

James Noble

Curriculum Vitae

Full Name Robert James Noble.
Qualifications **B.Sc.(Hons) Ph.D. VUW**
Nationality New Zealand Citizen, British Citizen.
Date and place of birth 17 August 1967, London, England.

Address for correspondence

James Noble,
Creative Research & Programming,
5 Fernlea Ave,
Darkest Karori, Wellington,
New Zealand.

Telephone: +64 21 321 559
Internet Mail: kjx@acm.org
WWW: <http://www.ecs.vuw.ac.nz/~kjx>

Present Position

I am independent creative researcher and programming, based in Darkest Karori, Wellington, NZ.

Professional Memberships

Fellow, Information Technology Professionals, New Zealand
Fellow of the British Computer Society
Senior Member of the Association for Computing Machinery
Member of the Institute of Electrical and Electronics Engineers
Member of the Association Internationale pour les Technologies Objets
Certified Information Technology Professional, New Zealand

Research Areas

I conduct research in the following areas of Software Engineering and Computer Science:

Software Design especially Object-Oriented and Formally Correct approaches to design.
Programming Languages especially Ownership Types and Domain Specific Extensible Languages.
Design Patterns including Small Memory Systems; User Interaction; Agile Development.
Human Computer Interaction including Interface & Interaction Design, IDEs, Usability.
Software Visualisation and **Visual Languages** including design of development environments .
Development Methodologies especially Agile approaches involving users and customers.
Philosophy of Computer Science and Software Engineering, in particular semiotics of computation and postmodern approaches to software design.
Creative Programming especially live coding, modular synthesizer patching, computer music.

Teaching Experience

Since 2000 I have developed and taught courses from introductory computer science to specialized international tutorials.

Appointments Held

(Teaching appointments marked *)

Adjunct Professor
Software Engineering, Victoria University of Wellington
February 2022 — February 2024

* Professor
Computer Science, Victoria University of Wellington
December 2003 — February 2022

Visiting Professor
Department of Computing, Kings College, London
June 2016 – July 2019

Visiting Professor
Department of Computing, Imperial College, London
October 2009 – September 2017

* Associate Professor
Computer Science, Victoria University of Wellington
January 2003 — November 2003

* Senior Lecturer
Computer Science, Victoria University of Wellington
January 2002 — December 2002

* Lecturer
Computer Science, Victoria University of Wellington
October 1999 — December 2001

* Research Scientist
Microsoft Research Institute, Macquarie University,
July 1996 — October 1999.

* Postdoctoral Research Fellow
Center for Object Technology, Applications, and Research,
Department of Computing Sciences, University of Technology, Sydney,
May 1995 — June 1996.

* Lecturer
Department of Computing Sciences, University of Technology, Sydney,
August 1995 — November 1995.

* Assistant Lecturer
Victoria University of Wellington,
August 1992 — January 1993.

* Tutor, Victoria University of Wellington,
1993 — 1994.

* Teaching Assistant, Victoria University of Wellington,
February 1989 — July 1992.

* Programmer, Coopers & Lybrand, Wellington,
November 1986 - November 1988

Graduate Students

David Holmes, Principal Member of Technical Staff, Oracle Corporation. PhD 1999
Dave Clark, Professor, Computer Science, Uppsala University. PhD 2002
Michael Richmond, Principal Architect, NIO (San Jose). PhD 2002
Stuart Marshall, Senior Lecturer, Victoria University of Wellington. PhD 2006
Alex Potanin, Associate Professor, Victoria University of Wellington, PhD 2007
Rilla Khaled, Associate Professor, Concordia University. PhD 2008
Pippin Barr, Assistant Professor, Concordia University. PhD 2008.
Matthew Duignan, Product Manager, Microsoft Corporation. PhD 2008.
Angela Martin, Agile Team Facilitator, Xero. PhD 2009
Rashina Hoda, Associate Professor, Monash University. PhD 2011
Stephen Nelson, IP Projects Lead, Katalyst Interactive (Adelaide), PhD 2012
Craig Anslow, Senior Lecturer, Victoria University of Wellington, PhD 2013
Siva Kumar Dorairaj. Senior Consultant, Assurity. PhD 2013
Michael Homer, Lecturer Victoria University of Wellington, PhD 2014
Michael Waterman Solution Architect, GWRD. PhD 2014
Ahmed Khalifa. Developer. PhD 2014
Paley Li, Principal Member of the Technical Staff, Oracle Laboratories, PhD 2015
Tim Jones. Senior Software Engineer. Montoux. New York. PhD 2017
Mawarny Md Rejab Lectuer, Universiti Utara Malaysia. PhD 2017

Fiona Malikoff. Software Developer (Brisbane). MSc 1999
Rilla Khaled, Associate Professor, Concordia University. MSc 2002
Pippin Barr, Assistant Professor, Concordia University. MSc 2003
Michael McGavin, Solutions Developer, Kapish. MSc 2005
Darren Willis. Technical Lead Manager, Google (Tokyo). MSc 2006
Jennifer Ferreria, Lecturer, Victoria University of Wellington. MSc 2007
Craig Anslow, Senior Lecturer, Victoria University of Wellington, MSc 2008
Donald Gordon. Minister. PCANZ. MSc 2008
Frank Schmager, Software Architect, Bandwith.com. MSc 2011
Radu Muschevici. Research Fellow, TU Darmstadt. MSc 2011
Jack Purvis. Software Developer, Optimal Workshop. MSc 2020.

Haowei Ruan. Senior Solution Architect, Aon (Sydney). PGDipSci 2010

Currently supervising 1 PhD student.

Prizes and Awards

International Conference on Software Engineering “Distinguished Paper Award” – 2017, 2015.

Association Internationale pour les Technologies Objets (AITO) “Dahl-Nygaard Senior Career Award” 2016

Association for Computing Machinery Special Interest Group on Programming Languages (ACM SIGPLAN) “Most Notable Onward Paper in the last 10 years” Award 2012

VUW PGSA Award: Best Supervisor in the Faculty of Engineering 2010.

Asia-Pacific Software Engineering Conference “Best Paper Award” 2010.

Association for Computing Machinery Special Interest Group on Programming Languages (ACM SIGPLAN) “Most Influential OOPSLA Paper in the last 10 years” Award 2008

Australian Software Engineering Conference “Most Influential Paper in the last 10 years” Award 2008

VUW Research Excellence Award 2008

Neil B. Harrison Shepherding Award, Ninth European Conference on Pattern Languages of Program Design (EuroPLoP) 2004

Best Paper Award, Eighth European Conference on Pattern Languages of Program Design (EuroPLoP) 2003

External Research Support

Gradual concurrency: correctness, simplicity, and performance via dynamic ownership. James Noble, Tobias Wrigstad. Julian Mackay. Royal Society of New Zealand: Marsden Fund. 21-VUW-145. NZD 715,000. 2021 Round (2022-2024)

“Programming Made Hard” Made Easier: Improving Dafny’s Human Factors. James Noble. Amazon Research Award. USD 60,000. 2022.

Holistic Specifications. James Noble. Agoric Inc. USD 10,000. 2021.

Object Capabilities for Robust Applications in the Open World. James Noble, Sophia Drossopoulou. Royal Society of New Zealand: Marsden Fund. 18-VUW-026. ORCA: NZD 679,000. 2018 Round (2019-2021)

Object Ownership for Capabilities and Provenance. Alex Potanin, David J. Pearce, James Noble. Oracle Corporation (Australia). NZD 67,081. 2017.

Making Capability Policies Explicit. James Noble, Sophia Drossopoulou. Royal Society of New Zealand: Marsden Fund. 13-VUW-062. CAPE: NZD 521,739. 2013 Round (2014-2017)

Reliable Software via Patterns and Ownership James Noble. Royal Society of New Zealand: James Cook Fellowship. 13-VUW-001. NZD 220,000. 2013 Round (2014-2016).

Visualisation of Large Codebases. Craig Anslow, Stuart Marshall, James Noble. Oracle Corporation (Australia). NZD 58,519. 2014.

Ministry of Science and Industry . Tech Jumpstart. Craig Anslow, S. Marshall, James Noble. NZD 10,000. 2013

Software Visualisation for Large Code Bases. James Noble, Stuart Marshall. Oracle Corp.,

NZD61,940, 2013

Ministry of Science and Industry. New Economy Research Fund Contract Extension. NZD 78,518.72. Sept 2011 – Sept 2012.

Design patterns and ownership types for reusable program verification. James Noble, Sophia Drossopoulou. RSNZ Marsden Fund, 2011-2013. NZD 640,000

More Expressive Ownership Types. James Noble. Build IT Postdoctoral Fellowship. NZD 116,000, July 2008-2010

Scoped Memory Types for Safety Critical RTSJ. James Noble. IBM Real Time Innovation Award. USD 16,000. Dec 2007

Software Product and Process Improvement for New Zealand Industry. John Grundy (Science Leader). John Hosking, Emilla Mendes, Stephen MacDonell, James Noble (Key Researchers). FRST New Materials, Technologies and Services. October 2007—June 2011.

Practical Ownership Types for Object and Aspect Programs Sophia Drossopoulou, James Noble and Dave Clarke. EPSRC, GBP 52,000, 2006.

Integrated Ownership and Debugging in Eclipse. James Noble, Robert Biddle. IBM Eclipse Innovation Award. USD 15,000. 2005

Eclipse support for Ownership Types. James Noble, Robert Biddle. IBM Eclipse Innovation Award. USD 20,000. 2004

Learning the Aspects of Object-Oriented Programming. Robert Biddle, James Noble. IBM Eclipse Innovation Award. USD 20,000. 2004

Ownership types for Object- and Aspect-Oriented Programming. James Noble, David Clarke, Robert Biddle. Marsden Grant. NZD 327,678 2004-2006.

Domain Specific Software Tools. John Hosking, John Grundy, Ewan Tempero, Emilia Mendes, Anna Liu, James Noble, Mark Apperly, Robert Biddle, Robert Amor, Nianping Zhu. Public Good Science Fund. NZD 762,000. 2002-2005.

Aliasing and Ownership in Object-Oriented Systems. James Noble. Fast Start Marsden Grant NZD 100,000. 2002-2003.

Object-Oriented Frameworks. Robert Biddle, James Noble, and Information Power Ltd. Technology for Business Growth Grant NZD 50,000. 2002-2003.

Interactive Metrics Visualisation for User Interface Design. James Noble and Larry Constantine. ARC Small Grant, 1998. AUD 16 000.

Scholastic and University Record

Ph.D., Victoria University of Wellington, 1996.

J.L. Stewart Scholarship, 1989.

IBM Postgraduate Scholarship, 1989 — 1992.

UGC Postgraduate Scholarship, 1989 — 1992.

B.Sc.(Hons) 1st class, Victoria University of Wellington, 1988.

B.Sc., Victoria University of Wellington, 1985 — 1987.

External Service

Appointments

Associate Editor, ACM Transactions on Programming Languages and Systems (ToPLAS), 2020-present.

Board Member, VeracityLab, 2021-2026

Panel member for Royal Society of New Zealand Marsden Fund Mathematical and Information Sciences (MIS) panel, 2006-2008, 2014.

Foundation Editor-In-Chief, Springer-Verlag Transactions on Pattern Languages of Programming, 2007-2019

Editorial Board member of journals: IET Software, Systems Signs and Actions; Intl. Journal Agile and Extreme Software Development, Journal of Object Technology.

Director of the Hillside Group, a US-based non-profit organization that promotes the study of design patterns worldwide, 2001-2008.

External Examiner for Masters & Doctoral Theses

The University of Melbourne, 2019

Università della Svizzera Italiana Lugano University, 2013

Monash University, 2005, 2012

Australian National University 2012

Queensland University of Technology, 2010

Massey University 2009, 2010

The University of Auckland, 2008, 2009

The University of New South Wales, 2007

The University of Madeira, Portugal, 2006

The University of Sheffield, 2005

Flinders University of South Australia, 2005

The University of Auckland 2004, 2003

Dublin City University, 2004

The Royal Melbourne Institute of Technology, 2002

Research Assessments

Member of the College of Assessors, Ministry of Business, Innovation & Enterprise, NZ. 2016-present

Member, Commission d'Evaluation, Institute of Computer Science France (INRIA) 2021-2022

Reviewer for Swiss National Science Foundation (SNSF) 2022

Reviewer for Flemish Research Organisation (FWO) 2018

Reviewer for Canadian Natural Sciences Engineering Research Council (NSERC) 2011, 2013, 2018.

Reviewer for Norwegian Centres of Research Excellence, 2017
Reviewer for British Computer Society Lovelace Medal, 2014.
Reviewer for Deutsche Forschungsgemeinschaft Gottfried Wilhelm Leibniz Prize, 2014
Reviewer for European Research Council, 2014.
Reviewer for Israel Science Foundation 2013
Reviewer for KU Leuven Research Council, 2012.
External Research Quality Evaluator, The Australian National University, 2012.
External Research Assessor, Excellence in Research Australia, 2012.
Reviewer for Canadian MITACS grant body, 2010.
Reviewer for ETH Zurich Research Commission, 2010
Reviewer for Hasler Stiftung, 2010
Reviewer for Hong Kong Research Grants Committee, 2009, 2012, 2017, 2021

Conference Committees

Object-Oriented Programming Systems, Languages, and Application (OOPSLA)

- 2001, 2002, 2005, 2006, 2009, 2012, 2015, 2021
- External Review Committee 2017
- Organising Committee OOPSLA 2002, 2005, 2021-2023.
Co-Chair, OOPSLA 2005 Onward! Stream
- Chair, OOPSLA 2002 Doctoral Symposium
- Doctoral Symposium Faculty Member 2001, 2003, 2006, 2007
- Onward! Essays Programme Committee, 2019

European Conference on Object-Oriented Programming (ECOOP)

- 2001, 2005, 2007, 2008, 2009, 2010, 2011, 2022
- Programme Chair 2012
- Inaugural Summer School Chair 2008-2014, 2019
- Doctoral Symposium Academic Panel Member 2008.

Dynamic Language Symposium (DLS) 2009, 2013-2015, 2019, 2020

- Programme Chair, 2009.

Software Language Engineering (SLE) 2018, 2020

Foundations of Software Engineering / European Software Engineering Conference (FSE/ESEC) 2017, 2021

International Conference on Software Engineering (ICSE)

- New Ideas and Emerging Results (NIER) 2015
- Student Research Competition (SRC) 2012

International Conference on Live Coding (ICLC) 2015-2019

IEEE Working Conference on Software Visualization (VISSOFT) 2015, 2022

Agile Processes in Software Engineering and Extreme Programming (XP) 2015

ACM Modularity

- Visions track 2015

Australian Software Engineering Conference (ASWEC)

- Programme co-Chair 2010, 2013

ACM Programming Languages Design and Implementation (PLDI).

- External Review Committee 2012, 2015

Aspect Oriented Software Development (AOSD) 2008, 2012

International Symposium On Wikis (WikiSym)

- Steering Committee 2005-2010
- Programme Chair WikiSym 2006
- Programme Committee WikiSym 2005-2008,2010.

ACM Hypertext 2007

Programme Chair for forUSE 2002.

Technology of Object-Oriented Languages and Systems (TOOLS Europe), 2010, 2011.

Technology of Object-Oriented Languages and Systems (TOOLS Pacific), 1997-2002, 2019.

- Program chair for TOOLS Pacific 2002.

Asia-Pacific Conference on Conceptual Modelling (ACCPM) 2009

Australian User Interface Conference 2000, 2002, 2004, 2005

Australian Computer Science Conference 2002, 2003. 2004.

Computing: Australian Theory Symposium (CATS) 2009.

European Conference on Pattern Languages of Programming (EuroPLOP) from 1998 to 2001.

- Focus Group Chair for EuroPLOP 2000.

European Symposium on Programming (ESOP) 2009, external reviewer 2012

Asian-Pacific Pattern Languages of Programming Conference (KoalaPLoP) 2000-2002.

- Conference Chair 2001
- Program Chair 2002.

Visual Languages/Human Centric Communication (VL/HCC) 2006, 2007, 2008

Diagrams 2008

Software Engineering - SIENZ 2007 Research Stream Programme Panel

INTERACT'99, 2001.

International Conference on Software Engineering (ICSE)^[SEP] New Ideas and Emerging Results (NIER) 2013, 2015^[SEP] Student Research Competition (SRC) 2012

Workshop Committees

International Workshop on Live Programming (LIVE) Steering Committee 2019-present, Reviewer 2018-present

Workshop on Cooperative and Human Aspects of Software Engineering (CHASE) at the International Conference on Software Engineering (ICSE) 2012, 2014, 2015, 2016, 2019, 2021

Workshop on New Object-Oriented Languages (NOOL) at SPLASH. 2015-2018

ECOOP Workshop: Formal Techniques for Java-like Programs (FTfJP) 2003, 2004, 2009, 2017, 2022

ECOOP Workshop from Scripts to Programs (STOP) 2015

Future Programming Workshop at Strangeloop and PLOP: 2014

ECOOP Workshop on Dynamic Languages (DYLA) 2011

Workshop on Evaluation and Usability of Programming Languages and Tools (PLATEAU) at the ACM Conference on Systems, Programming, Languages, Applications, for Humanity (SPLASH) 2010, 2014, 2015

ECOOP 2011 Intercontinental Workshop on Aliasing, Confinement, and Ownership in Object-Oriented Systems (IWACO): Twenty Years of Aliasing in Object-Oriented Languages. Nick Cameron, Dave Clarke, James Noble, Matthew Parkinson.

Dagstuhl 2010 Workshop on Relationships, Objects, Roles, and Queries in Modern Programming Languages, Stephanie Balzer, Gavin Bierman, Stephan Hermann, Stephen Nelson, Erik Meijer, James Noble, David Pearce, Friedrich Steimann, Frank Tip.

OOPSLA 2008 Relationships and Associations in Object-Oriented Languages. Stephen Nelson, Stephanie Balzer, Gavin Bierman, Erik Meijer, James Noble, David Pearce, Jiri Soukup, Frank Tip.

International Workshop on Aliasing, Confinement, and Ownership in Object-Oriented Systems (IWACO), at ECOOP. 1999, 2003, 2007, 2011, 2017.

FOOL/WOOD Foundations of Object-Oriented Languages workshop and the Workshop on Object-Oriented Developments 2006

OOPSLA 2002 Workshop on Refactoring Design Patterns. Brian Foote, James Noble, Charles Weir.

OOPSLA'98 Workshop on Prototype-Based Object-Oriented Programming. Antero Taivalsaari and James Noble.

ECOOP'97 Workshop on Prototype-Based Object-Oriented Programming. James Noble and Ivan Moore.

ECOOP'96 Workshop on Prototype-Based Object-Oriented Programming. Antero Taivalsaari and James Noble.

Journal Reviewing

ACM Transactions on Software Engineering and Methodology (TOSEM) 2017

ACM Transactions on Programming Languages and System (TOPLAS) 2011, 2014, 2021

Communications of the ACM (CACM) 2015

Information and Software Technology (IST) 2015

IEEE Transactions on Software Engineering. (TSE) 2007, 2008, 2011, 2012

IEEE Software 2008

Software—Practice & Experience 2008, 2011, 2012, Distinguished Reviewer 2018.

Empirical Software Engineering (ESE) 2018

Journals of System and Software (JSS), 2015, 2018

Automated Software Engineering (ASE) 2018

Science of Computer Programming (SCP) 2011 2017

ACM Inroads 2017

Journal of Visual Languages and Computation 2008

Invited Presentations / Keynotes

N things I hate about \nu*. James Noble. Invited presentation, <Programming '22> conference (online). 2022

programmingLanguage as Language. James Noble and Robert Biddle. Invited Keynote Presentation, ACM History of Programming Languages conference (HOPL), 2021.

The Act of Computer Programming, James Noble. Invited keynote presentation to Australian Computer Science Week (ACSW) 2021.

The Act of Computer Programming, James Noble. Invited keynote presentation to LIVE 2019 at OOPSLA 2019

"Holistic Specifications", James Noble. Public Lecture, Department of Computing, The University of Edinburgh 2019

"Through a Glass, Darkly (John Disco remix)" James Noble. Invited Keynote Presentation, Scottish Programming Language Society Nov 2019

Twenty Years of Ownership Types. James Noble. Invited presentation to SPLASH-I at OOPSLA 2018.

Through a glass, darkly. James Noble. Invited Keynote presentation to Workshop on Meta-Programming Techniques and Reflection (META) at OOPSLA 2018.

Half a Life in Software. James Noble. Keynote presentation to the European Conference on Object-Oriented Programming (ECOOP), 2016.

We Demand Rigidly Defined Areas of Doubt and Uncertainty. James Noble. Keynote presentation to the Workshop on Aspect-Oriented Software Development (AOSD 2016), at ECOOP 2016.

Agile Research into Agile Development. James Noble. Invited Keynote presentation to the 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE 2013) at ICSE 2013. May 2013.

Twenty Years of Aliasing in Object-Oriented Languages (a personal view). James Noble. Keynote presentation to the International Workshop on Aliasing, Confinement, and Ownership in Object-Oriented Systems at ECOOP 2011. July 2011.

We Demand Rigidly Defined Areas of Doubt and Uncertainty. James Noble. Keynote presentation to the 5th Asian Workshop on Aspect-Oriented and Modular Software Development (AOAsia 2009), at ASE 2009, November 2009

The Myths of Object-Orientation. James Noble. Invited presentation, ECOOP, Paphos. 2008.

We Demand Rigidly Defined Areas of Doubt and Uncertainty. James Noble. Keynote presentation to the ECOOP 2008 Workshop on Aspect Dependencies and Interactions, July 2008

The Lego Hypothesis. James Noble. Keynote presentation to QCon San Francisco, Nov 2007.

Roles and Relationships James Noble. Keynote presentation to ECOOP Workshop on Roles and Relationships, Berlin, July 2007.

The Lego Hypothesis. James Noble. Keynote presentation to OOP, Munich, Jan 2007.

Love in the Age of Software. James Noble and Robert Biddle. Keynote Presentation to XPDay, London, Nov 2006.

Postmodern Prospects for Conceptual Modelling. James Noble, Robert Biddle. Invited Presentation to

Asian-Pacific Conference on Conceptual Modelling (APCCM), Hobart, Jan 2006.

The Lego Hypothesis. James Noble. Keynote presentation to JAOO 2004, Aarhus, Denmark, Sept 2004.

Conference Tutorials

Notes on Postmodern Programming. James Noble. Curry On 2016.

Grace: an educational, object-oriented, open-source, portable, programming language. James Noble, Andrew Black, Kim Bruce. PLDI2012

Customers are from Venus, Programmers are from Mars. Angela Martin, Robert Biddle, James Noble. XP 2006, Agile 2006, OOPSLA 2006, XP 2006, XPDay 2006. Agile 2007.

The Six Million Dollar Customer. Angela Martin, Robert Biddle, James Noble. OOPSLA 2005.

Semiotics of Visual Languages. James Noble, Robert Biddle. VL/HCC2004.

Object-Oriented Techniques That Really Work. Robert Biddle, James Noble, Ewan Tempero. forUSE2002.

Patterns for Small Machines. Charles Weir and James Noble. OOPSLA 2000-2002, TOOLS Europe 2000, JAOO 2004

Playing with Patterns. Charles Weir and James Noble. OOPSLA'98, 99, 2000. TOOLS Europe 2000

Usage-Centered Design. Larry Constantine, Lucy Lockwood, and James Noble. OzCHI'98, OOPSLA'99, 2000. 2001. 2002, forUSE 2002.

Applying Design Patterns, John Potter and James Noble. TOOLS Pacific, Melbourne, 1997, ASWEC, Adelaide, 1998.

Industrial Courses

Object-Oriented Techniques That Really Work. Robert Biddle, James Noble, Ewan Tempero. Presented to various New Zealand companies, 2000-2002.

James Noble

Research and Publications

Book

1. *Small Memory Software: Patterns for Systems with Limited Memory*. James Noble and Charles Weir. Addison-Wesley, London. October 2000.

Edited Books

2. *Aliasing in Object-Oriented Systems — Types, Analysis, Verification*. Dave Clarke, Tobias Wrigstad, James Noble, editors. Springer Verlag LNCS 7850. 2013.
3. *Pattern Languages of Program Design 5*. Dragos Manolescu, Marcus Volter, and James Noble, editors. Addison-Wesley, 2006.
4. *Prototype-Based Programming: Concepts, Languages, Applications*. James Noble, Antero Taivalsaari, Ivan Moore, editors. Springer-Verlag, 1999.

Edited Journal Issue

5. Transactions on Pattern Languages of Programming IV — James Noble, Ralph Johnson, Uwe Zdun, Eugene Wallingford, editors. Springer-Verlag Lecture Notes in Computer Science (LNCS 10600) 2019.
6. Transactions on Pattern Languages of Programming III — James Noble, Ralph Johnson, Uwe Zdun, Eugene Wallingford, editors. Springer-Verlag Lecture Notes in Computer Science (LNCS 7840) 2013.
7. Transactions on Pattern Languages of Programming II — Special Issue on Applying Patterns. James Noble, Ralph Johnson, Paris Avgeriou, Neil B. Harrison, Uwe Zdun, editors. Springer-Verlag Lecture Notes in Computer Science (LNCS 6510) 2011.
8. Transactions on Pattern Languages of Programming I. James Noble, Ralph Johnson, editors. Springer-Verlag Lecture Notes in Computer Science (LNCS 5770) 2010.
9. Special issue of *SoftwarePractice and Experience* on Aliasing in Object-Oriented Systems. James Noble and Doug Lea, editors. 31:1, 2001.

Edited Conference Proceedings

10. Collaboration and learning through live coding (Dagstuhl Seminar 13382). Alan Blackwell, Alex McLean, James Noble, Julian Rohrerhuber. *Dagstuhl Reports* 3(9): 130-168 (2013)
11. Proceedings of the 26th European Conference on Object-Oriented Programming (ECOOP 2012), Beijing, China, June 11-16, 2012. James Noble, Editor. Springer 2012
12. Proceedings of the 21st Australian Software Engineering Conference (ASWEC 2010). James Noble and Colin Fidge, Editors. IEEE Press, 2010.
13. Proceedings of the 5th Symposium on Dynamic languages (DLS 2009). James Noble, Editor. ACM Press, 2009
14. Proceedings of the 2006 International Symposium on Wikis (WikiSym 2006). Dirk Riehle, James Noble, Editors. ACM Press, 2006.
15. Proceedings of the Third Asian-Pacific Conference on Pattern Languages of Programs KoalaPLoP 2002. James Noble and Paul Taylor, Editors. Volume 16 of *Conferences on*

Research and Practice in Information Technology, Australian Computer Society, 2002.

16. Objects for Mobile, Web, and Internet Computation: Proceedings of the Fortieth International Conference on Technology of Object-Oriented Languages and Systems - TOOLS Pacific 2002, Sydney, Australia, 2002. James Noble and John Potter. Editors. Volume 10 of *Conferences in Research and Practice in Information Technology*.
17. Proceedings of the Second Asian-Pacific Conference on Pattern Languages of Programs KoalaPLOP 2001. James Noble and Brian Wallis, Editors. Technical Report CS-TR-01-8, Computer Science, Victoria University of Wellington, New Zealand.

Refereed Book Chapters

18. Abstract and Concrete Data Types vs Object Capabilities. James Noble, Alex Potanin, Toby C. Murray, Mark S. Miller. In *Principled Software Development: Essays Dedicated to Arnd Poetzsch-Heffter on the Occasion of his 60th Birthday*. Springer-Verlag 2018: 221-240
19. The Essence of Inheritance. Andrew Black, Kim Bruce and James Noble. In *A List of Successes that can Change the World*. Sam Lindley, Conor McBride, Phil Trinder, Don Sannella, editors. Springer Verlag LNCS 2016.
20. Understanding Ownership Types with Dependent Types. Nicholas Cameron, Sophia Drossopoulou, James Noble. In *Aliasing in Object-Oriented Systems — Types, Analysis, Verification*. Dave Clarke, Tobias Wrigstad, James Noble, editors. Springer Verlag LNCS 7850. 2013.
21. Beyond the Geneva Convention on the Treatment of Object Aliasing. Dave Clarke, James Noble, Tobias Wrigstad. In *Aliasing in Object-Oriented Systems — Types, Analysis, Verification*. Dave Clarke, Tobias Wrigstad, James Noble, editors. Springer Verlag LNCS 7850. 2013.
22. How, Then, Should We Program? James Noble. In *Aliasing in Object-Oriented Systems — Types, Analysis, Verification*. Dave Clarke, Tobias Wrigstad, James Noble, editors. Springer Verlag LNCS 7850. 2013.
23. An Ideal Customer: A Grounded Theory of Requirements Elicitation, Communication and Acceptance on Agile Projects. Angela Martin, Robert Biddle, James Noble. In *Agile Software Development: Current Research and Future Directions*. Nils Brede Moe, Tore Dybå, Torgeir Dingsøy, Editors. Springer Verlag. 2010.
24. A Semiotic Model of User-Interface Metaphor. Pippin Barr, Robert Biddle, James Noble. In *Virtual, Distributed and Flexible Organisations*, Kluwer Academic Publishers, 2004.
25. Visual Program Visualisation. James Noble and Robert Biddle. Chapter 12, pages 359-388 in *Software Visualisation: From Theory to Practice*. Kang Zhang, editor. Kluwer. 2003
26. Visualising Objects: Abstraction, Encapsulation, Aliasing and Ownership. James Noble. In Chapter 1, pages 58-72 of *Software Visualisation, State of the Art Survey*. Springer-Verlag Lecture Notes in Computer Science (LNCS 2269). 2002.
27. The Object System Pattern. James Noble. Chapter 5, pages 53-72 in *Pattern Languages of Program Design 4*. Neil Harrison, Brian Foote and Hans Rohnert, editors. Addison-Wesley, 2000.
28. Basic Relationship Patterns. James Noble. Chapter 6, pages 73-89 in *Pattern Languages of Program Design 4*. Neil Harrison, Brian Foote and Hans Rohnert, editors. Addison-Wesley, 2000.
29. High-level and Process Patterns from the Memory Protection Society. James Noble and Charles Weir. Chapter 12, pages 221-238 in *Pattern Languages of Program Design 4*. Neil Harrison, Brian Foote and Hans Rohnert, editors. Addison-Wesley, 2000.
30. Prototype-Based Programming for Abstract Program Visualisation. James Noble. In *Prototype-Based Programming: Concepts, Languages, Applications*. James Noble, Antero Taivalsaari, Ivan Moore, editors. Springer-Verlag, 1999.

Refereed Journal Articles

31. Challenging software developers: dialectic as a foundation for security assurance techniques. Charles Weir, Awais Rashid, James Noble. *J. Cybersecur.* 6(1) 2020
32. Long Term Software Security: Creating a Lightweight Program of Assurance Techniques for Developers. Charles Weir, Ingolf Becker, James Noble, Lynne Blair, M. Angela Sasse, Awais Rashid *Software--Practice & Experience* 50(3) 275-289, 2020
33. Lessons in Combining Block-Based and Textual Programming. Michael Homer, James Noble. *Journal of Visual Languages and Sentient Systems.* 3:22-39 01 Jul 2017
34. Grace's Inheritance. James Noble, Andrew Black, Kim Bruce, Michael Homer, Tim Jones. *Journal of Object Technology (JOT)*16(2) 2:1-35 2017.
35. Self-Organizing Roles on Agile Software Development Teams, Rashina Hoda, James Noble, Stuart Marshall. *IEEE Transactions on Software Engineering (TSE)*, 39(3): 442-444. 2013.
36. Developing a Grounded Theory to Explain the Practices of Self-Organizing Agile Teams. Rashina Hoda, James Noble, Stuart Marshall. *Empirical Software Engineering.* 17(6), 609-639 2012.
37. Implementing a language with Flow-Sensitive and Structural Typing on the JVM. David J. Pearce, James Noble. *Electronic Notes on Theoretical Computer Science.* 279(1), 47-59. 2011.
38. The Impact of Inadequate Customer Involvement on Self-Organizing Agile Teams. Rashina Hoda, James Noble, Stuart Marshall. *Information Software and Technology.* Vol 53 Issue 5 pp.521-534. 2011.
39. Abstraction and Activity in Computer Mediated Music Production. Matthew Duignan, James Noble, Robert Biddle. *Computer Music Journal*, 34:4, Winter 2010.
40. JavaCOP: Declarative Pluggable Types for Java. Shane Markstrum, Daniel Marino, Matthew Esquivel, Todd Millstein, Chris Andreae, James Noble. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 23:2, Article 4, 2010.
41. Implicit ownership types for memory management. Tian Zhao, Jason Baker, James Hunt, James Noble, Jan Vitek. *Science of Computer Programming.* 71(3): pp213-241. 2008
42. Point/Counterpoint (Every Good Designer Uses Patterns). Peter Sommerlad, James Noble. *IEEE Software* 24(4) 2008. pp68-71.
43. Scoped types and aspects for real-time Java memory management. Chris Andreae, Yvonne Coady, Celina Gibbs, James Noble, Jan Vitek, Tian Zhao. *Real-Time Systems.* Vol 37 No 1, pp 1-44. Oct 2007.
44. Videogame Values: Human-Computer Interaction and Games. Barr, P., Biddle, R., and Noble, J. *Interacting with Computers: HCI Issues in Computer Games*, 19(2): pp180-195, 2007.
45. Featherweight Generic Confinement. Alex Potanin, James Noble, Dave Clarke, and Robert Biddle. *Journal of Functional Programming.* Vol 16 Issue 6, pp 793-811. Sep 2006.
46. Patterns as Signs: A Semiotics of Object-Oriented Design Patterns. James Noble, Robert Biddle, Ewan Tempero. *Systems, Signs, and Actions.* 2:1, pp3-40, 2006.
47. Exclusion and Potential Concurrency for Composite Objects. Abdelsalam Shanneb, John Potter and James Noble. *Science of Computer Programming.* 38:3, 344-365, Dec 2005
48. Scale-Free Geometry of Object-Oriented Programs. Alex Potanin, James Noble, Marcus Freen, Robert Biddle. *Communications of the ACM.* 48:5, 99-103, May 2005
49. Checking Ownership and Confinement. Alex Potanin and James Noble. *Concurrency and Computation, Practice and Experience* Volume 16, Issue 7, pp671 - 687, 2004.
50. Scalable Visualisations of Object-Oriented Systems using Ownership Trees. Trent Hill, James Noble, and John Potter. In the *Journal of Visual Languages and Computation*, vol 13 no 3 June

2002.

51. Aliasing in Object-Oriented Systems: Introduction to the Special Issue. James Noble and Doug Lea. *Software Practice and Experience*. 31:1, p.1565, 2001. Reprinted in *Software Focus*. Wiley. Jan 2001.
52. Arguments and Results. James Noble. *The Computer Journal*. vol 46 no 6. pp.439-450. 2001.
53. Object Oriented Program Visualisation in Tarraingim. James Noble, Lindsay Groves, and Robert Biddle. *Australian Computer Journal*, Special Issue on Software Visualisation. 27(4), November 1995.
54. Tarraingim— A Program Animation Environment. R. J. Noble and L. J. Groves. *New Zealand Journal of Computing*, 4(1), December 1992.

Refereed Conference Papers

Conferences marked ** are highly prestigious: e.g. accepting less than 1/5 of submitted papers (32 papers)

Papers marked # won best paper or best presentation awards. (7 papers)

55. Necessary Specifications are Necessary for Robustness. Julian Mackay, Sophia Drossopoulou, James Noble, Susan Eisenbach. Under Review. 2022.
56. Rusty Links in Local Chains. James Noble, Julian Mackay, Tobias Wrigstad. Under review. 2022.
57. More Programming Than Programming: Teaching Formal Methods in a Software Engineering Programme. James Noble. David Streader, Isaac Oscar Gariano, Miniruwani Samarakoon. To appear in the NASA Formal Methods Symposium (NFM) 2022.
58. Dala: a simple capability-based dynamic language design for data race freedom. Onward! 2021
59. Holistic Specifications for Robust Programs. Sophia Drossopoulou, James Noble, Julian Mackay, Susan Eisenbach FASE 2020.
60. Dalarna: A Simplistic Capability-Based Dynamic Language Design For Data Race Freedom. Kiko Fernandez-Reyes, James Noble, et al. FtFJP 2020.
61. Design Patterns for Angular Hotdraw. Kirita-Rose Escott, James Noble. EuroPLoP 2019.
62. Transient Typechecks are (Almost) Free. Richard Roberts, Stefan Marr, Michael Homer, James Noble. European Conference on Object-Oriented Programming. (ECOOP) 2019.
63. First-class Dynamic Types. Michael Homer, Timothy Jones, James Noble. Dynamic Language Symposium (DLS) 2019. 1-14.
64. Which of my transient type checks are not (almost) free? Isaac Oscar Gariano, Richard Roberts, Stefan Marr, Michael Homer, James Noble. VMIL 2019 58-66
65. CallE: an effect system for method calls. Isaac Oscar Gariano, James Noble, Marco Servetto. Onward! 2019: 32-45
66. Cognitive Dimensions of Modular Noise Improvisation James Noble. Psychology of Programming Interest Group (PPIG) 2019
67. Promoting Developer Security: Creating a Lightweight Program of Assurance Techniques. Charles Weir, Lynne Blair, Ingolf Becker, M. Angela Sasse, James Noble, Awais Rashid IEEE International Conference on Software Engineering Software Engineering in Practice (SEIP) 2019 41-50.
68. Agile Practices in Practice: Towards A Theory of Agile Adoption and Process Evolution. Brendan Julian, James Noble, and Craig Anslow. Agile Processes in Software Engineering and Extreme Programming (XP) 2019. 3-18.
55. What Programming Languages Do Developers Use? A Theory of Static vs Dynamic Language Choice. Aaron Pang, Craig Anslow, James Noble. ^[L]_[SEP]In Visual Languages and Human-Centric

Computing (VL/HCC) 2018: 239-247

56. Light-Touch Interventions to Improve Software Development Security. Charles Weir, Lynne Blair, Ingolf Becker, M. Angela Sasse, James Noble. IEEE Cybersecurity Development (SecDev) 2018: 85-93
57. **# Becoming Agile: A Grounded Theory of Agile Transitions in Practice. Rashina Hoda, James Noble. International Conference on Software Engineering (ICSE) 141-151. 2017.
58. The Left Hand of Equals. James Noble, Andrew P. Black, Kim B. Bruce, Michael Homer, Mark S. Miller. Onward! Essay, 2016
59. ** Object Inheritance Without Classes. Timothy Jones, Michael Homer, James Noble, Kim B. Bruce. European Conference on Object-Oriented Programming. (ECOOP) 2016.
60. ** Brand Objects for Nominal Typing. Timothy Jones, Michael Homer, James Noble. In European Conference on Object-Oriented Programming (ECOOP) 2015: 198-221
61. **# How Much Up-front? A Grounded Theory of Agile Architecture. Michael Waterman, James Noble, George Allan. International Conference on Software Engineering (ICSE) 2015.
62. From APIs to languages: Generalising method names. Michael Homer, Timothy Jones, James Noble. Dynamic Language Symposium. (DLS) 2015: 1-12
63. Coordinating Expertise Outside Agile Teams. Mawarny Md.Rejab, James Noble, Stuart Marshall. Conference on Agile Processes and eXtreme Programming in Software Engineering, (XP) 2015
64. How to Break the Bank: Semantics of Capability Policies. Sophia Drossopoulou and James Noble. In Integrated Formal Methods (iFM) Sept 2014.
65. Combining Tiled and Textual Views of Code. Michael Homer and James Noble. In 2nd IEEE Working Conference on Software Visualization (VISSOFT) Sept 2014.
66. Livecoding the SynthKit: littleBits as an embodied programming language. James Noble. New Ideas and Emerging Results, In 2nd IEEE Working Conference on Software Visualization (VISSOFT) Sept 2014.
67. ** Graceful Dialects. Michael Homer, Timothy Jones, James Noble, Kim B. Bruce, Andrew P. Black. In the Proceedings of the European Conference on Object-Oriented Programming (ECOOP) 131-156. 2014.
68. Distributing Expertise in Agile Software Development Projects. Md. Rejab, James Noble, George Allan. In the Proceedings of AGILE. July 2014.
69. Locating Expertise in Agile Software Development Projects. Md. Rejab, James Noble, George Allan. In the Conference on Agile Processes and eXtreme Programming in Software Engineering, (XP) pp260-268. 2014
70. ** Are Your Incoming Aliases Really Necessary? Counting the Cost of Object Ownership. Alex Potanin. Monique Damitio. James Noble. In the International Conference on Software Engineering (ICSE) pp. 743-751. 2013.
71. ** What Programmers do with Inheritance in Java? Ewan Tempero, Hong Yul Yang, James Noble. In the Proceedings of the European Conference on Object-Oriented Programming (ECOOP) 2013.
72. ** The Billion-Dollar Fix: Safe Modular Circular Initialisation with Placeholders and Placeholder Types. Marco Servetto, Julian Mackay, Alex Potanin, James Noble. In the Proceedings of the European Conference on Object-Oriented Programming (ECOOP) 2013.
73. SourceVis: Collaborative Software Visualization for Co-located Environments. Craig Anslow, Stuart Marshall, James Noble, Robert Biddle. In the IEEE International Working Conference on Software Visualization (VISSOFT), Eindhoven, The Netherlands, 2013.

74. A tile-based editor for a textual programming language. Michael Homer, James Noble. In the IEEE International Working Conference on Software Visualization (VISSOFT), pp1-4. Eindhoven, The Netherlands, 2013.
75. Seeking Grace: A New Object-Oriented Language for Novices. Kim Bruce, Andrew Black, Michael Homer, James Noble, Amy Ruskin, Richard Yallow. In the Technical Symposium on Computer Science Education (SIGCSE) pp129-134. 2013.
76. Designing Grace: Can an Introductory Programming Language Support the Teaching of Software Engineering? James Noble, Michael Homer, Kim B. Bruce, and Andrew P. Black. In the 26th Conference on Software Engineering Education and Training (CSEE&T). 2013.
77. The Effect of Complexity and Value on Architecture Planning in Agile Software Development. Michael Waterman, James Noble, George Allan. In Agile Processes in Software Engineering and Extreme Programming, 13th International Conference (XP). pp238-252. 2013.
78. Agile Software Development with Distributed Teams: Senior Management Support. Siva Dorairaj, James Noble, George Allan. pp197-205. (ICGSE) 2013.
79. Grace: the absence of (inessential) difficulty. Andrew P. Black. Kim B. Bruce. Michael Homer. James Noble. In Onward! 2012.
80. Patterns as Objects in Grace. Michael Homer, James Noble, Kim B. Bruce, Andrew P. Black, David J. Pearce. In Dynamic Language Symposium, (DLS) 2012.
81. Profiling Field Initialization for Java. Stephen Nelson, David J. Pearce and James Noble. In Runtime Verification (RV) 2012.
82. Graceful Patterns for Patterns in Grace. Michael Homer, James Noble. In Pattern Languages of Program Design (PLOP) 2012.
83. Knowledge Management in Distributed Agile Software Development, Siva Dorairaj, James Noble, Petra Malik. AGILE 2012, Dallas, Texas, USA, Aug 13-17, 2012.
84. Understanding Lack of Trust in Distributed Agile Teams: A Grounded Theory Study. Siva Dorairaj, James Noble, Petra Malik. Evaluation and Assessment in Software Engineering, 16th International Conference, EASE 2012, Ciudad Real, Spain, May 14-15, 2012.
85. Understanding Team Dynamics in Distributed Agile Software Development. Siva Dorairaj, James Noble, Petra Malik. Agile Processes in Software Engineering and Extreme Programming, 13th International Conference, XP 2012, Malmo, Sweden, May 21-25, 2012.
86. SourceVis: A Tool for Multi-touch Software Visualization. Craig Anslow, Stuart Marshall, James Noble, and Robert Biddle. In Proceedings of the International Conference on Interactive Tabletops and Surfaces (ITS), Kobe, Japan, Oct 2011.
87. Effective Communication in Distributed Agile Software Development Teams. Siva Dorairaj, Petra Malik, James Noble. In the Conference on Agile Processes and eXtreme Programming in Software Engineering, XP 2011, pp.102-166
88. Supporting Self-Organizing Agile Teams: What's Senior Management Got To Do With It. Rashina Hoda, Stuart Marshall, James Noble. In the Conference on Agile Processes and eXtreme Programming in Software Engineering, XP 2011.
89. Bridging Cultural Differences — A Grounded Theory. Siva Dorairaj, Petra Malik, James Noble. In the Indian Software Engineering Conference (ISEC) 2011 p3-10.
90. # Qualitas Corpus: A Curated Collection of Java Code for Empirical Studies. Ewan Tempero, Craig Anslow, Jens Dietrich, Ted Han, Jing Li, Markus Lumpe, Hayden Melton and James Noble. In the Asia Pacific Software Engineering Conference (APSEC2010), December 2010
91. ** Tribal Ownership. Nicholas Cameron, James Noble, Tobias Wrigstad. In the ACM Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA) 2010, p618-

633.

92. ** Agility in Context. Rashina Hoda, Philippe Kruchten, James Noble, Stuart Marshall. In the ACM Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA) 2010, p74-88.
93. ** Organizing Self-Organizing Teams. Rashina Hoda, James Noble, Stuart Marshall. International Conference on Software Engineering (ICSE) 2010, p285-294.
94. An empirical study of overriding in open source Java. Ewan D. Tempero, Steve Counsell, James Noble. In Australian Computer Science Conference (ACSC) 2011. Pp366-345.
95. Craig Anslow, Stuart Marshall, James Noble, Ewan D. Tempero, Robert Biddle: User Evaluation of Polymetric views using a large visualization wall. ACM Symposium on Software Visualisation (SOFTVIS) 2010, p25-34.
96. Haowei Ruan, Craig Anslow, Stuart Marshall, James Noble. Exploring the Inventor's Paradox: Applying Jigsaw to Software Visualization. ACM Symposium on Software Visualisation (SOFTVIS) 2010, p83-92.
97. Encoding Ownership Types in Java. Nicholas Cameron, James Noble. In Technology of Object-Oriented Languages and Systems (TOOLS 48) 2010, p271-290.
98. Understanding the Impact of Collection Contracts on Design. Stephen Nelson, David Pearce, James Noble. In Technology of Object-Oriented Languages and Systems (TOOLS 48) 2010, p61-78.
99. An Empirical Study of Overriding in Open Source Java. Ewan Tempero, Steve Counsell and James Noble. In Australasian Computer Science Conference (ACSC) 2010.
100. Agile Undercover: When Customers Don't Collaborate. Rashina Hoda, James Noble and Stuart Marshall. In the Conference on Agile Processes and eXtreme Programming in Software Engineering, XP 2010, 73-87.
101. What Language Does Agile Speak?. Rashina Hoda, James Noble and Stuart Marshall. To appear in the Conference on Agile Processes and eXtreme Programming in Software Engineering, XP 2010, p387-388.
102. Understanding the Importance of Trust in Distributed Agile Projects: A Practical Perspective. Siva Dorairaj, James Noble and Petra Malik. In the Conference on Agile Processes and eXtreme Programming in Software Engineering, XP 2010, p172-177.
103. The XP Customer Team: A Grounded Theory. Angela Martin, Robert Biddle, James Noble. In the Agile Development Conference (AGILE) 2009.
104. XP Customer Practices: A Grounded Theory. Angela Martin, Robert Biddle, James Noble. In the Agile Development Conference (AGILE) 2009.
105. ** The Myths of Object-Orientation. James Noble. In the Proceedings of the European Conference on Object-Oriented Programming (ECOOP) 2009
106. Negotiating Contracts for Agile Projects: A Practical Perspective. Rashina Hoda, James Noble and Stuart Marshall. In the Conference on Agile Processes and eXtreme Programming in Software Engineering, XP 2009.
107. Keith Cassell, Peter Andreae, Lindsay Groves, James Noble: Towards Automating Class-Splitting Using Betweenness Clustering. ASE 2009: 595-599
108. ** Multiple Dispatch in Practice. Radu Muschevici, Alex Potanin, Ewan Tempero, and James Noble. In the ACM Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA) 2008.
109. ** Caching and Incrementalization for the Java Query Language. Darren Willis, David J. Pearce, and James Noble. In the ACM Conference on Object-Oriented Programming, Systems,

Languages and Applications (OOPSLA) 2008.

110. **UpgradeJ: Incremental Typechecking for Class Upgrades. Gavin Bierman, Matthew Parkinson, James Noble. . In Proceedings of the European Conference on Object-Oriented Programming (ECOOP) 2008 pp235-295.
111. **How do Java Programs use Inheritance? Ewan Tempero. James Noble, Hayden Melton. In Proceedings of the European Conference on Object-Oriented Programming (ECOOP) 2008, pp667-691.
112. A Qualitative Study of Culture and Persuasion in a Smoking Cessation Game. Rilla Khaled, Ronald Fischer, James Noble, Robert Biddle. PERSUASIVE 2008. pp224-236.
113. Patterns for Program Query Optimisation. David J. Pearce, James Noble. To appear in the Proceedings of the Conference on Pattern Languages of Programs (PLOP) 2008.
114. Exception Handling Bug Patterns in Aspect Oriented Programs. Roberta Coelho, Awais Rashid, Uira Kulesza, Arndt von Staa, Carlos Lucena, James Noble To appear in the Proceedings of the Conference on Pattern Languages of Programs (PLOP) 2008.
115. **Multiple Ownership. Nicholas Cameron, Sophia Drossopoulou,, James Noble, Matthew Smith. In ACM Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA) 2007. pp441-460.
116. **MAO; Ownership and Effects for more Efficient Reasoning about Aspects. Curtis Clifton, Gary T. Leavens, James Noble. In Proceedings of the European Conference on Object-Oriented Programming (ECOOP).2007, pp451-475.
117. Fine Tuning the Persuasion in Persuasive Games. Rilla Khaled, Pippin Barr, James Noble, Ronald Fischer, Robert Biddle . PERSUASIVE 2007. pp36-47.
118. Agile Development Iterations and UI Design. Jennifer Ferreira, James Noble, Robert Biddle. Agile 2007, pp50.58.
119. Up-Front Interaction Design in Agile Development. Jennifer Ferreira, James Noble, Robert Biddle. XP 2007. pp9-16
120. Dynamic Ownership in a Dynamic Language. Donald Gordon and James Noble. Dynamic Languages Symposium (DLS2007), pp9-16.
121. Tribe: a simple virtual class calculus . Dave Clarke, Sophia Drossopoulou, James Noble, Tobias Wrigstad. AOSD 2007. pp121-134
122. Problem Frame Patterns. James Noble, Rebecca Wirfs-Brock, Paul Taylor. In the Proceedings of the Conference on Pattern Languages of Programs (PLOP) 2006.
123. Programmers are from Mars: Customers are from Venus: A practical guide to working with customers on XP Projects. Angela Martin, James Noble, Robert Biddle. In the Proceedings of the Conference on Pattern Languages of Programs (PLOP) 2006.
124. ** Generic Ownership for Generic Java. Alex Potanin, James Noble, Dave Clarke, Robert Biddle. ACM Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA). Portland, 2006. pp311-324.
125. ** A Framework for Implementing Pluggable Type Systems. Chris Andraea, James Noble, Shane Markstrum, Todd Millstein. ACM Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA). Portland, 2006. pp57-74.
126. ** Understanding the Shape of Java Software. Gareth Baxter, Marcus Frean, Hayden Melton, James Noble, Mark Rickerby, Hayden Smith, Ewan Tempero, Matt Visser. . ACM Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA). Portland, 2006. pp397-412.
127. Changing the Virtual Self: The Avatar Transformation Activity in Popular Games. Barr, P.,

- Brown, J., Biddle, R., Khaled, R., and Noble, J, Joint International Conference on CyberGames and Interactive Entertainment, 2006.
128. Patterns for Relationship Aspects. David Pearce and James Noble. In Proceedings of European Conference on Pattern Languages of Programs. Irsee, Germany, 2006
 129. ** STARS: Scoped Types and Aspects for Real-Time Systems. Chris Andreae, Yvonne Coady, Celina Gibbs, James Noble, Jan Vitek, Tian Zhao. In Proceedings of the European Conference on Object-Oriented Programming (ECOOP). Nantes, 2006.
 130. ** Efficient Object Querying for Java. Darren Willis, David Pearce. James Noble. In Proceedings of the European Conference on Object-Oriented Programming (ECOOP). Nantes, 2006.
 131. ** Relationship Aspects. David Peace and James Noble. In Proceedings of Aspect-Oriented Software Development (AOSD). Bonn, 2006.
 132. Investigating Social Software as Persuasive Technology, Rilla Khaled, Pippin Barr, James Noble, Robert Biddle, in Proceedings of First International Conference on Persuasive Technology for Human Well-Being (Persuasive06), Eindhoven, Netherlands, 2006.
 133. Our Place or Mine?: Exploration into Collectivism-Focused Persuasive Technology Design, Rilla Khaled, Pippin Barr, James Noble, Robert Biddle, in Proceedings of First International Conference on Persuasive Technology for Human Well-Being (Persuasive06), Eindhoven, Netherlands, 2006.
 134. Postmodern Prospects for Conceptual Modelling. James Noble and Robert Biddle. Invited Presentation in Proceedings of the Asian-Pacific Conference on Conceptual Modelling (APCCM). Hobart. pp11-20. 2006
 135. From Pushing Buttons to Play and Progress: Value and Interaction in Fable. Pippin Barr ,James Noble, Robert Biddle, Rilla Khaled. In Proc. Seventh Australasian User Interface Conference (AUIC), Hobart, Australia. CRPIT, 50. Piekarski, W., Ed. ACS. 61-68. 2006.
 136. A Case for Iconic Icons. Jennifer Ferreira, James Noble, Robert Biddle. In Proc. Seventh Australasian User Interface Conference (AUIC), Hobart, Australia. CRPIT, 50. Piekarski, W., Ed. ACS. 97-100. 2006
 137. Persuasive Interaction for Collectivist Cultures. Rilla Khaled, Robert Biddle, James Noble, Pippin Barr, Ron Fischer. In Proc. Seventh Australasian User Interface Conference (AUIC), Hobart, Australia. CRPIT, 50. Piekarski, W., Ed. ACS. 73-80. 2006
 138. **Efficient Multimethods in a Single-Dispatch Language. Brian Foote, Ralph Johnson, James Noble. In Proceedings of the European Conference on Object Oriented Programming (ECOOP). July 2005.
 139. A Taxonomy of Sequencer User-Interfaces, Matthew Duignan, James Noble, Robert Biddle. In Proceedings of the International Computer Music Conference (ICMC). 2005.
 140. Oppositional Play: Gathering Negative Evidence for Computer Game Values. Barr, P., Marsen, S., Noble, J., and Biddle, R. In Proceedings of the Second Australasian Conference on Interactive Entertainment (IE2005), 2005.
 141. Amethodology. Charles Weir, James Noble. In Proceedings of the Tenth European Conference on Pattern Languages of Programs. Irsee, Germany, 2005.
 142. Clicki: A framework for light-weight web-based visual applications. Donald Gordon, James Noble, Robert Biddle. In Proceedings of Sixth Australasian User Interface Conference (AUIC). Newcastle, Australia, January 2005.
 143. The Semiotics of User Interface Redesign. Jennifer Ferreira, Pippin Barr, James Noble. In Proceedings of Sixth Australasian User Interface Conference (AUIC). Newcastle, Australia, January 2005.
 144. A Practical Application of a Taxonomy of User-Interface Metaphor. Pippin Barr, Rilla Khaled,

- James Noble, Robert Biddle. In Proceedings of Sixth Australasian User Interface Conference (AUIC). Newcastle, Australia, January 2005.
145. Using a System of Tutorials and Groups to Teach User Interface Design. Wright, T., Noble, J. and Marshall, S. In Proc. Seventh Australasian Computing Education Conference (ACE2005), Newcastle, Australia. CRPIT, 42. Young, A. and Tolhurst, D., Eds., ACS. p187-19. January 2005.
 146. Scoped Types for Real-Time Java. Tian Zhao, James Noble, and Jan Vitek. In The 25th IEEE Real-Time Systems Symposium (RTSS 2004). Lisbon, Portugal, December 2004.
 147. Notes on Notes on Postmodern Programming. James Noble and Robert Biddle. In Proceedings of the Onward! stream at the Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA). Vancouver, October, 2004.
 148. My Friend the Customer. Charles Weir, James Noble, Angela Martin, Robert Biddle. In Proceedings of the Ninth European Conference on Pattern Languages of Programs. Irsee, Germany, 2004.
 149. Metaphorscape. Patterns for XP System Metaphors. Rilla Khaled, Pippin Barr, James Noble, Robert Biddle. In Proceedings of the Ninth European Conference on Pattern Languages of Programs. Irsee, Germany, 2004.
 150. Metaphors for Electronic Music Production in Reason and Live. Matthew Duignan, Pippin Barr, Robert Biddle. In Proceedings of the 6th Asian Pacific Conference on Computer Human Interaction (APCHI 2004), pages 111-120. IEEE Computer Society. Springer-Verlag. 2004.
 151. Towards a General Model for Assisting Navigation. Mike McGavin, James Noble, Robert Biddle and Judy Brown. In Proceedings of the 6th Asian Pacific Conference on Computer Human Interaction (APCHI 2004), pages 262-271. IEEE Computer Society. Springer-Verlag. 2004.
 152. Snapshot Query-Based Debugging. Alex Potanin, James Noble, Robert Biddle. In the Australasian Software Engineering Conference (ASWEC 2004). Melbourne, Australia. IEEE CS Press. April 2004.
 153. XML Database Support for Program Trace Visualisation. Craig Anslow, Stuart Marshall, Robert Biddle, James Noble, and Kirk Jackson. Australasian Symposium on Information Visualisation (INVIS 2004). Neville Churcher and Clare Churcher (eds). January 2004.
 154. Delegation Diagrams: Visual Support for the Development of Object-Oriented Designs. Ewan Tempero, James Noble, Robert Biddle. In Proc. Fifth Australasian User Interface Conference (AUIC2004), Dunedin, New Zealand. CRPIT, 28. Cockburn, A., Ed. ACS. 83-89. 2004.
 155. A Web User Interface For An Interactive Software Repository. Stuart Marshall and Robert Biddle and James Noble. In Proc. Fifth Australasian User Interface Conference (AUIC2004), Dunedin, New Zealand. CRPIT, 28. Cockburn, A., Ed. ACS. 57-64. 2004.
 156. Less Extreme Programming. James Noble, Stuart Marshall, Stephen Marshall, Robert Biddle. In Proc. Sixth Australasian Computing Education Conference (ACE2004), Dunedin, New Zealand. CRPIT, 30. Lister, R. and Young, A. L., Eds., ACS. 217-226. 2004
 157. ** Saving the World from Bad Beans: Deployment-time Confinement Checking. Dave Clarke, Michael Richmond, James Noble. To Appear in the ACM Conference on Object-oriented Programming, Languages, Systems and Applications (OOPSLA), October, 2003. Anaheim.
 158. No Name: Notes on Reuse. Robert Biddle, Angela Martin, James Noble and Ewan Tempero. To Appear In Proceedings of the Onward! stream at the Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA). Anaheim, October, 2003.
 159. Use Case Analysis with Narrative Semiotics. Sky Marsen, Robert Biddle, James Noble. In Proceedings of the Australian Conference on Information Systems (ACIS), Perth, 2003.
 160. A technology for lightweight web-based visual applications. Donald Gordon, Robert Biddle, James Noble, and Ewan Tempero. To Appear in Margaret Burnett and John Grundy, editors,

- IEEE Symposium on Visual and Multimedia Software Engineering, Auckland, New Zealand, 2003.
161. # The Hitch-Hiker's Guide to Google. Charles Weir and James Noble. In Proceedings of the Eighth European Conference on Pattern Languages of Program Design (EuroPLOP). Irsee. Universitäts Verlag Konstanz. 2003.
 162. A Window in Your Pocket. Charles Weir, James Noble. In Proceedings of the Eighth European Conference on Pattern Languages of Program Design (EuroPLOP). Irsee. Universitäts Verlag Konstanz. 2003.
 163. Interface Ontology: Creating a World for Computer Interfaces. Pippin Barr, Robert Biddle, James Noble. In Proceedings of the Eighth European Conference on Pattern Languages of Programs. Irsee, Germany. 2003
 164. Software Visualisation of Java Programs in InspectJ. Rilla Khaled, James Noble, Robert Biddle. In Proceedings of the Eighth European Conference on Pattern Languages of Programs. Irsee, Germany. 2003
 165. Being Jane Malkovich: a Look into the World of an XP Customer. Angela Martin, James Noble, Robert Biddle. In Proceedings of the Fourth International Conference on eXtreme Programming and Agile Processes in Software Engineering, Lecture Notes in Computer Science, Springer-Verlag. 2003
 166. InspectJ: Program monitoring for visualisation using AspectJ. Rilla Khaled, James Noble, and Robert Biddle. In Michael Oudshoorn, editor, Proceedings of the 26th Australasian Computer Science Conference, Conferences in Research and Practice in Information Technology, Vol 16, Adelaide, South Australia, Australian Computer Society 2003.
 167. Spreadsheet structure inspection using low level access and visualization. Daniel Ballinger, Robert Biddle, and James Noble. In Robert Biddle and Bruce Thomas, editors, Proceedings of the 4th Australasian User Interface Conference, Conferences in Research and Practice in Information Technology, Vol 18, Adelaide, South Australia. Australian Computer Society. 2003.
 168. Icons R icons: User interface icons, metaphor and metonymy. Pippin Barr, Robert Biddle, and James Noble. In Robert Biddle and Bruce Thomas, editors, Proceedings of the 4th Australasian User Interface Conference, Conferences in Research and Practice in Information Technology, Vol 18, Adelaide, South Australia. Australian Computer Society 2003.
 169. A lightweight web based case tool for UML class diagrams. Daniel Mackay, Robert Biddle, and James Noble. In Robert Biddle and Bruce Thomas, editors, Proceedings of the 4th Australasian User Interface Conference, Conferences in Research and Practice in Information Technology, Vol 18, Adelaide, South Australia, Australian Computer Society 2003.
 170. Spreadsheet visualisation to improve end-user understanding. Daniel Ballinger, Robert Biddle, and James Noble. In Tim Pattison and Bruce Thomas, editors, Proceedings of the Australasian Symposium on Information Visualisation, Conferences in Research and Practice in Information Technology, Vol 24, Adelaide, South Australia, Australian Computer Society 2003.
 171. A case study of visualisation using AspectJ. Rilla Khaled, James Noble, and Robert Biddle. In Tim Pattison and Bruce Thomas, editors, Proceedings of the Australasian Symposium on Information Visualisation, Conferences in Research and Practice in Information Technology, Vol 24, Adelaide, South Australia, Australian Computer Society 2003.
 172. Teaching the evaluation of object-oriented designs. Robert Biddle, James Noble, and Ewan Tempero. In Tony Greening and Raymond Lister, editors, Proceedings of the 5th Australasian Computing Education Conference, Conferences in Research and Practice in Information Technology, Vol 20, Adelaide, South Australian Computer Society 2003.
 173. A Lightweight Case Tool for UML Sequence Diagrams. Rilla Khaled, Dan Mackay, Robert Biddle, James Noble, Ewan Tempero. SIGCHINZ Workshop on Human-Computer Interaction. 2002.

174. A Taxonomy of User-Interface Metaphors. Pippin Barr, Robert Biddle, James Noble. SIGCHINZ Workshop on Human-Computer Interaction. 2002.
175. # Notes on Postmodern Programming. James Noble and Robert Biddle. In Proceedings of the Onward! stream at the Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA). Seattle, October, 2002.
176. Patterns for Self. James Noble and Brian Foote. To appear in Proceedings of the Third Asian-Pacific Conference on Pattern Languages of Programs (KoalaPLoP). James Noble and Paul Taylor, Editors. Australian Computer Society, 2002.
177. Essential Use Case Bodies. Robert Biddle, James Noble, Ewan Tempero. To appear in Proceedings of the Third Asian-Pacific Conference on Pattern Languages of Programs (KoalaPLoP). James Noble and Paul Taylor, Editors. Australian Computer Society, 2002.
178. ** Patterns as Signs. James Noble, Robert Biddle. In Proceedings of the European Conference on Object Oriented Programming, Spain. (ECOOP). pp368-391. Springer-Verlag, 2002.
179. Supporting Reusable Use Cases. Robert Biddle, James Noble, Ewan Tempero. In Proceedings of the International Conference on Software Reuse (ICSR). Austin. pp 210-226 Springer-Verlag 2002
180. Sokoban: A System Object Case Study. Robert Biddle, James Noble, Ewan Tempero. In Proceedings of Technology of Object-Oriented Languages and Systems (TOOLS 37) Sydney. pp103-114. Australian Computer Society. 2002.
181. Reflections on CRC Cards and OO Design. Robert Biddle, James Noble, Ewan Tempero. In Proceedings of Technology of Object-Oriented Languages and Systems (TOOLS 37) Sydney. pp201-206. Australian Computer Society. 2002.
182. Metaphor and Metonymy in Object-Oriented Design Patterns. James Noble, Robert Biddle and Ewan Tempero. In Proceedings of the Australian Computer Science Conference (ACSC), Melbourne Australia, pp187-196. 2002.
183. Essential Use Cases and Responsibility in Object-Oriented Development. Robert Biddle, James Noble, and Ewan Tempero. In Proceedings of the Australian Computer Science Conference (ACSC), Melbourne Australia, pp 7-16. 2002.
184. Program Visualisation for Visual Programs. James Noble and Robert Biddle. In Proceedings of the Australian User Interfaces Conference (AUIIC), Melbourne, Australia, pp 29-38. 2002.
185. Visualising 1,051 Visual Programs: Module Choice and Layout In Nord Modular Patch Language. James Noble and Robert Biddle. In Proceedings of INVIS 2001, Sydney, Australia, pp 121-128. 2001.
186. Role-play and Use Case Cards for Requirements Review. Robert Biddle, James Noble, Ewan Tempero. In Proceedings of the Australian Conference on Information Systems (ACIS), Coffs Harbour, Australia, 2001.
187. ** Simple Ownership Types for Object Containment. David Clarke, James Noble, John Potter. In Proceedings of the European Conference on Object Oriented Programming, Brussels. (ECOOP) pp 2-27. Springer-Verlag, 2001.
188. ** Capabilities for Sharing: A Generalization of Uniqueness and Read-Only. John Boyland, James Noble, William Retert. In Proceedings of the European Conference on Object Oriented Programming, Brussels. (ECOOP) pp 53-76 Springer-Verlag, 2001.
189. Need to Know: Patterns for Coupling Contexts and Components. James Noble. In Proceedings of the Second Australasian Conference on Pattern Languages for Program Design (KoalaPLoP'01). pp 181-195. 2001
190. Patterns for Essential Use Cases. Robert Biddle, James Noble, Ewan Tempero. In Proceedings of the Second Australasian Conference on Pattern Languages for Program Design (KoalaPLoP'01)

pp 196-206. 2001.

191. Reflections on Remote Reflection. Michael Richmond and James Noble. In Proceedings of the Australian Computer Science Conference (ACSC). Gold Coast, January 2001. IEEE Computer Society Press.
192. ** Exclusion for Composite Objects. James Noble, David Holmes, John Potter. In Proceedings of the Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA) Minneapolis. pp 13-28. ACM Press, 2000.
193. Scalable Visualisations with Ownership Trees. Trent Hill, James Noble, John Potter. In Proceedings of Technology of Object-Oriented Languages and Systems (TOOLS 37) Sydney. IEEE Computer Society Press, 2000.
194. # Visualising the Structure of Object-Oriented Systems. Trent Hill, James Noble, John Potter. In Proceedings of the IEEE Symposium on Visual Languages (VL). Seattle. pp 191-198 IEEE Computer Society Press, 2000.
195. Small User Interfaces. James Noble and Charles Weir. In Proceedings of the European Conference on Pattern Languages of Program Design, Irsee. pp 327-343. Universitäts Verlag Konstanz 2000.
196. Process Patterns for Small Systems. Charles Weir and James Noble. In Proceedings of the European Conference on Pattern Languages of Program Design. Irsee. pp 344-356. Universitäts Verlag Konstanz. 2000.
197. Iterators and Encapsulation. James Noble. In Proceedings of Technology of Object-Oriented Languages and Systems (TOOLS 33) Mont-St-Michel. pp 431-442. IEEE Computer Society Press, 2000.
198. Natural Creation. James Noble. In Proceedings of KoalaPLOP 2000, The First Asian Pacific Conference on Patterns Languages of Programs. Melbourne, May 2000.
199. Small Interfaces. James Noble and Charles Weir. In Proceedings of KoalaPLOP 2000, The First Asian Pacific Conference on Patterns Languages of Programs. Melbourne, May 2000.
200. Conglomeration: Realising Aliasing Protection. John Potter and James Noble. In Proceedings of the Australian Computer Science Conference (ACSC). Canberra, January 2000. IEEE Computer Society Press.
201. Natural Creation: A Composite Pattern for Creating Objects. James Noble. In Proceedings of Technology of Object-Oriented Languages and Systems (TOOLS 32) Melbourne, pp 78-88. IEEE Computer Society Press, 1999.
202. Object Ownership for Dynamic Alias Protection. In Proceedings of Technology of Object-Oriented Languages and Systems (TOOLS 32) Melbourne, pp 176-187. IEEE Computer Society Press, 1999.
203. Using Secondary Storage. James Noble and Charles Weir. In Proceedings of the Fourth European Conference on Pattern Languages of Programming and Computing. Irsee, Germany, 1999.
204. Process Patterns for Personal Practice. Charles Weir and James Noble. In Proceedings of the Fourth European Conference on Pattern Languages of Programming and Computing. Irsee, Germany, 1999.
205. ** Flexible Alias Protection. James Noble, Jan Vitek, John Potter. In Proceedings of the European Conference on Object Oriented Programming, Brussels. (ECOOP) Springer-Verlag, 1998.
206. ** # Ownership Types for Flexible Alias Protection. David Clarke, John Potter, James Noble. In Proceedings of the Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA) Vancouver. ACM Press, 1998.

207. Objects and Constraints. James Noble. In Proceedings of Technology of Object-Oriented Languages and Systems (TOOLS Pacific), Melbourne, IEEE Computer Society Press, 1998.
208. Towards a Pattern Language for Object-Oriented Design. James Noble. In Proceedings of Technology of Object-Oriented Languages and Systems (TOOLS Pacific), Melbourne, IEEE Computer Society Press, 1998.
209. Integrating Metric Visualisation into a User-Interface Builder. James Noble. In Proceedings of OzCHI'98.
210. # The Ins and Outs of Objects. John Potter, James Noble, David Clarke. In Proceedings of the Australian Software Engineering Conference (ASWEC), Adelaide, IEEE Computer Society Press, 1998.
211. Classifying Relationships between Object-Oriented Design Patterns. James Noble. In Proceedings of the Australian Software Engineering Conference (ASWEC), Adelaide, IEEE Computer Society Press, 1998.
212. A Mode System for Flexible Alias Protection. John Potter, David Clarke, James Noble. In Proceedings of Formal Methods Pacific, Canberra, 1998.
213. Patterns for Small Machines. James Noble and Charles Weir. In Proceedings of the European Conference on Pattern Languages of Program Design, Irsee, Germany 1998. Universitäts Verlag Konstanz.
214. The Object System Pattern. James Noble. In Proceedings of the European Conference on Pattern Languages of Program Design, Irsee, Germany 1998. Universitäts Verlag Konstanz.
215. The Evolution of Tarraingím. James Noble, Robert Biddle, and Lindsay Groves. In Proceedings of the Australian Computer Science Conference (ACSC), Perth. Chris McDonald (ed). Springer-Verlag, 1998. pp 307-318.
216. Prototype-Based Programming for Abstract Program Visualisation. James Noble, In Proceedings of the Australian Software Visualisation Workshop, Adelaide. Chris Marlin (ed) Flinders University. ISBN 0-7258-0630-3. 1997.
217. Aspects of Synchronisation. David Holmes, James Noble, John Potter. In Proceedings of Technology of Object-Oriented Languages and Systems (TOOLS Pacific 25) Pacific, Melbourne, 1997. pp 7-18. IEEE Computer Society Press.
218. Patterns for Finding Objects within Designs. James Noble. In Proceedings of Technology of Object-Oriented Languages and Systems (TOOLS Pacific 25) Pacific, Melbourne, 1997. pp 103-110. IEEE Computer Society Press.
219. Change Detection for Aggregate Objects with Aliasing. James Noble and John Potter. In Proceedings of the Australian Software Engineering Conference (ASWEC), Sydney, 1997. IEEE Computer Society Press.
220. Visualising Alignment with Adaptive Gridlines. James Noble. (short paper). In Proceedings of the 13th IEEE Symposium on Visual Languages, 1997.
221. Arguments and Results. James Noble. In Proceedings of Conference on Pattern Languages of Program Design. Monticello, Illinois. 1997.
222. GOF Patterns for GUI Design. James Noble. In Proceedings of the European Conference on Pattern Languages of Program Design. Irsee, Germany. 1997.
223. Basic Relationship Patterns. James Noble. In Proceedings of the European Conference on Pattern Languages of Program Design. Irsee, Germany. 1997.
224. Interactive Design Metric Visualisation: Visual Metric Support for User Interface Design. James Noble and Larry L. Constantine. In Proceedings of OzCHI'96, Hamilton, NZ. 1996.
225. Scribble: A Diagram Editor with a Minimal Interface. James Noble. In Proceedings of

OzCHI'96, Hamilton, NZ. 1996.

226. Some Patterns for Relationships. James Noble. In Proceedings of Technology of Object-Oriented Languages and Systems (TOOLS Pacific 21), Melbourne, 1996. Prentice-Hall.
227. Found Objects: A Small Pattern Language for Finding Objects in Designs. James Noble. In Proceedings of the European Conference on Pattern Languages of Program Design. Irsee, Germany. 1996.
228. Explicit Relationships in Object Oriented Development. James Noble and John Grundy. In Proceedings of Technology of Object-Oriented Languages and Systems (TOOLS Pacific 18), Melbourne, 1995.
229. Tarraingím— A Program Animation Environment. R. James Noble and Lindsay J. Groves. In Proceedings of the 12th New Zealand Computer Conference, Dunedin, NZ. 1991.
230. An Introduction to the Tarraingím Program Animation Environment. R. James Noble and Lindsay J. Groves. In Proceedings of Technology of Object-Oriented Languages and Systems (TOOLS Pacific 6), Sydney, 1991.

Refereed Workshop Papers And Reports

231. Early Report: How to Improve Programmers' Expertise at App Security? Charles Weir, Awais Rashid, James Noble. IMPS Workshop at ESSoS 2016
232. Hacking with multi-touch for Java (MT4j). Craig Anslow, Stuart Marshall, James Noble, Robert Biddle. Mobile! Workshop at SPLASH 2016.
233. Reaching the masses: a new subdiscipline of app programmer education. Charles Weir, Awais Rashid, James Noble. SIGSOFT FSE 2016.
234. How to Improve the Security Skills of Mobile App Developers? Comparing and Contrasting Expert Views. Charles Weir, Awais Rashid, James Noble. WSI workshop at SOUPS 2016
235. Permission and Authority Revisited towards a formalisation. Sophia Drossopoulou, James Noble, Mark S. Miller, Toby C. Murray. In the Proceedings of the Workshop on Formal Techniques for Java-like Programs (FTfJP) at the European Conference for Object-Oriented Programming, 2016.
236. Abstract Data Types in Object-Capability Systems. James Noble, Sophia Drossopoulou, Mark Miller, Toby Murray, Alex Potanin. In the Proceedings of the International Workshop on Aliasing, Capabilities and Ownership (IWACO) at the European Conference for Object-Oriented Programming, 2016.
237. Beyond Types: Extending the Gradual Guarantee. James Noble, Michael Homer, Timothy Jones, Sophia Drossopoulou, Andrew Black, Kim Bruce. In the Script to Program Evolution Workshop (STOP) at the European Conference for Object-Oriented Programming, 2016.
238. Operators and precedence in programming languages. Najwani Razali, James Noble, Stuart Marshall. In the proceedings of PLATEAU 2015 workshop at OOPSLA 2015.
239. [Demo Abstract] littleBits Synth Kit as a physically-embodied, domain specific functional programming language. James Noble, Timothy Jones. Refereed Demonstration to the 2nd ACM SIGPLAN Workshop on Functional Art, Music, Modelling and Design (FARM2014). Sep 2014.
240. On Owners as Accessors. James Noble and Alex Potanin. In the International Workshop on Aliasing, Capabilities, and Ownership (IWACO) at the European Conference for Object-Oriented Programming. July 2014.
241. Dynamic Semantics of Sheep cloning: Proving Cloning. Paley Li, Nicholas Cameron and James Noble. In the International Workshop on Aliasing, Capabilities, and Ownership (IWACO)

at the European Conference for Object-Oriented Programming. July 2014.

242. Tinygrace: A Simple Structurally Typed Language. Timothy Jones, James Noble. In the Proceedings of the Workshop on Formal Techniques for Java-like Programs (FTfJP) at the European Conference for Object-Oriented Programming, 2014.
243. A Rational Reconstruction of the Escrow Example. In the Proceedings of the Workshop on Formal Techniques for Java-like Programs (FTfJP) at the European Conference for Object-Oriented Programming, 2014.
244. Exploring Collaborative Software Visualization with Multi-touch Tables. Craig Anslow, Stuart Marshall, James Noble, Robert Biddle. In Proceedings of the Workshop on Collaboration meets Interactive Surfaces: Walls, Tables, Tablets, and Phones at the at the ACM International Conference on Interactive Tabletops and Surfaces (ITS), St Andrews, Scotland, 2013.
245. The Need for Capability Policies. Sophia Drossopoulou and James Noble. In the Proceedings of the Workshop on Formal Techniques for Java-like Programs (FTfJP) at the European Conference for Object-Oriented Programming, 2013.
246. Modules as Gradually-Typed Objects. Michael Homer, James Noble, Kim Bruce and Andrew Black. In the Proceedings of the 7th Workshop on Dynamic Languages and Applications (DYLA) at the European Conference for Object-Oriented Programming, 2013.
247. Sheep Cloning with Ownership Types . Paley Li, Nicholas Cameron and James Noble, in Foundations of Object-Oriented Programming (FOOL) at the ACM Conference on Systems, Programming, Languages, Applications, for Humanity (SPLASH) 2012.
248. Interactive Multi-touch Surfaces for Software Visualization. Craig Anslow, Stuart Marshall, James Noble, and Robert Biddle. In Proceedings of the Workshop on Data Exploration for Interactive Surfaces (DEXIS) at the ACM International Conference on Interactive Tabletops and Surfaces (ITS), Kobe, Japan, 2011.
249. Cloning in Ownership. Paley Li, Nicholas R. Cameron, James Noble. in the Companion to the Proceedings of at the ACM Conference on Systems, Programming, Languages, Applications, for Humanity (SPLASH) p63-66, 2011.
250. Balancing Acts: Walking the Agile Tightrope. Rashina Hoda, James Noble, Stuart Marshall. In the Proceedings of the Workshop on Cooperative and Human Aspects of Software Engineering (CHASE) at ICSE2010.
251. Does Size Matter? A Preliminary Investigation of the Consequences of Powerlaws in Software. Joshua Lindsay, James Noble, Ewan Tempero. In the Proceedings of the Workshop on Emerging Trends in Software Metrics at ICSE 2010.
252. Understanding the Importance of Trust in Distributed Agile Projects: A Practical Perspective. Siva Dorairaj, James Noble and Petra Malik. To Appear in the Conference on Agile Processes and eXtreme Programming in Software Engineering, XP 2010.
253. OGJ gone Wild. Nicholas Cameron and James Noble. To be Presented at the International Workshop on Aliasing, Confinement, and Ownership in Object-Oriented Programming (IWACO) at ECOOP 2009.
254. From Java to UpgradeJ: An empirical study. Ewan Tempero, Gavin Bierman, James Noble, Matthew Parkinson. In the Proceedings of the First ACM Workshop on Hot Topics in Software Upgrades (HotSWUp), 2008.
255. Visualizing the word structure of Java class names. Craig Anslow, Hames Noble, Stuart Marshall, Ewan D. Tempero. Poster presentation in the OOPSLA Companion, 2008. pp777-778.
256. Relationships and Associations in Object-Oriented Languages. Stephen Nelson, Stephanie Balzer, Gavin M. Bierman, Erik Meijer, James Noble, David Pearce, Jiri Soukup, Frank Tip. Workshop abstract in the OOPSLA Companion, 2008. 855-856.

257. A for Agile, Issues with Awareness and Adoption. Rashina Hoda, James Noble, and Stuart Marshall. In the Proceedings of the Research-In-Progress Track at Agile 2008.
258. First Class Relationships for OO Languages. Stephen Nelson, David J. Pearce, James Noble. ECOOP 2008 Workshop on Multiparadigm Programming with Object-Oriented Languages (MPOOL) 2008.
259. Object State Querying for Optimisation. David J. Pearce, James Noble. ECOOP 2008 Workshop on Multiparadigm Programming with Object-Oriented Languages (MPOOL) 2009.
260. Web Software Visualization Using Extensible 3D (X3D) Graphics. Craig Anslow, James Noble, Stuart Marshall, and Robert Biddle. Poster, SoftVis, 2008.
261. Towards End-User Web Software Visualization. Craig Anslow, James Noble, Stuart Marshall, and Ewan Tempero. Graduate Consortium, VLHCC, 2008.
262. Towards Unifying Immutability and Ownership. Paley Li, Alex Potanin, James Noble, and Lindsay Groves. International Workshop on Aliasing, Confinement, and Ownership (IWACO), ECOOP 2008.
263. Introducing Alloy in a Software Modelling Course. James Noble, David J. Pearce, Lindsay Groves. ETAPS 2008 Workshop on Formal Methods in Computer Science Education (FORMED) 2008. pp81-91.
264. Aliasing, Confinement, and Ownership in Object-Oriented Programming Dave Clarke, Sophia Drossopoulou, James Noble, Tobias Wrigstad:. ECOOP Workshops 2007: 40-49
265. A High Integrity Profile for Memory Safe Programming in Real-time Java. Alex Potanin, James Noble, Tian Zhao, Jan Vitek. In Proceedings of the 3rd Workshop on Java Technologies for Real-time and Embedded Systems, San Diego, CA, USA, October 2005.
266. Querying in Java. Darren Willis, David J. Pearce, James Noble. Presented at the OOPSLA Workshop on Multiparadigm Programming (MPOOL2005), San Diego, October 2005.
267. An Analysis of Persuasive Technology Tool Strategies. Khaled, R., Noble, J., and Biddle, R. In Proceedings of the International Workshop on the Internationalisation of Products and Systems (IWIPS) 2005, Amsterdam, Netherlands.
268. Featherweight Generic Ownership. Alex Potanin, James Noble, Dave Clarke, Robert Biddle. In proceedings of the workshop on Formal Techniques for Java-like Programs (FTfJP) in European Conference for Object-Oriented Programming, Glasgow, Scotland, July 2005.
269. Defaulting Generic Java to Ownership. Alex Potanin, James Noble, Dave Clarke, Robert Biddle. In proceedings of the workshop on Formal Techniques for Java-like Programs (FTfJP) in European Conference for Object-Oriented Programming, Oslo, Norway, June 2004.
270. Featherweight Generic Confinement. Alex Potanin, James Noble, Dave Clarke, and Robert Biddle. In proceedings of the International Workshop on Foundations of Object-Oriented Languages (FOOL), affiliated with Symposium on Principles of Programming Languages, Venice, Italy, January 2004.
271. Towards a Model of Encapsulation. James Noble, Robert Biddle, Ewan Tempero. Alex Potanin, David Clarke. Presented at the ECOOP 2003 International Workshop on Aliasing and Confinement (IWACO). 2003.
272. Larry Constantine, Robert Biddle, and James Noble. Usage-centered design and software engineering: Models for integration. In IFIP Working Group 2.7/13.4, editor, ICSE 2003 Workshop on Bridging the Gap Between Software Engineering and Human-Computer Interaction, Portland, Oregon, 2003
273. Angela Martin, Brenda Chawner, Robert Biddle, and James Noble. Requirements with XP and the open source customer. In Brian Fitzgerald and David Parnas, editors, Proceedings of the Workshop on Free and Open Source Software, Genoa, Italy, 2003

274. Checking Ownership and Confinement Properties. Alex Potanin and James Noble. Presented at the ECOOP 2002 Workshop on Formal Techniques for Java-like Programs.
275. ECOOP'99 Intercontinental Workshop on Aliasing in Object-Oriented Systems. James Noble and Jan Vitek and Paulo Sergio Almeida. In Object-Oriented Technology: ECOOP'99 Workshop Reader. Ana Moreira and Serge Demeyer (eds). LNCS 1743. Springer-Verlag 1999.
276. The Objects of Aliasing. James Noble. ECOOP'99 Workshop on Aliasing in Object-Oriented Systems. In Object-Oriented Technology: ECOOP'99 Workshop Reader. Ana Moreira and Serge Demeyer (eds). LNCS 1743. Springer-Verlag 1999.
277. Three Features for Component Frameworks. James Noble. ECOOP'99 Workshop on Component Oriented Programming. In Object-Oriented Technology: ECOOP'99 Workshop Reader. Ana Moreira and Serge Demeyer (eds). LNCS 1743. Springer-Verlag 1999.
278. Overcoming Representation Exposure. David Clarke, James Noble, John Potter. ECOOP'99 Workshop on Aliasing in Object-Oriented Systems. In Object-Oriented Technology: ECOOP'99 Workshop Reader. Ana Moreira and Serge Demeyer (eds). LNCS 1743. Springer-Verlag 1999.
279. Toward Reusable Synchronisation for Object-Oriented Languages. David Holmes, James Noble, John Potter. In Object-Oriented Technology: ECOOP'98 Workshop Reader. Serge Demeyer and Jan Bosch (Eds.) LNCS 1543, Springer-Verlag, 1998.

Patent Application

280. Industrial automation display arrangement and method. Helmut Windl, Helmut Fritz, Larry Constantine, James Noble. United States Patent Application 20020070972, 2002.

Workshop Papers and Reports

281. Recent changes and language design issues in Grace. Andrew Black, Kim Bruce, James Noble. In the Workshop on the Grace Programming Language (GRACE) at the European Conference for Object-Oriented Programming, 2016.
282. Swapsies on the Internet: First Steps towards Reasoning about Risk and Trust in an Open World. Sophia Drossopoulou, James Noble, Mark S. Miller. PLAS@ECOOP 2015: 2-15
283. Delegation isn't quite inheritance. James Noble. The 0th Workshop on New Object-Oriented Languages (NOOL) 2015, at OOPSLA 2015, Pittsburgh, PA, 2015
284. Should Conferences Respect Software. Shriram Krishnamurthi, James Noble, Jan Vitek. Panel Abstract, Companion to the Proceedings of ACM SPLASH, 2013.
285. Wikipublisher: a print-on-demand Wiki. John Rankin, Craig Anslow, James Noble, Brenda Chawner, Donald Gordon. International Symposium on Wikis (WIKISYM) 2009
286. Introduction to Computer Science & Software Engineering at Victoria University of Wellington. Presentation to EU COST Action IC0701 Meeting, Lisbon, Portugal, June 2009
287. Abstraction vs Verification: Classes vs Patterns: Presentation to EU COST Action IC0701 Meeting, Lisbon, Portugal, June 2009
288. The Art of Funding your Research. Presentation to the BuildIT Emerging Researchers' Talent Camp, Auckland, 2009.
289. Experience on the Human Side of Agile. Angela Martin, Robert Biddle, James Noble. XP2008.
290. Teaching Object-Oriented Modelling with Alloy. James Noble, David J. Pearce, Lindsay Groves. OOPSLA Educators' Symposium, Nashville, 2008.
291. Introducing Software Modelling with Alloy at VUW. James Noble, David J. Pearce, Lindsay

- Groves. SIENZ — Software Engineering New Zealand 2007.
292. X3D Web Software Visualisation in Action. Craig Anslow, James Noble, Stuart Marshall, Robert Biddle. OOPSLA Companion 2007. p734
 293. Software Visualization Tools for Component Reuse. Craig Anslow, Stuart Marshall, James Noble, and Robert Biddle. Second Workshop on Method Engineering for Object-Oriented and Component-Based Development at OOPSLA 2004. Magdy Serour, eds. Centre for Object Technology Applications and Research (COTAR), Sydney, 2004.
 294. Angela Martin, Brenda Chawner, Robert Biddle, and James Noble. Requirements with XP and the open source customer. In Brian Fitzgerald and David Parnas, editors, Proceedings of the Workshop on Free and Open Source Software, Genoa, Italy, 2003
 295. Checking Ownership and Confinement Properties. Alex Potanin and James Noble. Presented at the ECOOP 2002 Workshop on Formal Techniques for Java-like Programs.
 296. Lightweight Web-Based Tools for Usage-Centered and Object-Oriented Design. James Noble and Robert Biddle. Presented at forUSE 2002, Portsmouth, New Hampshire, August 2002.
 297. The Language of Programming. Robert Biddle and James Noble. Presented at the Feyerabend Workshop at ECOOP 2002.
 298. Visualising Objects-Oriented Systems. James Noble. Presented at Schloss Dagstuhl Seminar 01211, Schloss Dagstuhl Report No. 307, 2001
 299. Dynamic Flexible Alias Protection in a Prototype-Based Language. James Noble, David Clarke, John Potter. Thinking with Prototypes — OOPSLA'98 Workshop on Prototype-based Object-Oriented Programming.
 300. GUI Iterator and GUI Decorator. James Noble. OOPSLA'97 Workshop on Non-software Examples of Design Patterns. 1997.
 301. Prototype Based User Interfaces. James Noble. ECOOP'97 Workshop on Prototype-Based Programming. Jyvaskyla, Finland. 1997.
 302. Aspects of Synchronisation. David Holmes, James Noble, and John Potter. ECOOP'97 Workshop on Aspect Oriented Programming, Jyvaskyla, Finland. 1997.
 303. On Prototype Based User Interfaces. James Noble. COTAR'96 Workshop. Melbourne, 1996.
 304. ECOOP'96 Workshop on Prototype Based Object Oriented Programming. James Noble and Antero Taivalsaari. ACM SIGPLAN Notices, December 1996.
 305. Encapsulators in Self. James Noble. ECOOP'96 Workshop on Prototype Based Object Oriented Programming. Linz, Austria. 1996
 306. Scribble: A Prototype Lightweight Design Tool. James Noble. COTAR'95 Workshop. Sydney, 1995.
 307. Some Patterns of Tarraingim. James Noble. OOPSLA'95 Workshop on Frameworks for Software Toolsets. Austin, Texas. 1995.
 308. Moses and Aaron. James Noble and Karin Kolbe. OOPSLA'95 Workshop on CASE Frameworks. Austin, Texas. 1995.

Technical Reports and Miscellaneous Publications

309. Shallow Types for Insightful Programs: Grace is Optional, Performance is Not. Richard Roberts, Stefan Marr, Michael Homer, James Noble. CoRR abs/1807.00661 2018
310. More Reasoning about Risk and Trust in an Open World. S. Drossopoulou, J. Noble, M. S. Miller, and T. Murray. ECSTR-15-08, School of Engineering and Computer Science, Victoria

University of Wellington, October 2015.

311. Towards Capability Policy Specification and Verification. Sophia Drossopoulou and James Noble. ECSTR14-05. School of Engineering and Computer Science, Victoria University of Wellington, July 2014.
312. Towards a Semiotics of Object- and Aspect-Oriented Design. James Noble, Robert Biddle, Ewan Tempero. Abstract accepted for presentation at the European Conference on Computing and Philosophy (ECAP 2006), Trondheim Norway.
313. Oppositional Play in Action, Barr, P., Noble, J., and Biddle, R. Poster presented at Children's Learning in a Digital World Conference, 2005
314. Visualisation for learning OOP, using AOP and eclipse. Khaled, R., Luxton, A, Noble, J., Ferres, L., Judy, B., and Biddle, R. Poster at the 19th annual ACM SIGPLAN conference on Object-Oriented Programming Systems, Languages, and Applications, 2004.
315. Oh! Gee! Java! Ownership types for (almost) free. James Noble, Robert Biddle. 2003.
316. Object Ownership to Order. David Clarke, Ryan Shelswell, John Potter, James Noble, 1998.
317. Adaptive Gridlines for Aligning Objects. James Noble, MRI Technical Report, 1997.
318. Abstract Constraints. James Noble. MRI Technical report, 1996.
319. Writers' Workshop Patlets. James Noble. MRI Technical report, 1996.
320. Emergent Architecture. James Noble. MRI Technical Report, 1996.
321. Java to Eiffel Translator — Project Bruce Initial Report. James Noble. MRI Technical Report, 1996.
322. Shaping Object Oriented Programs. Barry Jay and James Noble. UTS Technical Report, 1996.
323. Book Review: Object-Oriented Development: Building CASE Tools with C++ by David Brumbaugh. R. James Noble. SIGPLAN Notices
324. Book Review: Parallel Logic Programming by Akikazu Takeuchi. R. James Noble. SIGPLAN Notices, 30(4), April 1995.