedings of the 24th International Conference on Software Engineering*, pages 406–416, New York, NY, USA, 2002. ACM Press.
- [131] M. P. Robillard and G. C. Murphy. Representing concerns in source code. *ACM Trans. Softw. Eng. Methodol.*, 16(1):3, 2007.
- [132] R. Roshandel, A. V. D. Hoek, M. Mikic-Rakic, and N. Medvidovic. Mae—a system model and environment for managing architectural evolution. *ACM Trans. Softw. Eng. Methodol.*, 13(2):240–276, 2004.
- [133] S. Sarkar and S. Thonse. Eaml- architecture modeling language for enterprise applications. In *CEC-EAST '04: Proceedings of the E-Commerce Technology for Dynamic E-Business, IEEE International Conference on (CEC-East'04)*, pages 40–47, Washington, DC, USA, 2004. IEEE Computer Society.
- [134] S. Schleimer, D. S. Wilkerson, and A. Aiken. Winnowing: local algorithms for document fingerprinting. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 76–85, New York, NY, USA, 2003. ACM Press.
- [135] A. Selvin. Leveraging existing hypertext functionality to create a customized environment for team analysis. In *Proceedings of the Second International Workshop on Incorporating Hypertext Functionality Into Software Systems*, March 1996.
- [136] M. Shaw. What makes good research in software engineering? *International Journal*

- on Software Tools for Technology Transfer (STTT)*, 4(1):1–7, October 2002.
- [137] M. Shaw, R. DeLine, D. V. Klein, T. L. Ross, D. M. Young, and G. Zelesnik. Abstractions for software architecture and tools to support them. *IEEE Trans. Softw. Eng.*, 21(4):314–335, 1995.
- [138] M. Shaw and D. Garlan. *Software architecture: perspectives on an emerging discipline*. Prentice-Hall, Inc., 1996.
- [139] <http://www.sigcse.org/>. The ACM Special Interest Group on Computer Science Education (SIGCSE) website.
- [140] M. Sinnema, S. Deelstra, J. Nijhuis, and J. Bosch. Covamof: A framework for modeling variability in software product families. In *Third International Conference on Software Product Lines (SPLC)*, volume 3154 of *LNCS*, pages 197–213, 2004.
- [141] M. Sinnema, J. S. van der Ven, and S. Deelstra. Using variability modeling principles to capture architectural knowledge. In *Proceedings of the Workshop on SHaring and Reusing architectural Knowledge (SHARK 2006)*, June 2006.
- [142] Y. Smaragdakis and D. S. Batory. Implementing layered designs with mixin layers. In *Proceedings of the 12th European Conference on Object-Oriented Programming (ECCOP)*, pages 550–570, London, UK, 1998. Springer-Verlag.
- [143] G. F. Smith and G. J. Browne. Conceptual foundations of design problem solving. *IEEE Transactions on Systems, Man and Cybernetics*, 23(5):1209–1219, September 1993.
- [144] G. Sneltling and F. Tip. Reengineering class hierarchies using concept analysis. In *Proceedings of the 6th ACM SIGSOFT international symposium on Foundations of software engineering*, pages 99–110. ACM Press, 1998.
- [145] Software engineering institute software architecture definition page. <http://www.sei.cmu.edu/architecture/definitions.html>.
- [146] I. Sommerville. *Software Engineering*. Addison-Wesley, 8 edition, 2007.
- [147] Z. R. Stepenson. *Change Management in Families of Safety- Critical Embedded Systems*. PhD thesis, University of York, 2002.
- [148] M. R. T. Capturing software architecture design expertise with armani. Technical Report CMU-CS-98-163, Carnegie Mellon University School of Computer Science, October 1998.
- [149] A. Tang, M. A. Babar, I. Gorton, and J. Han. A survey of the use and documentation of architecture design rationale. In *Proceeding of the Fifth Working IEEE / IFIP Conference on Software Architecture (WICSA 2005)*, pages 89–99, November 2005.
- [150] A. Tang, M. A. Babar, I. Gorton, and J. Han. A survey of architecture design rationale. *Journal of Systems & Software*, 79(12):1792–1804, 2006.
- [151] A. Tang, Y. Jin, and J. Han. A rationale-based architecture model for design traceability and reasoning. *Journal of Systems and Software*, 80(6):918–934, June 2007.
- [152] A. Tang, Y. Jin, J. Han, and A. E. Nicholson. Predicting change impact in architecture design with bayesian belief networks. In *Proceeding of the Fifth Working IEEE / IFIP Conference on Software Architecture (WICSA 2005)*, pages 67–76. IEEE Computer Society, November 2005.
- [153] P. Tarr, H. Ossher, W. Harrison, and J. Stanley M. Sutton. N degrees of separation:

- multi-dimensional separation of concerns. In *Proceedings of the 21st international conference on Software engineering*, pages 107–119. IEEE, 1999.
- [154] R. N. Taylor, N. Medvidovic, K. M. Anderson, E. J. Whitehead, Jr., J. E. Robbins, K. A. Nies, P. Oreizy, and D. L. Dubrow. A component- and message-based architectural style for gui software. *IEEE Trans. Softw. Eng.*, 22(6):390–406, 1996.
- [155] P. Tonella and A. Potrich. Static and dynamic c++ code analysis for the recovery of the object diagram. In *Proceedings of the International Conference on Software Maintenance*, pages 54–63, October 2002.
- [156] P. W. Trygve Reenskaug and O. A. Lehne. *Working with Objects The OOram Software Engineering Method*. Manning Publications, 1995.
- [157] C. R. Turner, A. Fuggetta, L. Lavazza, and A. L. Wolf. A conceptual basis for feature engineering. *Journal of Systems & Software*, 49(1):3–15, 1999.
- [158] J. Tyree and A. Akerman. Architecture decisions: Demystifying architecture. *IEEE Software*, 22(2):19–27, 2005.
- [159] V. Tzerpos and R. C. Holt. ACDC: An algorithm for comprehension-driven clustering. In *Working Conference on Reverse Engineering (WCRE 2000)*, pages 258–267. IEEE Computer Society, 2000.
- [160] <http://www.uml.org/>. The Unified Modeling Language (UML) website.
- [161] A. van der Hoek, M. Mikic-Rakic, R. Roshandel, and N. Medvidovic. Taming architectural evolution. In *Proceedings of the 8th European software engineering conference*, pages 1–10. ACM Press, 2001.
- [162] J. S. van der Ven, A. G. J. Jansen, P. Avgeriou, and D. K. Hammer. Using architectural decisions. In *Second International Conference on the Quality of Software Architecture (Qosa 2006)*, 2006.
- [163] J. S. van der Ven, A. G. J. Jansen, J. A. G. Nijhuis, and J. Bosch. Design decisions: The bridge between rationale and architecture. In A. H. Dutoit, R. McCall, I. Mistrik, and B. Paech, editors, *Rationale Management in Software Engineering*, chapter 16, pages 329–348. Springer-Verlag, March 2006.
- [164] A. van Deursen. Software architecture recovery and modelling: [wcre 2001 discussion forum report]. *ACM SIGAPP Applied Computing Review*, 10(1):4–7, 2002.
- [165] A. van Deursen, M. de Jonge, and T. Kuipers. Feature-based product line instantiation using source-level packages. In *Proceedings of the Second International Conference on Software Product Lines (SPLC)*, pages 217–234, London, UK, 2002. Springer-Verlag.
- [166] A. van Deursen, C. Hofmeister, R. Koschke, L. Moonen, and C. Riva. Symphony: View-driven software architecture reconstruction. In *Proceedings of the 4th IEEE/IFIP Working Conference on Software Architecture (WICSA 2004)*, page 122. IEEE Computer Society, 2004.
- [167] A. van Deursen and T. Kuipers. Identifying objects using cluster and concept analysis. In *21st International Conference on Software Engineering, ICSE-99*, pages 246–255. ACM, 1999.
- [168] J. van Gorp and J. Bosch. Design erosion: Problems & causes. *Journal of Systems & Software*, 61(2):105–119, March 2002.

- [169] R. van Ommering. *Building Product Populations with Software Components*. PhD thesis, University of Groningen, 2004.
- [170] R. van Ommering, F. van der Linden, J. Kramer, and J. Magee. The koala component model for consumer electronics software. *IEEE Computer*, 33(3):78–85, March 2000.
- [171] Velocity website, <http://jakarta.apache.org/velocity>.
- [172] M. Visconti and C. R. Cook. Assessing the state of software documentation practices. In F. Bomarius and H. Iida, editors, *PROFES*, volume 3009 of *Lecture Notes in Computer Science*, pages 485–496. Springer, 2004.
- [173] Z. Wang, K. Sherdil, and N. H. Madhavji. ACCA: An architecture-centric concern analysis method. In *5th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, pages 99–108, November 2005.
- [174] D. W. Weber. Change sets versus change packages: Comparing implementations of change-based scm. In *Proceedings of the SCM-7 Workshop on System Configuration Management*, pages 25–35. Springer-Verlag, 1997.
- [175] <http://www.webster.com>, 2006.
- [176] T. Wiggerts, H. Bosma, and E. Fiel. Scenarios for the identification of objects in legacy systems. In *Fourth Working Conference on Reverse Engineering (WCRE '97)*, pages 24–32. IEEE Computer Society, October 1997.
- [177] H. Yan, D. Garlan, B. R. Schmerl, J. Aldrich, and R. Kazman. Discotect: A system for discovering architectures from running systems. In *Proceedings of the 26th International Conference on Software Engineering (ICSE 2004)*, pages 470–479. IEEE Computer Society, 2004.
- [178] P. Zave. Feature-oriented description, formal methods, and dfc. In *Proceedings of the FIRE-works Workshop on Language Constructs for Describing Features*, pages 11–26, May 2000.
- [179] M. V. Zelkowitz and D. R. Wallace. Experimental validation in software engineering. *Information & Software Technology*, 39(11):735–743, 1997.
- [180] H. Zhuge. *The Knowledge Grid*. World Scientific Publishing Company, 2004.
- [181] B. Zimmermann and A. M. Selvin. A framework for assessing group memory approaches for software design projects. In *Proceedings of the conference on Designing interactive systems*, pages 417–426. ACM Press, 1997.