

Mark Harman's CV Summary

Independent data sources on Mark: Google Scholar; DBLP; Semantic Scholar; EPSRC.

Research Grants

Total funding as lead investigator (PI): **£14,687,806**

Career EPSRC funding proposal success rate: **74%** (17/23)

5 of my projects were ranked in first place by the prioritization panel

Grant highlightsI was principal investigator EPSRC **platform (2009-2014)** and EP-SRC **programme (2012-2017)** grants and currently hold an **ERC advanced fellowship (2017-2022)**.

PhD Supervision

Successfully completed PhDs: **27** (19 as first supervisor; 8 as second supervisor)

Peer reviewed papers

Total: 295

Journals: 103

Conferences: 192

Principal Peer Esteem Indicators

Editorial boards: TSE, TOSEM, JSS, EMSE, SEJ, STVR, SQJ, IST, JSEP.

Program chair: ICSE '18, FSE '15, ISSTA '13, ICST '11, ICSME '04 and others.

Program committee membership:	243
Program chair:	13
General chair:	8
Special issue editor:	17
Keynotes and invited conference talks:	35
Best paper awards:	11
H-index on Google Scholar:	70 (20,124 citations; 28th March 2018)

Member of the EPSRC college since 2003 & ICT Strategic Advisory Team (SAT) 2008–2013. I have 3 papers in the all-time top 100 highly cited papers on Software Engineering¹ and 5 of the top 10 on Search based Software Engineering².

Recent Management and Leadership Roles

I am currently engineering manager of the Sapienz Search Based Software Engineering team at Facebook, London. Previously, I was CREST centre director (2006-2017; 30 direct reports). Head of Software Systems Engineering (2012-2017; ~80 transitive reports). I co-founded the field of Search Based Software Engineering (SBSE), which has ~ 1700 active researchers in 42 different countries. Departmental Research Excellence Framework (REF2014) Submission Lead: I was the principal architect of UCL CS's first place ranking in the 2014 Research Excellence Framework.

Scientific Advisory Roles and Notable Awards/Prizes

I am a member of the scientific advisory board of the Swedish Wallenberg \$100M Autonomous Systems Program (WASP) and the University of Durham Department of Computer Science. GECCO human competitive results (HUMIES), Gold Medal and Bronze Medal 2016, silver medal in 2017 and 2014.

¹I&ST 2016

²SSBSE 2011

Mark Harman's CV Summary (Industry Page)

Vision for the Software Industry

I co-founded the field of Search Based Software Engineering (SBSE) in 2001. The key principle that underpins all my work is to shift software engineering effort from the construction of specific solutions to the construction of search spaces in which those solutions reside. As a result of this shift, many different computational search strategies can now be used to intelligently (and largely automatically) search the space. I believe that, in this way, SBSE offers us a near-optimal combination of the complementary abilities of humans and machines for the task of software development.

Current Industrial Work

Since 2017, at Facebook, the team I manage has been applying this foundational SBSE principle to the development of a system for practical and scalable search based software test design. This is the first system in the world to provide fully friction-free fault finding at this scale: no friction for software developers (who are freed from the tedious, time-consuming task of test case design). Also, no friction for software users (who are freed from finding faults in the software products they use).

Selected Previous Industrial Projects

My PhD students, postdocs and I worked on projects for many companies including Amazon, Daimler, Ericsson, Google, Huawei, IBM, Microsoft, and Motorola. Here is a sample of such industrial projects in reverse-chronological order:

Visa Inc; 2014–2016: I led a team of 4 working on metamorphic testing and test case prioritisation for fraud detection systems, and the application of game theory to ameliorate the fault-severity inflation problem.

Google; 2009-2011: I worked with my PhD student Shin Yoo (now associate prof. at KAIST) on optimised regression testing. The work demonstrated how SBSE techniques could reduce the time needed to find regressions faults. Its findings were published at FSE 2011 and Google's conference GTAC. Subsequent work on regression optimisation at Google is ongoing, but now led by others in the research community.

Microsoft; 2007-2008: I worked with my PhD student Kiran Lakhotia (now CTO at KyePot) on search based software testing, which was incorporated into the Pex tool, and subsequently released as part of Visual Studio in 2012. This work has been cited by Microsoft for its research impact (at ASE 2014) and was one of the research impact case studies included in the UK Research Excellence Framework 2014.

DaimlerChrysler; 2001-2009: I led a team 5 working on incorporating static analysis and search based software testing into DaimlerChrysler's automated testing framework. We had a series of projects, many of which led to highly-cited and influential research publications. I designed and implemented the variable dependence analysis system, VADA, which was successfully deployed to DaimlerChrysler's developers.

Start-ups and Spin-outs

I co-founded the automated test optimisation start-up MAJICKE (liquidated due to the three founders moving to Facebook) and the app analytics spin-out APPREDICT.

Mark Harman's CV (Details)

Contact Details

Mark Harman,
Engineering Manager, Facebook London,
Professor of Software Engineering,
CREST Centre,
Systems Software Engineering Group,
Department of Computer Science,
University College London.

UCL e-mail: Mark.Harman@ucl.ac.uk

UCL web: <http://www.cs.ucl.ac.uk/staff/mharman>

Facebook Research web: <https://research.fb.com/people/harman-mark/>

UCL Personal Assistant: Katie Bourke: crest-admin@ucl.ac.uk; +44 (0)20 7679 0325 (Direct Dial)

Qualifications

The Emmbrook Comprehensive School, Wokingham, Berkshire. 1977-1984
8 O'Levels, 2 A/O Levels and 3 A'levels (all grade A).

Imperial College of Science & Technology, University of London. 1984-1988.
M.Eng. (2:1) in Software Engineering.

Polytechnic of North London. 1988-1992.

Ph.D. in Computer Science.

Thesis title: *Functional Models of Procedural Programs.*

Examiners: John Darlington (Imperial) & Dan Simpson (Brighton).

Employment History

Institution	Dates	Position
Polytechnic of North London	9/1988 – 1/1991	Research Assistant
Polytechnic of North London	1/1991 – 7/1995	Lecturer in Computing
University of North London	7/1995 – 7/1997	Departmental Director of Research
University of North London	7/1997 – 1/1998	Head of Department
Goldsmiths College	1/1998 – 4/2000	Lecturer in Computer Science
Brunel University	4/2000 – 8/2003	Lecturer in Computing
Brunel University	9/2003 – 9/2004	Reader in Computing
Brunel University	9/2004 – 8/2004 [†]	Professor of Computing
King's College, London	8/2004 – 8/2010	Professor of Software Engineering
University College London	8/2010 – 2/2017	Professor of Software Engineering
University College London	2/2017 –	Professor of Software Engineering (part time)
Facebook, London	2/2017 –	Engineering Manager (full time)

[†]I moved to KCL before the promotion took effect.



Figure 1: The Global Spread of SBSE Authorship 2001-2014

Research Overview and Principal Contributions

My principal research contributions lie in the areas of Source Code Analysis, Testing and Search Based Software Engineering (SBSE).

Of the three, SBSE will probably prove to have the greatest long term impact. In 2001 I published the paper on Search based Software Engineering (SBSE) that coined the term and, essentially, founded the discipline. Other researchers played critical roles in founding this field of study and there were also papers on the application of computational search to aspects of Software Engineering prior my 2001 paper. However, my 2001 paper was the first to argue that computational search (metaheuristic algorithms) could and should be applied right across the spectrum of software engineering activities. In this paper and my subsequent work, I demonstrated that there is a strong intellectual and technical fit between the problems in Software Engineering and their potential solution using computational search algorithms.

From 1998 to 2002, I was the director of the SEMINAL network (Software Engineering using Metaheuristic INnovative ALgorithms). Since then, I have devoted a large part of my effort to developing the SBSE agenda, providing methods, tools and techniques for SBSE. Figures 1 and 2 show the growth and global spread of authorship of scientific papers on SBSE since my 2001 paper. As can be seen, there has been an explosion of SBSE activity. My own citation growth profile, according to Google Scholar, can be seen in Figure 3.

SBSE is now sufficiently large a topic to warrant more than 12 detailed surveys and analyses of aspects of the literature and regional workshops on SBSE in Brazil (founded in 2010), China (founded in 2012) and North America (founded in 2015), as well as an international conference (since 2009) and track at GECCO (since 2002). A recent retrospective³ on the growth of SBSE in Brazil published in the *Journal of Systems and Software* highlights the key role played by my work in founding and leading the development of the international SBSE research agenda.

There are also a number of SBSE tools that build on the scientific foundations of SBSE and which support SBSE applications right across the range of software engineering activities, from release planning, through design to testing refactoring and patching. My team and I contributed several of these, notably in the area of testing, where our tools have been used by Daimler, Google and Microsoft among others.

Recent surveys of the SBSE literature covering specific subareas such as estimation, non-functional properties, requirements, project management, software design, software product lines and testing as well as more general topics in software engineering. The prevalence of these surveys demonstrates that the general area of SBSE, is already developing flourishing and growing sub-areas and associated research communities.

³ *Search Based Software Engineering: Review and analysis of the field in Brazil* by Thelma Elita Colanzi, Silvia Regina Vergilio, Wesley Klewerton Guez Assuncao and Aurora Pozo. *Journal of Systems and Software*, 86(4):970-984; April 2013.

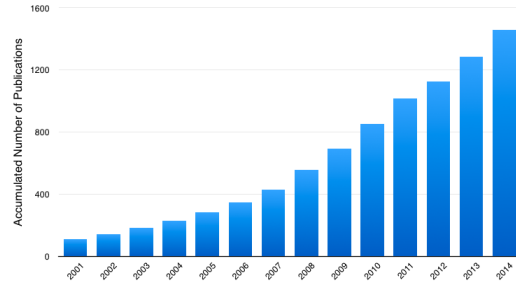


Figure 2: Growth in SBSE publications; 2001–2014 (source: SBSE repository; 2014 last year for which it was feasible, due to rapid growth, to collect data).

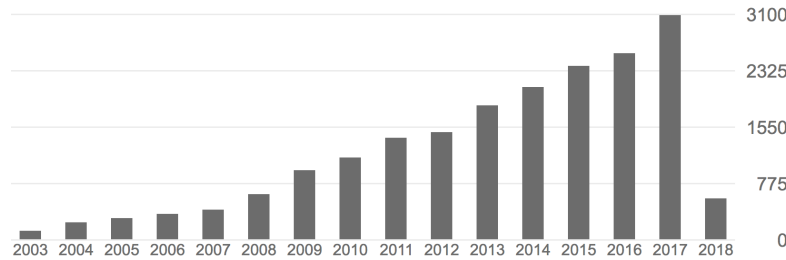


Figure 3: Annual Growth in citations to Mark Harman's work (since 2003). Total cites 20,124. H index: 70 (source: Google Scholar, as of March 28th 2018)

Quantitative Evidence for Research Quality

My **H index is 70** and I have **over 20,000 citations** to my work according to Google Scholar. Based on the ten years' research, 2002–2012, Microsoft Academic Search ranked me in **2nd place in the world for Software Engineering (out of 80,155 software engineers)** and its 5 year ranking placed me in **1st place (out of 27,262 software engineers)**. The Microsoft Academic Search system was decommissioned in 2012, but I believe these rankings have held up or improved since then as suggested by the Google scholar citation profile (see Figure 3).

A 2011 ten-year retrospective⁴ on SBSE work since I coined the term in 2001 ranked me first place among the (800 then, now more than 1,600) authors in the field of SBSE. The same survey reported that 5 of my papers occupied positions in the international top 10 papers over all authors by citation count.

A more recent survey of the 'top 100 papers' in Software Engineering⁵, listed three of my papers in the top 100 (only one scholar, Vic Basili, had more, with four, according to the survey).

I have been invited to give over **40 keynotes** at international scientific conferences and workshops (most recently including SSBSE 2018, SEFM 2018, ICSR 2017, CBSOft 2016, ICST 2015, SPLC 2014, WCRE 2013, GECCO 2013, ASE 2012 and ESEM 2012), and many other invited talks. I have published widely on SBSE across a variety of software engineering applications from requirements engineering and software management through to software maintenance and testing. Many of my papers are in the top 1% of all cited papers in Computer Science, with several in the top 0.1% (according to Thomson Reuters Essential Science Indicators) and my co-authors and I have received 11 best paper awards, 2 SSBSE challenge track winners, an ACM distinguished paper award (2015) and four GECCO Humie medals (Gold and Bronze in 2016 and Silver in 2017 and 2014).

⁴F. G. Freitas and J. T. Souza: *Ten years of search based software engineering*, 3rd International Symposium on Search based Software Engineering (SSBSE 2011), pages 18?-32, Springer LNCS, 2011.

⁵V. Garousi and J. M. Fernandes: *Highly-cited papers in software engineering*, Information and Software Technology, Volume 71, Pages 108?128, 2016.

Producing and Nurturing Leaders

I have successfully supervised many PhD students. Among them are many excellent scientists, who are developing as leaders in their own right. For example:

Shin Yoo, who is now a lecturer (assistant professor) at KAIST and a world-leading expert on SBSE for regression testing. Dr. Yoo was SBSE track co-chair for the ACM Genetic and Evolutionary Computation Conference (GECCO 2012) and program chair for the 7th International Symposium on Search Based Software Engineering (SSBSE 2014). His 2012 Software Testing, Verification and Reliability paper (co-authored with me) on Regression Testing has over 800 citations.

Zheng Li, who is now full professor at Beijing University of Chemical Technology and a leader of the Chinese SBSE community. Prof. Li was the chair of the 1st Chinese SBSE workshop (CSBSE 2012) and the 12th IEEE Working Conference on Source Code Analysis and Manipulation. His 2007 paper (co-authored with me) on test case prioritisation in IEEE Transactions on Software Engineering already has nearly 600 citations.

Yuanyuan Zhang, who is now a principal research associate in the CREST centre at UCL and world-leading authority of SBSE for requirements. Dr. Zhang was the program co-chair for the 5th International Symposium on Search Based Software Engineering (SSBSE 2013). Her 2009 ACM Surveys paper on SBSE (co-authored with me) has over 600 citations (including its previous Technical Report version).

Yue Jia, who is a full time engineer in my team at Facebook, a co-founder of the MaJicKe start-up and part-time lecturer (aka Assistant Professor) at UCL. Yue is also a world-leading mutation testing expert. Dr. Jia has been program chair for two IEEE Mutation Testing workshops and guest editor of a special issue of Software Testing, Verification and Reliability on Mutation Testing. His 2011 paper (co-authored with me) on Mutation Testing in IEEE Transactions on Software Engineering already has over 900 citations.

Research Excellence Framework (REF 2014)

I the leader and principal architect of the University College London Computer Science submission to the Research Excellence Framework (REF 2014). We moved up in research ranking from equal **5th (out of 81) in 2008 to 1st place (out of 89) in 2014** according to grade point average (3.57/4) and percentage of 4* research (61%). These are the two criteria used by HEFCE itself, the funding body that implements REF, to identify high-performing units of assessment.

This performance improvement was undoubtedly largely the product of hard work and the department's excellent environment and its recruitment and retention of world-leading faculty members. Nevertheless, I believe my own skills in optimisation, management and leadership played an important role in ensuring that we achieved the best possible outcome.

I was a **co-author of 13 of the research papers submitted** to the exercise by UCL, and was also the **author of an impact case study** submitted by King's College London Department of Informatics (which also performed well in the assessment, moving from 31st place to 14th place according to GPA).

Funding

Quantitative Overview

Project Title	Source of Funding	year awarded	Principal or Co Investigator	Amount of Funding
SEMINAL	EPSRC	1999	Principal	£51,112
GUSTT	EPSRC	2000	Principal	£135,111
TeTra	EPSRC	2003	Principal	£173,998
SBSE	EPSRC	2003	Principal	£10,319
ASTRENet	EPSRC	2004	Principal	£60,372
CONTRACTS	EPSRC	2004	Principal*	£381,347
Two development grants	DaimlerChrysler	2001–2004	Principal	£29,000
SEBASE	EPSRC	2006	Principal	£1,145,357
EvoTest	EU STReP	2006	Principal	£251,000
Osaka Collaboration	Diawa Foundation	2006	Principal	£3,900
TAIC PART 2006	EPSRC	2006	Principal	£15,208
Dependence Analysis	DaimlerChrysler	2006	Principal	£25,000
CREST support grant	Vizuri	2006	Principal	£5,000
TAIC PART 2007	EPSRC	2007	Principal	£27,531
A-Club	EPSRC	2007	Principal	£59,898
SLIM	EPSRC	2008	Principal	£593,000
SCALE	PMI	2008	Principal	£35,000
Motorola Foundation	Motorola	2009	Principal	£20,000
SSBSE	EPSRC	2009	Principal	£17,398
CREST platform grant	EPSRC	2009	Principal	£1,130,441
MOERTest	Google	2010	Principal**	£5,533
FITTEST	EU	2010	Principal	£480,000
RECAST	EPSRC	2010	Principal	£353,915
GISMO	EPSRC	2011	Principal	£502,415
SCARTO	Google	2012	Principal	£24,517
DAASE	EPSRC	2012	Principal	£6,809,874
GGGP	EPSRC	2015	Principal	£581,560
EPIC	ERC	2017	Principal	£1,760,000
TOTAL for which I am Principal Investigator				£14,687,806
TOTAL income for my own institution and for which I am Principal Investigator				£10,177,932

Project Title	Source of Funding	year awarded	Principal or Co Investigator	Amount of Funding
FORTEST	EPSRC	2001	Co	£59,385
TestCom	EPSRC	2004	Co	£9,297
Linear Schemas	EPSRC	2006	Co	£301,452
Hewlett Packard FOSSOLOGY	HP	2010	Co	£25,000
InfoTestSS: Info. theory & Test Suite Selection	EPSRC	2016	Co	£448,002
TOTAL for which I am Co-Investigator				£843,136

Current Funded Projects for which I am Principal Investigator

EPIC: Evolutionary Program Improvement Collaborators: EP/M025853/1: October 1st 2017 - 30th September 2022

EPIC is an European Research Council (ERC) Advanced Fellowship Grant (2.1m euros). EPIC will automatically construct Evolutionary Program Improvement Collaborators (called Epi-Collaborators) that suggest code changes that improve software according to multiple functional and non-functional objectives. The Epi-Collaborator suggestions will include transplantation of code from a donor system to a host, grafting of entirely new features ‘grown’ (evolved) by the Epi-Collaborator, and identification and optimisation of tuneable ‘deep’ parameters (that were previously unexposed and therefore unexploited).

A key feature of the EPIC approach is that all of these suggestions will be underpinned by automatically-constructed quantitative evidence that justifies, explains and documents improvements. EPIC aims to introduce a new way of developing software, as a collaboration between human and machine, exploiting the complementary strengths of each; the human has domain and contextual insights, while the machine has the ability to intelligently search large search spaces. The EPIC approach directly tackles the emergent challenges of multiplicity: optimising for multiple competing and conflicting objectives and platforms with multiple software versions.

GGGP: Grow and Graft Genetic Programming: EP/M025853/1: 28th October 2015 - 27th October 2019

The GGGP project is motivated by the frustration often expressed as questions such as “Why do software engineers spend so long repeatedly performing the same tedious low level software development tasks?” and “How many times do programmers work out how to express the idea of null pointer checking in a particular context or adapt existing code for searching an iterated data structure?” We propose a new approach to software development: Grow and Graft Genetic Programming (GGGP), in which a new feature is grown (using genetic programming) and subsequently grafted into an existing system. This grow and graft development approach aims to reduce the amount of tedious effort required by human programmer in order to develop and add new functionality into an existing system. Our initial proof of concept work found that surprisingly little human guidance and domain knowledge is required from the programmer to guide Grow and Graft Genetic Programming. The funding is primarily for the support of the senior research fellow, Bill Langdon, who will develop this grow and graft genetic programming research agenda. The initial industrial partners are Microsoft and Visa Europe Limited. I handed over the PI role of GGGP to David Clark in February 2017 when I moved full time to Facebook.

DAASE: Dynamic Adaptive Automated Software Engineering: EP/J017515/1: 1st June 2012– 30th November 2019

The aim of this project is to develop dynamic optimisation techniques that will provide automated adaptivity for both software products and the processes that produce them. This will establish a whole new way of making and deploying software. DAASE is an EPSRC programme grant for which I am PI and for which UCL is the lead institution. The total is £6.8m. Though all of the funding comes to UCL in the first instance, as is standard practice with programme grants, the lead institution contracts out research to the other sites. Therefore, of the overall £6.8m, only approximately £2.5m will be spent at UCL. The CIs who will lead at the other sites are John Clark (York, then Sheffield), Xin Yao (Birmingham), Gabriela Ochoa (Stirling) and Edmund Burke (Stirling, then Queen Mary). The local team at UCL also includes Drs. Bentley, Black, Krinke and Yoo. We hired 22FTE three year RAs (or equivalent) over the 6 years of the project and also recruit 26 PhD students. We also expect to appoint at least 6 lectureships and a business development manager. The initial industrial partners were ABB, Berner and Mattner, BT, Ericsson, GCHQ, Honda, IBM, Microsoft, Motorola and NorthropGrumman, though we expect to

add more as the project progresses. I handed over the PI role of DAASE to Earl Barr in February 2017 when I moved full time to Facebook.

Completed Projects for which I was Principal Investigator

GISMO: Genetic Improvement of Software for Multiple Objectives: EP/I033688/1: 2011–2015

This project developed a new way of automatically generating software to meet non-functional requirements from software that already largely meets functional requirements. I believe that our approach has tremendous potential to achieve many of the original hoped-for goals of genetic programming. Bill Langdon is the named Senior Research Fellow and I am the PI. Our industrial partners are IBM UK, Motorola UK and nVidia USA.

SCARTO: 2012

This is a Google award to develop techniques for optimised regression testing and dependence analysis. Like the MORETest project that preceded it, this is a proposal I wrote with Dr. Yoo and though I am officially the PI, I regard him as joint PI.

Motorola Foundation grant: 2009

This was a gift of £20k from Motorola USA to support the work of the CREST lab I lead.

CREST platform: EP/G060525: 2009–2014

This project is a prestigious EPSRC platform grant for the CREST centre that I direct. To be successful, applicants have to satisfy EPSRC peer review of their internationally leading status. In the words of the EPSRC: “Applicants are required to have an internationally leading reputation and a high international profile.”.

RE-COST: REducing the Cost of Oracles for Software Testing: EP/I010165/1: 2010–2013

This project developed techniques to balance the benefits of software testing against the costs involved using multi objective search based software engineering. Previous work on automated test data generation had only focused on the benefits to be gained, which, we argued was unrealistic. Our industrial partners are Berner and Mattner, Berlin, Motorola UK, and Sogeti UK Ltd. I was the PI for the UCL component, and overall project lead, while Dr. Phil McMinn was the PI for the Sheffield component. The UCL income from the grant was £353,915. We opened up a new field of research in *realistic* test case generation that I believe will prove to be important in making automated test generation techniques more effective. Too often automated test systems generate tests which no human tester can understand; we showed how we can enhance automated testing with realistic test case generation.

FITTEST: EU grant agreement 257574: 2010–2013

This project extended and applied the work of the Evotest project to Future internet testing. I was the PI for UCL and I was one of the four primary authors of the proposal (though all partners contributed). My part of the project funded two post doctoral RAs for three years each. We developed and deployed tools for search based testing which have been used by the partners (and others) to find previously unknown bugs in their systems. The project had several practical contributions to better testing. My colleagues and I developed new ways to combine Dynamic Symbolic Execution and Search Based Software Testing that combine the strengths of both and which I hope and believe open up new avenues of research.

SLIM: EP/F059442: 2008-2011

The project developed algorithms and theory for slicing Extended Finite State Machines. Slicing had been well-studied for programs, but less so for state machines. The project employed two post-doctoral researchers, and was supported by industrial partners: Berner and Mattner and Motorola and visiting fellow Prof. Bogdan Korel from Illinois Institute of Technology. We developed new techniques for slicing systems at the model level which

extended slicing to system models. I believe that these techniques are particularly effective since they reduce models from unmanageable size to those that can be fitted onto a single side of A4 paper.

SEBASE: Submitted Jan 2005. Revised and resubmitted July 2005. Funded Nov 2005. EP/D050863: 2006-2011

This was a project in collaboration with Professor Xin Yao (Birmingham), Professor John Clark (York), IBM, DaimlerChrysler and Motorola. I was the PI for the King's College (then UCL, when I moved) part of the project and was also overall project director. The project was for a major research programme in Search Based Software Engineering for £2.7M over five years. The King's portion of the grant was £1,145,357. SEBASE provided scientific evidence for SBSE's applicability throughout the wide spectrum of software engineering activities. The project attracted significant international interest and follow-on work from other researchers. The tools and techniques were developed and deployed in several companies including Google and Microsoft. The project was a finalist for the Times Higher Education Supplement's "Project of the year" award in November 2012.

() Google Award: 2010**

This was an award to pay for a secondment of my then research associate, Shin Yoo, to work with Google on Automated Optimised Regression Testing. The project was called MORETEST (Multi Objective Regression Testing). The work is on-going and has already been published at the Google Testing conference GTAC 2010 (acceptance rate 10%). I wrote the proposal jointly with Dr. Yoo. (**)Formally speaking, I am the principal investigator, but I regard his role as 'joint PI' in this case.

SCALE: PMI25: 2008-2010

SCALE was a project funded by the Prime Minister's Initiative (PMI). It was a partnership between the British Council and the Prime Minister's office. The project explored the link between Mega Software Engineering (as developed at Osaka University) and Search Based Software Engineering (introduced by Harman's Group at King's CREST). The project was primarily a grant for travel to support visits and a 6 month secondment from Osaka to CREST for Dr. Makoto Matsushita.

SSBSE: EP/G04872X: 2009-2009

This project was a small grant to run a symposium on SBSE. The symposium was an outstanding success, with over 60 participants from 14 different countries and helped to secure the UK's leading role in the development of the rapidly growing field of SBSE. I was the general chair. It launched the SSBSE conference, which has continued to develop and grow since then.

A-Club: EP/F010443: 2007-2009

A visiting fellow grant to allow Professor Binkley to spend 6 months in the CREST lab, together with support for a further year of periodic travel to CREST. Dependence clusters were shown to be a major problem in our earlier collaborative work and in the Dependence Analysis project with DaimlerChrysler, 2006. The A-Club project developed techniques to identify the causes of these clusters and ways to remove them from program source code.

Evolutionary Testing (EvoTest) This is a STReP project, awarded in January 2006, with partners in DaimlerChrysler, Motorola, RILA (Bulgarian SME), Zenon (Greek SME), Fraunhofer, INRIA, University of Valencia and King's College London. I was the leader for the King's component of the project, while Tanja Vos, then at the University of Valencia, was the overall project co-ordinator. The King's college portion supported one RA and one PhD and associated costs. The total finding is £2.1M, of which King's received £251,000. The project developed evolutionary testing techniques and tools which were used to find bugs in systems at partner sites, with notable uptake by Daimler Chrysler.

Osaka Collaboration: 2006-2007

This was a small grant from the Diawa foundation to support collaboration with Professor Katsuro Inoue at Osaka University on Mega Software Engineering and Search Based Software Engineering. This pump-primed the larger SCALE project, currently funded by the PMI initiative for £35,000 for which I was also the PI.

TAIC PART 2007: EP/F012535

This was a grant from EPSRC to support the Testing Academia and Industry Conference - Practice and Research Techniques 2007, for which I was the general chair. I also secured sponsorship for TAIC PART 2007 from Ericsson, ElectroMind, Gerrard Consulting, Nokia LDRA and Vizuri. The conference website is at <http://www2007.taicpart.org/>.

(*)CONTRACTS: GR/T22865/01 (PI) GR/T22872/01 (CI): 2004-2007

Ranked number one by the prioritization panel.

CONTRACTS (CONcept assignment To Raise the abstrACTION level of Slicing). This was a grant proposal (GR/T22865) in collaboration with Nicolas Gold at UMIST (GR/T22872). The proposal was for two research assistants, one based at each site. The idea was to combine the expressiveness of the extraction criterion of concept assignment with the semantic guarantees provided by slicing. *I was the principal investigator for the King's portion (GR/T22865) of the grant, which was £174,000. The proposal went to the prioritization panel on 29/7/2004 and was ranked number one. Subsequently, I recruited Nicolas Gold to King's College as a lecturer (he is now SL). EPSRC does not allow two PIs within a single institution so I decided to let Nicolas be the PI for the combined project, which was re-costed as GR/T22872/01 at £381,347. However, we regarded ourselves as joint PIs throughout.

ASTReNET GR/S93684/01: 2004-2007

ASTRENET (Analysis, Slicing and Transformation REsearch NETwork) was network that I led. It received £60,372 from EPSRC. The network combined the analysis, slicing and transformation producer communities with the potential software engineering user communities. We held 22 workshops over a period of 3 years and helped to develop a relationship between software engineering and program analysis that is still evident in collaboration and interactions between the two communities.

TAIC PART 2006: EP/D077095

Evaluated as 'Outstanding' by EPSRC

A grant from EPSRC to support the Testing Academia and Industry Conference - Practice and Research Techniques 2006, for which I was the general chair. I also secured sponsorship for TAIC PART 2006 from DaimlerChrysler, Ericsson, Motorola, LDRA, IPL and Vizuri. The conference website is at <http://www2006.taicpart.org/>.

TeTra: Testability Transformation GR/R98938/01: 2003-2006

Ranked number one by the prioritization panel.

Evaluated as 'Tending to Outstanding' by EPSRC

The goal was to transform programs for which it is hard to automatically generate test data, into programs which are easier to test. This involved novel transformations that need not preserve functional equivalence, but that must preserve test adequacy criteria instead. The funding was for £173,998. The proposal went to the prioritization panel on 24/4/2002 and was ranked number one. It started on 1st Jan 2003. I was the principal investigator. Testability transformation is now an established and widely-used technique for extending the reach of automated test generation tools and techniques.

GUSTT GR/M58719: 1999-2002.

Evaluated as 'Outstanding' by EPSRC

The GUSTT project was a collaboration with Durham University which had a separate grant for £123,000 and for which Malcolm Munro was the Principal Investigator. The GUSTT project grew out of my work on amorphous slicing. It interleaved slicing and transformation with applications to comprehension, reuse, testing and debugging. I wrote the Case For Support and was the principal investigator for the Brunel half of the project (£135,111), where I was a lecturer (aka assistant professor) at the time. On completion, the project was rated 'outstanding' (the highest grade achievable) by the grant review panel that considered it. Amorphous slicing is now a widely studied and well-established part of the overall body of work on program slicing and dependence analysis.

SEMINAL GR/M78083: 1999–2002.**Evaluated as ‘Tending to Outstanding’ by EPSRC**

SEMINAL (Software Engineering using Metaheuristic Innovative Algorithms) was a network project for which I was the Principal Investigator. The partners were Brunel University (where I was at the time), Bristol University, York University, Reading University, BAe, Goldsmiths College, University of Glamorgan, Bournemouth University, BT Labs, QMW, Strathclyde University and Daimler-Chrysler. The grant was held by Brunel and was for a total of £51,112. The aim of the SEMINAL project was to recast software engineering problems as search problems, for which the applicable trade-off criteria can be used to define fitness functions. This project started as an EPSRC Network, but is now an on-going international networked research community with a track in the main international conference on Evolutionary Computation (GECCO) and annual workshops and symposia on SBSE and specific subfields.

SBSE: search Based Software Engineering, GR/S56177/01: 2003**Evaluated as ‘Outstanding’ by EPSRC**

This was a proposal to hold a workshop in September 2003 on Search-Based Software Engineering (SBSE). I was the principal investigator for the project and program and general chair of the workshop. The EPSRC APM handling the grant said that the referees’ comments were the best that he had ever seen. This reflected the growing appreciation of the importance of SBSE. The project was rated ‘outstanding’ and ‘internationally leading’ across the board on completion, with overall evaluation ‘outstanding’.

CREST support grant: 2007

A small grant from Vizuri limited to support the purchase of additional lighting and soft furnishings for the CREST laboratory, raising the standard of the working accommodation. Only a small amount, but vital, since this support was unavailable from any other source.

Dependence Analysis: 2006

A grant from DaimlerChrysler to allow myself and my colleagues, Dr. Gold, Mr Li and Professor Binkley to work on dependence analysis of Daimler’s third party software. This grant was a cross between research and consultancy and was charged accordingly.

DaimlerChrysler Grant/Consultancy: Testability Transformation: 2001–2002

Evolutionary testing is hard to apply in the presence of side effects, unstructuredness and flag variables. This project investigated ways in which program transformation can overcome these barriers to evolutionary program testing. It was funded by DaimlerChrysler. Brunel’s portion is €15,000. I was the principal investigator (and at Brunel University at the time).

DaimlerChrysler Grant/Consultancy: Search Space Reduction: 2002–2003

Evolutionary testing effectiveness is reduced by large search spaces. Some variables which are apparently members of the search space, because they are input variables, do not contribute to the computation under test. Slicing can identify these variables. It was funded by DaimlerChrysler, which has provided €30,000. I was the principal investigator. Chris Fox, Essex University, was the co-investigator.

Current Funded Projects for which I am a Co–Investigator

InfoTestSS: Information theory and Test Suite Selection: EP/P005888/1: 1st October 2016 – 31st March 2020

The project aims to use information theory to understand test suite selection. I have long believed that information theory has a lot to teach us about software testing, and I'm delighted to have the opportunity to collaborate with David Clark and Rob Hierons on this through this project. This project is led by David Clark, in my group who is the PI for the UCL part of the grant. It is a collaboration with Brunel University, for which Rob Hierons is the PI. Industrial partners are Berner and Mattner and J.P. Morgan.

Completed Projects for which I was a Co–Investigator

HP FOSSOLOGY project: 2010

This project grew out of my work with Dr. Jens Krinke and my PhD student Yue Jia on clone detection. The project is mainly the work of Dr. Krinke. I provided some help and guidance with the application. We will use the funding to consider the provenance of open source code using clone detection techniques.

Linear Schemas for Program Dependence EP/E002919: 2006–2009

This project grew out of my theoretical work with Dr. Danicic, my longest serving collaborator. The project reinvigorated research in program schemas, by connecting the theoretical outcomes of this work to practical questions about program analysis. A notable achievement was the demonstration that Weiser's slicing algorithm is minimal for linear, liberal, free schemas, which resolved a problem that had remained open since 1979. It was also possible to prove that equivalence is decidable for linear, liberal, free schemas. This was one of the first practical and positive decidability results since Ianov's foundational work in the early 1960s.

FORTEST GR/R43150: 2001–2004

FORTEST (Formal Methods and Testing Network) was a collaborative network project, for which Rob Hierons (at Brunel) was the co-ordinator and principal investigator. I was a co-investigator and provided a significant contribution to the writing of the Case For Support.

IFIP TestCom Conference GR/S79015/01: 2004

This was a project to support the IFIP TestCom conference which was held at St. Anne's College, Oxford, United Kingdom, 17-19th March 2004. Rob Hierons was the PI and I was the CI. The project was rated 'outstanding' on completion.

Early Career Minor Travel Grants

For the first nine years of my career I was first PhD student, then lecturer and finally head of department (briefly) at the Polytechnic of North London (PNL), which was later the University of North London (UCL) and then subsequently to become London Metropolitan University (some time after I had departed). Until 1997, when modest success in the Research Assessment Exercise (RAE 1996) yielded a small amount of income (£31k per annum for the department), there was no support for research. Consequently there was little or no funding for travel to conferences. To overcome this difficulty, I applied for a number of small travel grants. I also managed and wrote the 1996 RAE submission for UNL's School of Computing.

Organisation	Purpose	Year	Amount
University of Southwestern Louisiana	Research Visit	1998	£400
EU (through TMR)	Conference attendance	1998	£400
The Corporation of London, HERSEF	Conference attendance	1997	£500
Kyoto University	Conference attendance	1996	£400
The Royal Society	Conference attendance	1996	£750
Bournemouth University	Conference attendance	1996	£200
US PDI	Conference attendance	1995	£350
TOTAL			£3,000

Research Supervision

Completed PhDs (first supervisor)

1. Sebastian Danicic. PhD, UNL, 1999: *Dataflow minimal slicing*.
2. Yoga Sivagurunathan. PhD, UNL, 2004: *Modelling dynamic memory allocation and deallocation using amorphous slicing*.
3. Lin Hu. PhD, Brunel, 2004: *Amorphous program slicing*.
4. Kiarash Mahdavi. PhD, Brunel, 2005: *A clustering genetic algorithm for software modularisation with a multiple hill climbing approach*.
5. Ayodeji Fatiregun. PhD, KCL, 2006: *Search-based program transformation for amorphous slicing and program comprehension*.
6. Tao Jiang. PhD, KCL, 2008: *Search based slicing for program dependency structures of interest*.
7. Zheng Li. PhD, KCL, 2009: *Identifying high-level dependence structures using slice-based dependence analysis*.
8. Kiran Lakhotia. PhD, KCL, 2009: *Search-based testing*.
9. Shin Yoo. PhD, KCL, 2009: *Extending the boundaries in regression testing: complexity, latency, and expertise*.
10. Yuanyuan Zhang. PhD, KCL, 2010: *Multi-objective search-based requirements selection and optimisation*.
11. Nadia Alshahwan. PhD, UCL, 2012: *Utilizing output in web application server-side testing*.
12. Jian Ren. PhD, UCL, 2013: *Search based software project management*.
13. Mustafa Bozkurt. PhD, UCL, 2013: *Automated realistic test input generation and cost reduction in service-centric system testing*.
14. Yue Jia. PhD, UCL, 2013: *Higher order mutation testing*.
15. Efstathios Panayi, PhD, UCL, 2015: *Modelling empirical features and liquidity resilience in the Limit Order Book*.
(joint first supervisor with Gareth Peters, UCL Statistics).
16. William Martin, PhD, UCL, 2017 *App Store Analysis and Mining*.
17. Ke Mao, PhD, UCL, 2017 *Crowdsourced software engineering*.
18. Fan Wu, PhD, UCL, 2017 *Genetic Improvement of Dynamic Memory Allocation*.
19. Lingbo Li, PhD, UCL, 2017 *Search Based Requirements optimisation*.

Completed PhDs (second supervisor)

1. Mike Laurence. PhD, Goldsmiths, 2004: *Program schematology*.
First Supervisor: Sebastian Danicic.
2. Lahcen Ouarbya, PhD, Goldsmiths, 2005: *A lazy semantics for program slicing*.
First Supervisor: Sebastian Danicic.
3. Qiang (Larry) Guo, Brunel, PhD, 2005: *Search-based testing*.
First Supervisor: Robert Hierons.
4. Pauline Kan, PhD, KCL, 2006: *A method for safety critical software development*.
First Supervisor: Kevin Lano.
5. Dave Daoudi, PhD, Goldsmiths, 2006: *Conditioned slicing using light-weight theorem proving*.
First Supervisor: Sebastian Danicic.
6. Karnig Derderian, PhD, Brunel, 2007: *Search-based generation of UIOs for FSMs*.
First Supervisor: Robert Hierons.
7. Marian Mohr, PhD, KCL, 2010: *Service oriented software engineering performance enhancement*.
First Supervisor: Nicolas Gold.

8. Syed Islam, PhD, UCL, 2014: *Dependence Clusters*
First Supervisor: Jens Krinke.

Current PhD students

These are students for all whom I was initially either first supervisor (or joint first), during the period where I was full time at UCL. However, in 2017, when I became part time, I moved to become second (or even third) supervisor for most of them (Afnan, Bobby, Carlos, and Matheus). This was to allow other more junior staff to take on more senior PhD supervision roles and to reflect the reduced time I had available for supervision.

Afnan Al Subaihin: *Feature Movements in App Stores*.

— funded with full overseas fees by the Saudi Arabian Government Funding Council.

Bobby Bruce: *Genetic improvement to reduce energy consumption*.

— funded by DAASE project GTA allocation.

Alexandru Marginean: *Automated software transplantation*. (Joint first supervisor with Earl Barr)

— funded by UCL department of computer science excellence award.

Matheus Paixao: *Search based requirements and architecture optimisation*.

— funded with full overseas fees by the Brazilian government's Science without Borders (SwB) programme.

Carlos Gavidia: *Game theoretic software engineering and metamorphic testing*. (Joint first supervisor with Earl Barr)

— funded by Visa-UCL impact studentship.

Currently Second Supervisor

Zheng Gao: *Testing floating point computation*.

— Funded by Earl Barr's startup package.

First Supervisor: Earl Barr.

Gunel Jahangirova: *The Oracle problem in software testing*.

— funded by the FBK scholarship programme I set up in partnership with Paolo Tonella of FBK.

First Supervisor: David Clark.

Siti Omer: *Software analysis*.

— funded with full overseas fees by the Malaysian government.

First Supervisor: Jens Krinke.

Donggyun Han: *modern code review*.

— funded by UCL department of computer science start-up grant (for Shin Yoo). First Supervisor: Jens Krinke.

Research Seminars Given

I have given invited research seminars at the following institutions:

University of Bath,
BCS FACS,
BCS SIGIST,
Beijing University of Chemical Technology
(BUCT), Beijing, China,
Beijing Institute of Technology (BIT), Bei-
jing, China,
Birkbeck,
Birmingham University(2),
Bournemouth University(4),
Brighton University(2),
Bristol University,
Brunel University,
University of Cambridge
Cardiff University,
City University,
Coventry University,
DeMontfort University (Milton Keynes),
DeMontfort University (Leicester),
DaimlerChrysler, Berlin, Germany (5),
The University of Durham(3),
Essex University,
Exeter University
ETH, Zurich, Switzerland,
The University of Glamorgan,
Goldsmiths College,
University of Hertfordshire,
IBM, Hursley (3)
Illinois Inst. Tech., Chicago, USA,
Imperial College (London),
Institute of Software, Chinese Academy of
Sciences (ISCAS), Beijing, China,

The University of Kent at Canterbury(2),
King's College, London (2),
Loyola College, Maryland, USA,
Oxford Brookes University
Peking University, Beijing, China (2),
Royal Holloway University of Salerno,
Italy,
University of Sannio, Italy,
The University of Sheffield(2),
Simula Labs, Norway,
The Southampton Institute,
South Bank University(2),
Stirling University
Strathclyde University(2),
Symbian, London,
University College London,
University of Trento, Italy,
The University of Leicester,
The University of Liverpool,
UMIST,
Microsoft, Cambridge
Middlesex University,
Motorola Labs Basingstoke(3),
University of Nebraska Lincoln, USA,
The University of North London,
The University of Oxford,
The University of Reading(2),
Tsing Hua University, Beijing, China
Vizuri Ltd., London,
The University of Westminster,
The University of York(2).

Press and Media Coverage for scientific work

August 22nd 2017: Ars Technica article about Sapienz deployment, available online.

August 4th 2015: BBC Radio programme 'Click': Live radio interview about automated software transplantation research.

30th July 2015: WIRED article about automated software transplantation. It was the most shared (on social media) of the WIRED articles that week.

9th July 2013: Pre-recorded interview by Frank Grotelschen for German National Radio concerning evolutionary computation for software engineering.

May 5th and 6th 2012: BBC TV programme 'Click'. Pre-recorded interview by Lara Lewington on "why technology appears to fail without explanation".

5th April 2012: BBC Radio 4 programme 'Material World'. Live radio interview by Quentin Copper on the alleged government plans to monitor internet communications data in real time. The five minute discussion briefly touched on issues of false positives and false negatives, recommender systems, Turing, Kafka and the Law of Tendency to Executability.

Teaching

A summary of courses I have taught is given below:

Subject	Level
Software Engineering	B.Sc.
Introduction to Computer Science	B.Sc.
Algorithms and Data Structures	B.Sc.
Programming in Java, C, Modula-2 and Hope	B.Sc.
Distributed Systems	B.Sc.
Programming in C++	M.Sc.
Formal Methods (in Z)	M.Sc.
Testing	M.Sc.
Mathematics for Computer Science	B.Sc. and M.Sc.
Measurement and Testing	B.Sc. and M.Sc.

In 1996 I co-wrote an introductory textbook on Programming in C++ (269pp) aimed at beginners. Approx. 5,000 copies sold (1996-2006).

In 1998 I wrote a study guide (116pp) for the Software Engineering and Development module of the University of London External Programme in Computing and Information Systems (University of London press, 1999, ISBN 07187 1587X).

In 1999 I wrote a study guide (110pp) for the Introduction to Computing module of the University of London External Programme in Computing and Information Systems.

In January 2001 I co-developed and presented a one-week series of industry seminars and lectures to Motorola employees as part of an M.Sc. module on software verification, run by the University of Bath.

Since 1998 I have been an assessor for the University of London External programme, during which time I set and marked courseworks on software engineering and have marked many BSc final year projects.

In 2012 and 2013 I set up and was general chair for the Student Conference on Optimisation of Software (StuConOS) at UCL. In 2013 we combined this with a project prize competition for final year project students. We received 151 project reports and shortlisted 20 projects for the prize, running a two day workshop for the students to present their work.

In 2006 I received a King's College Teaching Excellence Award.

In 2009 I received a King's College Doctoral Supervisory Excellence Award.

Conference Program Committee Memberships (partial list; last updated November 2016)

2018 So far

40th ACM/IEEE International Conference on Software Engineering (ICSE) — Program co-chair

2017 So far

24th IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER)

1st International Workshop on Testing Extra-Functional Properties and Quality Characteristics of Software Systems

9th International Symposium on Search Based Software Engineering (SSBSE)

39th ACM/IEEE International Conference on Software Engineering (ICSE) — Program Board

11th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE)

10th IEEE International Conference on Software Testing, Verification and Validation (ICST)

25th IEEE International Conference on Program Comprehension (ICPC)

39th ACM/IEEE International Conference on Software Engineering - New Ideas and Emerging Results (ICSE-NIER)

20th International Conference on Fundamental Approaches to Software Engineering (FASE)

International Genetic and Evolutionary Computation Conference (GECCO)

2016

38th ACM/IEEE International Conference on Software Engineering (ICSE) — Program Board

23rd IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER)

8th International Symposium on Search-based Software Engineering (SSBSE)

Empirical Software Engineering and Measurement (ESEM)

Automated Testing (A-TEST)

1st International Workshop on Software Refactoring (IWorR)

24th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE)

9th International Workshop on Search-Based Software Testing (SBST)

First International Workshop on Metamorphic Testing (Met)

The 11th IEEE/ACM International Workshop on Automation of Software Test (AST)

5th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE)

The 11th Workshop on Testing: Academia-Industry Collaboration, Practice and Research Techniques (TAIC-PART)

16th IEEE International Working Conference on Source Code Analysis and Manipulation (SCAM)

24th International Conference on Program Comprehension (ICPC)

32nd IEEE International Conference on Software Maintenance and Evolution (ICSME)

IEEE International Conference on Software Testing, Verification and Validation (ICST)

International Genetic and Evolutionary Computation Conference (GECCO)

2015

11th Joint meeting of the European Software Engineering Conference (ESEC)

and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE) — Program co-chair

International Genetic and Evolutionary Computation Conference (GECCO)

IEEE 31st International Conference on Software Maintenance and Evolution (ICSME)

9th International Symposium on Empirical Software Engineering and Measurement (ESEM)

18th International Conference on Fundamental Approaches to Software Engineering (FASE)

IEEE 11th International Conference on Predictive Models in Software Engineering (PROMISE)

23rd International Conference on Program Comprehension (ICPC)

IEEE Automated Software Testing Workshop (AST)

10th International Workshop on Mutation Analysis

International Symposium on Software Crowdsourcing (ISSC)

First North American Search Based Software Engineering Symposium (NasBASE)

4th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE)

IEEE 8th Search Based Software Testing workshop (SBST)

IEEE 15th International Working Conference on Source Code Analysis and Manipulation (SCAM)

IEEE 10th Testing Academia and Industry Conference - Practice and Research Techniques (TAIC PART)

Second Edition of the Workshop on Quality Assurance for Self-adaptive, Self-organising Systems (QA4SASO)

Automated Software Testing (A-TEST)
3rd International Workshop on Software Development Lifecycle for Mobile (DeMobile)
12th Working Conference on Mining Software Repositories (MSR)

2014 Program committee memberships:

36th ACM/IEEE International Conference on Software Engineering (ICSE) – Program board
28th IEEE/ACM International Conference on Automated Software Engineering (ASE) — ERP
ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE)
International Genetic and Evolutionary Computation Conference (GECCO)
8th International Symposium on Empirical Software Engineering and Measurement (ESEM)
IEEE 7th International Conference on Software Testing (ICST)
11th Working Conference on Mining Software Repositories (MSR)
IEEE 9th Mutation Workshop
IEEE 7th Search Based Software Testing workshop (SBST) – Program co-chair
IEEE Automated Software Testing Workshop (AST)
IEEE 14th International Working Conference on Source Code Analysis and Manipulation (SCAM)
6th International Workshop on Search based Software Engineering (SSBSE)
IEEE 9th Testing Academia and Industry Conference - Practice and Research Techniques (TAIC PART)
3rd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE)
1st International Workshop on Combining Modelling with Search- and Example-Based Approaches (CMSEBA 2014)

2013 Program committee memberships:

35th ACM/IEEE International Conference on Software Engineering (ICSE)
28th IEEE/ACM International Conference on Automated Software Engineering (ASE)
10th Joint meeting of the European Software Engineering Conference (ESEC)
and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE)
ACM International Symposium on Software Testing and Analysis, program chair (ISSTA)
International Genetic and Evolutionary Computation Conference (GECCO) – SBSE track co-chair
IEEE Automated Software Testing Workshop (AST)
1st International Workshop on Combining Modelling and Search-Based Software Engineering (CMSBSE)
IEEE 17th European Conference on Software Maintenance and Reengineering (CSMR)
7th International Symposium on Empirical Software Engineering and Measurement (ESEM)
IEEE 29th International Conference on Software Maintenance (ICSM)
IEEE 6th International Conference on Software Testing (ICST)
10th Working Conference on Mining Software Repositories (MSR)
IEEE 8th Mutation Workshop
IEEE 9th International Conference on Predictive Models in Software Engineering (PROMISE)
IEEE 6th Search Based Software Testing workshop (SBST)
IEEE 13th International Working Conference on Source Code Analysis and Manipulation (SCAM)
5th International Workshop on Search based Software Engineering (SSBSE)
IEEE 8th Testing Academia and Industry Conference - Practice and Research Techniques (TAIC PART)
4th International Workshop on Emerging Trends in Software Metrics (WETSOM)
2nd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE)

2012 Program committee memberships:

34th ACM/IEEE International Conference on Software Engineering (ICSE)
ACM International Genetic and Evolutionary Computation Conference (GECCO)
ACM International Symposium on Software Testing and Analysis (ISSTA)
IEEE 28th International Conference on Software Maintenance (ICSM)
IEEE 19th Working Conference on Reverse Engineering (WCRE)
IEEE 5th International Conference on Software Testing (ICST)
IEEE 17th European Conference on Software Maintenance and Reengineering (CSMR)
IEEE 20th International Conference on Program Comprehension (ICPC)
IEEE 12th International Working Conference on Source Code Analysis and Manipulation (SCAM)

IEEE Automated Software Testing Workshop (AST)
IEEE 7th Testing Academia and Industry Conference - Practice and Research Techniques (TAIC PART)
IEEE 7th Mutation Workshop
IEEE 8th International Conference on Predictive Models in Software Engineering (PROMISE)
4th International Workshop on Search based Software Engineering (SSBSE)
IEEE 5th Search Based Software Testing workshop (SBST)
IEEE 2nd Workshop on Regression Testing (Regression 2012)
34th ACM/IEEE International Conference on Software Engineering (ICSE)
50th International Conference on Objects, Models, Components and Patterns (TOOLS Europe)
3rd International Workshop on Automating Test Case Design, Selection and Evaluation (ATSE)
6th International Symposium on Empirical Software Engineering and Measurement (ESEM)
3rd International Workshop on Program Debugging (IWPD)
3rd International Workshop on Emerging Trends in Software Metrics (WETSOM)
Aligning Enterprise, System, and Software Architectures (AESSA)
8th Haifa Verification Conference (HVC)
1st International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE)

2011 Program committee memberships:

ACM International Genetic and Evolutionary Computation Conference (GECCO)
IEEE 27th International Conference on Software Maintenance (ICSM)
IEEE 18th Working Conference on Reverse Engineering (WCRE)
IEEE 4th International Conference on Software Testing (ICST)
IEEE 16th European Conference on Software Maintenance and Reengineering (CSMR)
IEEE 19th International Conference on Program Comprehension (ICPC)
8th Workshop on Model Engineering, Verification, and Validation (MODEVVA)
IEEE Automated Software Testing Workshop (AST)
IEEE 6th Testing Academia and Industry Conference - Practice and Research Techniques (TAIC PART)
IEEE 6th Mutation Workshop
IEEE 7th International Conference on Predictive Models in Software Engineering (PROMISE)
IEEE 3rd International Workshop on Search based Software Engineering (SSBSE)
IEEE 4th Search Based Software Testing workshop (SBST)
ACM International Symposium on the Foundations of Software Engineering (FSE)
IEEE 11th International Workshop on Principles of Software Evolution (IWPSE)
IEEE 1st Workshop on Refactoring and testing (REFTEST)
IEEE 1st Workshop on Regression Testing (Regression 2011)
1st International Workshop on End-to-end Test Script Engineering (ETSE)
21st International Symposium on Logic-Based Program Synthesis and Transformation (LOPSTR)
Quality Assurance for Service-based Applications (QASBA)
37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)
ACM/IEEE 14th International Conference on Model Driven Engineering Languages and Systems (MODELS)

2010 Program committee memberships:

ACM International Genetic and Evolutionary Computation Conference (GECCO)
ACM International Symposium on Software Testing and Analysis (ISSTA)
IEEE 26th International Conference on Software Maintenance (ICSM)
IEEE 17th Working Conference on Reverse Engineering (WCRE)
IEEE 3rd International Conference on Software Testing (ICST)
IEEE 15th European Conference on Software Maintenance and Reengineering (CSMR)
IEEE 18th International Conference on Program Comprehension (ICPC)
IEEE 10th International Working Conference on Source Code Analysis and Manipulation (SCAM)
7th Workshop on Model Engineering, Verification, and Validation (MODEVVA)
IEEE Automated Software Testing Workshop (AST)
IEEE 5th Testing Academia and Industry Conference - Practice and Research Techniques (TAIC PART)
IEEE 5th Mutation Workshop

IEEE 6th International Conference on Predictive Models in Software Engineering (PROMISE)
IEEE ETOOLS workshop
IEEE 1st Workshop on Quality of Model-Based Testing (QuoMBaT)
IEEE 2nd International Workshop on Search based Software Engineering (SSBSE)
IEEE 3rd Search Based Software Testing workshop (SBST)

2009 Program committee memberships:

ACM International Genetic and Evolutionary Computation Conference (GECCO)
IEEE 25th International Conference on Software Maintenance (ICSM)
IEEE 16th Working Conference on Reverse Engineering (WCRE)
IEEE 2nd International Conference on Software Testing (ICST)
IEEE 14th European Conference on Software Maintenance and Reengineering (CSMR)
IEEE 17th International Conference on Program Comprehension (ICPC)
Belgian Netherlands Evolution Conference (BENEVOL)
ETAPS Fundamental Approaches to Software Engineering (FASE)
IEEE 9th International Working Conference on Source Code Analysis and Manipulation (SCAM)
6th Workshop on Model Engineering, Verification, and Validation (MODEVVA)
IEEE 4th Testing Academia and Industry Conference - Practice and Research Techniques (TAIC PART)
IEEE 4th Mutation Workshop
IEEE WebTest 2009
IEEE 2nd Search Based Software Testing workshop (SBST)
IEEE Software Engineering and Knowledge Engineering (SEKE)
IEEE 10th International Workshop on Principles of Software Evolution (IWPSE)

2008 Program committee memberships:

ACM International Genetic and Evolutionary Computation Conference
IEEE 24th International Conference on Software Maintenance
IEEE 15th Working Conference on Reverse Engineering
IEEE 1st International Conference on Software Testing
IEEE 13th European Conference on Software Maintenance and Reengineering
IEEE 16th International Conference on Program Comprehension
IEEE Congress on Evolutionary Computation
IEEE 8th International Working Conference on Source Code Analysis and Manipulation
IEEE Automated Software Testing Workshop (co-located with ICSE 2008)
IEEE 3rd Testing Academia and Industry Conference - Practice and Research Techniques

2007 Program committee memberships:

IEEE 23rd International Conference on Software Maintenance
IEEE 12th European Conference on Software Maintenance and Reengineering
IEEE 15th International Conference on Program Comprehension
ACM International Genetic and Evolutionary Computation Conference
IEEE 14th Working Conference on Reverse Engineering
IEEE 7th International Working Conference on Source Code Analysis and Manipulation
IEEE 2nd Testing Academia and Industry Conference - Practice and Research Techniques
ACM Symposium on Software Testing and Analysis
International Conference on Software Testing (ICSTEST)
3rd Mutation Testing Workshop
IEEE Automated Software Testing Workshop (co-located with ICSE 2007)

2006 Program committee memberships:

IEEE 22nd International Conference on Software Maintenance
IEEE 11th European Conference on Software Maintenance and Reengineering
IEEE 14th International Conference on Program Comprehension
ACM International Genetic and Evolutionary Computation Conference

IEEE 13th Working Conference on Reverse Engineering
IEEE 6th International Workshop on Source Code Analysis and Manipulation
IEEE 1st Testing Academia and Industry Conference - Practice and Research Techniques
IEEE ASTReNet Aspect Analysis Workshop
ACM Symposium on Software Testing and Analysis
LNCS Workshop on Computational Science in Software Engineering
International Conference on Software Testing (ICSTEST)
IBM Centre for Advanced Studies Conference (CASCON 2006)
2nd Mutation Testing Workshop

2005 Program committee memberships:

IEEE 21st International Conference on Software Maintenance
IEEE 10th European Conference on Software Maintenance and Reengineering
IEEE 11th Software Metrics Symposium
IEEE 12th Working Conference on Reverse Engineering
IEEE 13th International Workshop on Program Comprehension
ACM International Genetic and Evolutionary Computation Conference
IEEE 5th International Workshop on Source Code Analysis and Manipulation
6th Metaheuristics International Conference
IEEE International Workshop on Principles of Software Evolution
3rd UK Testing Workshop
Dagstuhl Slicing Seminar - Dagstuhl 05451

2004 Program committee memberships:

IEEE 20th International Conference on Software Maintenance
IEEE 9th European Conference on Software Maintenance and Reengineering
IEEE 12th International Workshop on Program Comprehension
AAAI International Genetic and Evolutionary Computation Conference
IEEE 10th Software Metrics Symposium
IEEE 11th Working Conference on Reverse Engineering
IEEE 4th International Workshop on Source Code Analysis and Manipulation

2003 Program committee memberships:

IEEE 19th International Conference on Software Maintenance
IEEE 8th European Conference on Software Maintenance and Reengineering
IEEE 7th European Conference on Software Maintenance and Reverse Engineering
2nd UK Testing workshop
IEEE 11th International Workshop on Program Comprehension
IEEE 3rd International Workshop on Source Code Analysis and Manipulation
AAAI International Genetic and Evolutionary Computation Conference
IEEE 1st International Workshop on Web Based Systems and Applications
IEEE 10th Working Conference on Reverse Engineering

2002 Program committee memberships:

IEEE 18th International Conference on Software Maintenance
IEEE 10th International Workshop on Program Comprehension
IEEE 26th International Symposium on Applied Computing
IEEE 2nd International Workshop on Source Code Analysis and Manipulation
IEEE 9th International Working Conference on Reverse Engineering
IEEE 8th Workshop on Empirical Studies of Software Maintenance
IEEE 4th International Workshop on Web Site Evolution
AAAI International Genetic and Evolutionary Computation Conference
14th Psychology of Programmers Interest Group workshop

2001 Program committee memberships:

IEEE 17th International Conference on Software Maintenance
IEEE 8th International Working Conference on Reverse Engineering
IEEE 9th International Workshop on Program Comprehension
IEEE 7th Workshop on Empirical Studies of Software Maintenance
IEEE 1st Workshop on Source Code Analysis and Manipulation
IEEE 1st Workshop on Software Engineering using Metaheuristic Innovative Algorithms

2000 Program committee memberships:

IEEE 16th International Conference on Software Maintenance
IEEE 6th Workshop on Empirical Studies of Software Maintenance
IEEE 8th International Workshop on Program Comprehension

1999 Program committee memberships:

IEEE 15th International Conference on Software Maintenance
IEEE 5th Workshop on Empirical Studies of Software Maintenance

Conference Management and Leadership

General chair for

IEEE 1st International Conference on Software Engineering SEMINAL workshop (2001)
IEEE 1st Source Code Analysis and Manipulation Workshop (2001)
AAAI 1st Genetic and Evolutionary Computation Conference track on Search-Based Software Engineering (2002)
IEEE ASTReNet Aspect Analysis Workshop (2006)
IEEE Testing Academic and Industry Conference - Practice and Research Techniques (2006)
IEEE Testing Academic and Industry Conference - Practice and Research Techniques (2007)
IEEE Mutation 2007
IEEE 1st Symposium on Search Based Software Engineering (2009)

Program chair for

40th International conference on software engineering (ICSE 2018)
11th Joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC-FSE 2015)
IEEE 7th Search Based Software Testing workshop (SBST)
1st International Workshop on Combining Modelling and Search-Based Software Engineering (CMSBSE 2013)
ACM Genetic and Evolutionary Computation Conference track on Search-Based Software Engineering (GECCO 2013)
ACM International Symposium on Software Testing and Analysis (ISSTA 2013)
IEEE 4th International Conference on Software Testing (ICST 2011)
AAAI Genetic and Evolutionary Computation Conference track on Search-Based Software Engineering (GECCO 2004)
IEEE 20th International Conference on Software Maintenance (ICSM 2004)
2nd UK Testing workshop (UKTest 2003)
AAAI Genetic and Evolutionary Computation Conference track on Search-Based Software Engineering (GECCO 2003)
IEEE 10th International Workshop on Program Comprehension (IWPC 2002)
IEEE 1st International Workshop on Source Code Analysis and Manipulation (SCAM 2001)

Steering committee for

ACM International Symposium on Software Testing and Analysis (ISSTA)
ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE)
ACM/IEEE International conference on software engineering (ICSE)
International Conference on Software Maintenance (2001–2004)
International Working Conference on Source Code Analysis and Manipulation (SC chair 2001-2004, re-elected 2004-2007)
International Conference on Software Testing (2008-2010)
Awarded leadership award by Steering committee of International Working Conference on Source Code Analysis and Manipulation in 2007.
International Symposium on Search Based Software Engineering (SC chair: 2009, re-elected 2013–)

Other conference chairs

FSE 2011 Doctoral Symposium Chair
ICST 2009 Workshops chair

Journal Editorial Work

Editorial Boards:

ACM Transactions on Software Engineering and Methodology (2013 –)
IEEE Transactions on Software Engineering (2005 – 2009)
IET Proceedings — Software (2005 –)
Information and Software Technology (2005 –)
Journal of Empirical Software Engineering (2005 –)
Journal of Software Testing Verification and Reliability (2002 –)
Journal of Software Maintenance and Evolution (2002 –)
Software Quality Journal (2003 –)
Journal of Systems and Software (2010 –)

Guest editor journal special issues:

Information and Software Technology (December 1998) on Program Slicing.
Information and Software Technology (October 2002) on Source Code analysis and Manipulation.
Software Quality Journal special issue of best papers from SCAM 2003 (2004).
Software Testing, Verification and Reliability special issue of best papers from UK testing workshop (2004).
IEEE Transactions on Software Engineering special issue on Software Maintenance (2006).
Formal Aspects of Computing special issue on Formal Approaches to Testing (2006).
Journal of Systems and Software special issue on Source Code Analysis and Manipulation (2006).
Journal of Empirical Software Engineering special issue on Maintenance and Metrics (2006).
Journal Computers and Operations Research Focussed Issue on Search Based Software Engineering (2006).
Journal of Software Testing, Verification and Reliability special issue of Extended Papers from the TAIC PART 2006
Journal of Software Maintenance and Evolution special issue on Search Based Software Maintenance and Evolution (2008)
Journal of Systems and Software special issue of Extended Papers from the TAIC PART 2007 and Mutation 2007
IEEE Transactions on Software Engineering special issue on Search Based Software Engineering (SBSE) 2010
Journal of Software Testing, Verification and Reliability special issue of best papers from ICST 2011 (2013)
Journal of Software Testing, Verification and Reliability special issue on Mutation testing (to appear)
Journal of Systems and Software special issue on Search Based Software Engineering (SBSE) (to appear)
ACM Transactions on Software Engineering and Methodology special issue of best papers from ISSA 2013 (to appear)

Management and Leadership

I am the manager for the Facebook Sapienz team, working on Scalable SBSE at Facebook London. The Sapienz team was founded at Facebook following the move of myself, Ke Mao and Yue Jia (my former PhD students) to Facebook, from the start-up we co-founded, MaJiCKe. Since we started at Facebook in 2017, we have received outstanding support for Facebook leadership, and the team has consequently doubled in size within its first year. This strong support and investment followed the team's successful deployment, in October 2017, of SBSE on Facebook's primary Android App, which has hundreds of millions of users world wide. I will be presenting the work of the team at F8, the primary Facebook conference in May 2018, together with Ke Mao from my team.

I was previously head of Software Systems Engineering at UCL (March 2012–February 2017). I was also the director of the UCL CREST - the Centre for Research in Evolution Search and Testing, since its foundation in 2006. CREST is the world leader in Search Based Software Engineering. It is also internationally leading in Software Testing and internationally competitive in Source Code Analysis.

I was the head of the King's Software Engineering Section, one of three sections in the department (2004–2010).

I led the ASTRENET, an EPSRC-funded network, working on slicing, analysis and transformation (2004–2007). I also led the EPSRC Software Engineering using Metaheuristic INnovative ALgorithms (SEMINAL) research Network which established the research area of Search Based Software Engineering (1999–2002). I was a member of the steering committee for the FORTEST Network (2003–2006).

I co-founded (with Malcolm Munro) the IEEE International Working Conference on Source Code Analysis and Manipulation (SCAM) in 2001. Since then it has grown steadily in submissions and attendance. I co-founded (with Phil McMinn) the IEEE Testing Academia and Industry Conference – Practice and research Techniques (TAIC PART) in 2006. I founded the International Symposium on Search based Software Engineering in 2009.

I co-founded the Special Interest Group in Teaching Software Engineering (SIGToSE), 1996–1998 and was on the steering committee for the Software Engineering Association (SEA), 1997–1998.

Consultancy

I have provided consultancy services to many organisations, including

Daimler Chrysler (several times)
University of Durham
Bradford and Bingley
Hertfordshire University
Huawei, China
Keele university
University of Leicester
MicroLimit Ltd.
The University of Westminster
Islington Chamber of Commerce
Visa Europe

I was the London Technology Network business fellow for the Department of Information Systems and Computing at Brunel (2002–2004).

UK Grant Reviewing

I have been a member of the EPSRC college since January 1st 2003. I served as a member of the EPSRC ICT Strategic Advisory Team (SAT) 2008-2013. I have served on EPSRC responsive mode panels in January 2003, May 2003, July 2005, December 2007, July 2008, January 2010, DREAM fellows panel (January 2010). I chaired the EPSRC responsive mode panel in December 2007 and the EPSRC large grants panel in July 2008.

I have reviewed proposals for the Leverhulme and Nuffield trusts.

International Grant Reviewing

I have acted as referee for funding proposals from the following international research funding bodies:

NSERC (the Canadian research council)

Research Council of Norway

Qatar National Research Fund

Vienna Science and Technology Fund: Wiener Wissenschafts-, Forschungs- und Technologiefonds

Austrian Research Council: Der Wissenschaftsfonds (FWF)

Swedish Research Council: Vetenskapsrådet

Swiss National Science Foundation (FNSNF)

Athens University of Economics and Business Basic Research Funding Program

The Italian Assessment of Quality in Scientific Research 2004-2010

ETH Zurich

University of Luxembourg

External Examining (updated February 2015)

I examined doctoral theses for the following UK Universities (24 examinations in total):

Brunel (1)
Cambridge (1)
DeMontfort (1)
Derby (1)
Durham (4)
King's London (2)
Oxford (1)
QMW (2)
Imperial (1)
Sheffield (1)
Strathclyde (1)
UCL (6)
York (2)

I examined doctoral theses for the following International Universities (10 examinations in total):

ETH Zurich (1)
Malaga, Spain (1)
University of Michigan (1)
University of Pisa (1)
Polytechnic of Montreal (1)
Queen's University, Canada (1)
Trinity College Dublin, Ireland (1)
TU Delft, Netherlands (1)
Trento, Italy (1)
University of Tampere (1)

I am external examiner at Bristol University (for BSc courses), 2014-. I was External Examiner (2005-2010) for the M.Sc. at the University of Nottingham. I was External Examiner (2008-2010) for BSc and MSc at University College London (UCL). I was External Examiner (2004-2007) for the M.Sc. at the University of Kent. I was the external examiner for the Southampton Institute B.Sc. in Computer Studies (1997-2001) and the University of Westminster Conversion and Advanced M.Sc. courses in Software Engineering and Parallel and Distributed Computing (1998-2001).

Academic Administrative Roles

At PNL (which became UNL in 1992) I had a wide variety of administrative roles, culminating in my appointment as Head of Department. I wrote the 1996 RAE submission for the department (achieving a 2 grade improvement).

At Goldsmiths I was Director of Post-Graduate Study.

At Brunel I was the co-ordinator for level one and represented the department at the Faculty Undergraduate Module Review Committee. I also mentored and guided junior staff in the development of their grant proposals (with some considerable success) and ran staff development sessions for the department and the university on grant writing and on criterion-based assessment. I was promoted to personal chair with effect from October 2004, but moved to King's College, London in August 2004 (before this took effect).

At King's College, London I was head of the Software Engineering Section, the largest group within the department. I wrote that part of the 2008 RAE submission relevant to the Software Engineering section. I was also the Chair of the Industrial Advisory Board, in which role I oversaw a significant increase in industrial involvement. I was the chair of the departmental recruitment committee, in which role I overhauled and led the development of recruitment activities, with significant success. I was the department funding opportunities co-ordinator, in which role I ran workshops and training days for staff on proposal writing and processes. In this role I provided advice and guidance on the following successfully funded EPSRC grants (in addition to those for which I am PI or CI):

EP/C545605: Quantitative Information Flow, PI: David Clark, £87,141,

EP/D059372: Scale-Free Structures: Models and Algorithms, PI: Colin Cooper, £10,065,

EP/D036852: SOSoRNet: Service-Oriented Software Research Network, PI: Nicolas Gold, £63,253,

EP/D062012: Stochastic Local Search Algorithms for Structural Proteomics, PI: Kathleen Steinhöfel, £219,751,

EP/G03012X: Higher-order Refinement Techniques for Model Driven Architecture, PI: Iman Poernomo, £315,590 (ranked number 1).

In 2006 I established the CREST centre (the Centre for Research on Evolution Search and Testing). Initially the centre had 2 lecturing staff (including myself), 3 RAs, 6 PhD students and one full time administrative post. In 2009 I secured an EPSRC platform grant to support the centre. By 2010 the centre had grown to 4 lecturing staff, 8 RAs, 12 PhD students and one full time admin post as well as several long term visiting academics. In August 2010 University College London Department of Computer Science recruited the entire centre from King's College London, occasioning my move to take up a chair at UCL. In 2012 we recruited two new lecturing faculty, bringing the staffing total in 2014 to 6 lecturing staff, 10 RAs, 10 PhD students and one full time admin post. In 2016 CREST celebrated its 10th anniversary with a workshop culminating in a cruise along the Thames, funded by generous sponsorship from Huawei and Zuhlke Engineering.

At UCL I was the departmental lead for the Research Excellence Framework (REF 2014), the head of Software Systems Engineering (SSE) and the director of the CREST centre. I was also the Departmental Funding Co-ordinator.

Publications (last updated Nov 27th 2016)

Journal Papers Published (103)

Ke Mao, Mark Harman, and Yue Jia. Robotic testing of mobile apps for truly black-box automation. *IEEE Software*, 34(2):11–16, 2017.

Shin Yoo, Xiaoyuan Xie, Fei-Ching Kuo, Tsong Yueh Chen, and Mark Harman. Human competitiveness of genetic programming in spectrum-based fault localisation: Theoretical and empirical analysis. *ACM Transactions on Software Engineering and Methodology*, 26(1):4:1–4:30, 2017.

Lingbo Li, Mark Harman, Fan Wu, and Yuanyuan Zhang. The value of exact analysis in requirements selection. *IEEE Transactions on Software Engineering*, 43(6):580–596, 2017.

Federica Sarro, Filomena Ferrucci, Mark Harman, Alessandra Manna, and Jian Ren. Adaptive multi-objective evolutionary algorithms for overtime planning in software projects. *IEEE Transactions on Software Engineering*, 43(10):898–917, 2017.

Ke Mao, Licia Capra, Mark Harman, and Yue Jia. A survey of the use of crowdsourcing in software engineering. *Journal of Systems and Software*, 126:57–84, 2017.

Justyna Petke, Saemundur O. Haraldsson, Mark Harman, William B. Langdon, David R. White, and John R. Woodward. Genetic improvement of software: a comprehensive survey. *IEEE Transactions on Evolutionary Computation*, 2018. To appear.

Anthony Finkelstein, Mark Harman, Yue Jia, William Martin, Federica Sarro, and Yuanyuan Zhang. Investigating the relationship between price, rating, and popularity in the black-berry world app store. *Information & Software Technology*, 87:119–139, 2017.

William Martin, Federica Sarro, Yue Jia, Yuanyuan Zhang, and Mark Harman. A survey of app store analysis for software engineering. *IEEE Transactions on Software Engineering*, 43(9), 2017.

Mel Ó Cinnéide, Iman Hemati Moghadam, Mark Harman, Steve Counsell, and Laurence Tratt. An experimental search-based approach to cohesion metric evaluation. *Empirical Software Engineering*, 22(1):292–329, 2017.

William B. Langdon, Brian Yee Hong Lam, Marc Modat, Justyna Petke, and Mark Harman. Genetic improvement of GPU software. *Genetic Programming and Evolvable Machines*, 18(1):5–44, 2017.

Fan Wu, Jay Nanavati, Mark Harman, Yue Jia, and Jens Krinke. Memory mutation testing. *Information & Software Technology*, 81:97–111, 2017.

William B. Langdon, José Javier Dolado, Federica Sarro, and Mark Harman. Exact mean absolute error of baseline predictor, MARP0. *Information & Software Technology*, 73:16–18, 2016.

Nadarajen Veerapen, Gabriela Ochoa, Mark Harman, and Edmund K. Burke. An integer linear programming approach to the single and bi-objective next release problem. *Information & Software Technology*, 65:1–13, 2015.

Justyna Petke, Myra B. Cohen, Mark Harman, and Shin Yoo. Practical combinato-

rial interaction testing: Empirical findings on efficiency and early fault detection. *IEEE Transactions on Software Engineering*, 41(9):901–924, 2015.

Earl T. Barr, Mark Harman, Phil McMinn, Muzammil Shahbaz, and Shin Yoo. The oracle problem in software testing: A survey. *IEEE Transactions on Software Engineering*, 41(5):507–525, May 2015.

William B. Langdon and Mark Harman. Optimising existing software with genetic programming. *IEEE Transactions on Evolutionary Computation (TEVC)*, 19(1):118–135, Feb 2015.

Mark Harman, Jens Krinke, Inmaculada Medina-Bulo, Francisco Palomo-Lozano, Jian Ren, and Shin Yoo. Exact scalable sensitivity analysis for the next release problem. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 23(2):19:1–19:31, April 2014.

Nadarajen Veerapen, Gabriela Ochoa, Mark Harman, and Edmund K. Burke. An integer linear programming approach to the single and bi-objective next release problem. *Information & Software Technology*, 65:1–13, 2015.

José Javier Dolado, Mark Harman, and Mari Carmen. Equivalence hypothesis testing in experimental software engineering. *Software Quality Journal*, 22(2):215–238, 2014.

Syed Islam, Jens Krinke, David Binkley, and Mark Harman. Coherent clusters in source code. *Journal of Systems and Software (JSS)*, 88:1–24, February 2014.

Kelly Androutsopoulos, David Clark, Mark Harman, Jens Krinke, and Laurie Tratt. State-based model slicing: A survey. *ACM Computing Surveys*, 45(4):53:1–53:36, August 2013.

Mustafa Bozkurt, Mark Harman, and Youssef Hassoun. Testing and verification in service-oriented architecture: a survey. *Journal of Software Testing, Verification and Reliability (STVR)*, 23(4):261313, June 2013.

David Binkley, Nicolas Gold, Mark Harman, Syed S. Islam, Jens Krinke, and Zheng Li. Efficient identification of linchpin vertices in dependence clusters. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 35(2 (Article 7)):7, 2013.

Saswat Anand, Antonia Bertolino, Edmund Burke, Tsong Yueh Chen, John Clark, Myra B. Cohen, Wolfgang Grieskamp, Mark Harman, Mary Jean Harrold, Jenny Li, Phil McMinn, and Hong Zhu. An orchestrated survey on automated software test case generation. *Journal of Systems and Software*, 86(8):1978–2001, August 2013.

Mark Harman, Kiran Lakhota, Jeremy Singer, David White, and Shin Yoo. Cloud engineering is search based software engineering too. *Journal of Systems and Software*, 86(9):2225–2241, 2013.

Shin Yoo, Mark Harman, and David Clark. Fault localization prioritization: Comparing information theoretic and coverage based approaches. *ACM Transactions on Software Engineering and Methodology*, 22(3 (Article 19)), July 2013.

Kelly Androutsopoulos, David J. Clark, Mark Harman, Robert M. Hierons, Zheng Li, and Laurence Tratt. Amorphous slicing of extended finite state machines. *IEEE Transactions on Software Engineering*, 39(7):892–909, July 2013.

- Shin Yoo, Mark Harman, and Shmuel Ur. GPGPU test suite minimisation: search based software engineering performance improvement using graphics cards. *Journal of Empirical Software Engineering*, 18(3):550–593, June 2013.
- Kiran Lakhotia, Mark Harman, and Hamilton Gross. AUSTIN: An open source tool for search based software testing of C programs. *Journal of Information and Software Technology*, 55(1):112–125, January 2013.
- Yuanyuan Zhang, Mark Harman, and Soo Ling Lim. Empirical evaluation of search based requirements interaction management. *Journal of Information and Software Technology*, 55(1):126–152, January 2013.
- Mark Harman, Afshin Mansouri, and Yuanyuan Zhang. Search based software engineering: Trends, techniques and applications. *ACM Computing Surveys*, 45(1):Article 11, November 2012.
- Luay H. Tahat, Bogdan Korel, Mark Harman, and Hasan Ural. Regression test suite prioritization using system models. *Journal of Software Testing, Verification and Reliability*, 22(7):481–506, 2012.
- Shin Yoo and Mark Harman. Test data regeneration: Generating new test data from existing test data. *Journal of Software Testing, Verification and Reliability*, 22(3):171–201, May 2012.
- Phil McMinn, Mark Harman, Youssef Hassoun, Kiran Lakhotia, and Joachim Wegener. Input domain reduction through irrelevant variable removal and its effect on local, global and hybrid search-based structural test data generation. *IEEE Transactions on Software Engineering*, 38(2):453 – 477, March&April 2012.
- Cu Nguyen, Simon Miles, Anna Perini, Paolo Tonella, Mark Harman, and Michael Luck. Evolutionary testing of autonomous software agents. *Journal of Autonomous Agents and Multi-Agent Systems*, 25(2):260–283, 2012.
- Shin Yoo and Mark Harman. Regression testing minimisation, selection and prioritisation: A survey. *Journal of Software Testing, Verification and Reliability*, 22(2):67–120, 2012.
- Juan José Durillo, Yuanyuan Zhang, Enrique Alba, Mark Harman, and Antonio J. Nebro. A study of the bi-objective next release problem. *Empirical Software Engineering*, 16(1):29–60, 2011.
- Yuanyuan Zhang, Mark Harman, Anthony Finkelstein, and Afshin Mansouri. Comparing the performance of metaheuristics for the analysis of multi-stakeholder tradeoffs in requirements optimisation. *Journal of Information and Software Technology*, 53(7):761–773, 2011.
- Alessandro Marchetto, Roberto Tiella, Paolo Tonella, Nadia Alshahwan, and Mark Harman. Crawlablity metrics for automated web testing. *Journal on Software Tools for Technology Transfer*, 13(2):131–149, 2011.
- Sebastian Danicic, Richard W. Barraclough, Mark Harman, John Howroyd, Ákos Kiss, and Michael Rupen Laurence. A unifying theory of control dependence and its application to arbitrary program structures. *Theoretical Computer Science*, 412(49):6809–6842, 2011.
- Mark Harman. Software engineering meets evolutionary computation. *IEEE Computer*, 44(10):31–39, October 2011.

Yue Jia and Mark Harman. An analysis and survey of the development of mutation testing. *IEEE Transactions on Software Engineering*, 37(5):649 – 678, September–October 2011.

David Binkley, Mark Harman, and Kiran Lakhotia. FlagRemover: A testability transformation for transforming loop assigned flags. *ACM Transactions on Software Engineering and Methodology*, 20(3), 2011.

Giulio Antoniol, Massimiliano Di Penta, and Mark Harman. The use of search-based optimization techniques to schedule and staff software projects: An approach and an empirical study. *Software — Practice and Experience*, 41(5):495–519, April 2011.

Kata Praditwong, Mark Harman, and Xin Yao. Software module clustering as a multi-objective search problem. *IEEE Transactions on Software Engineering*, 37(2):264–282, 2011.

Kiran Lakhotia, Phil McMinn, and Mark Harman. An empirical investigation into branch coverage for C programs using CUTE and AUSTIN. *Journal of Systems and Software*, 83(12):2379–2391, 2010.

William B. Langdon, Mark Harman, and Yue Jia. Efficient multi objective higher order mutation testing with genetic programming. *Journal of Systems and Software*, 83(12):2416–2430, 2010.

Torben Amtoft, Kelly Androutsopoulos, David Clark, Mark Harman, and Zheng Li. An alternative characterization of weak order dependence. *Information Processing letters*, 110(21):939–943, 2010.

Mark Harman. Automated patching techniques: The fix is in: technical perspective. *Communications of the ACM*, 53(5):108, 2010. A one page commentary on the paper by Wes Weimer, Stephanie Forrest, Claire Le Goues and ThanhVu Nguyen also in *CACM*, 53(5), 2010.

Shin Yoo and Mark Harman. Using hybrid algorithm for pareto efficient multi-objective test suite minimisation. *Journal of Systems and Software*, 83(4):689–701, 2010.

Richard Barraclough, David Binkley, Sebastian Danicic, Mark Harman, Rob Hierons, kos Kiss, and Mike Laurence. A trajectory-based strict semantics for program slicing. *Theoretical Computer Science*, 411(11–13):1372–1386, 2010.

Karnig Derderian, Rob Hierons, Mark Harman, and Qiang Quo. Estimating the feasibility of transition paths in extended finite state machines. *Journal of Automated Software Engineering*, 17(1):33–56, 2010.

Mark Harman and Phil McMinn. A theoretical and empirical study of search based testing: Local, global and hybrid search. *IEEE Transactions on Software Engineering*, 36(2):226–247, 2010.

David Binkley, Mark Harman, Youssef Hassoun, Syed Islam, and Zheng Li. Assessing the impact of global variables on program dependence and dependence clusters. *Journal of Systems and Software*, 83(1):96–107, 2009.

Mark Harman, David Binkley, Keith Gallagher, Nicolas Gold, and Jens Krinke. Dependence clusters in source code. *ACM Transactions on Programming Languages and Systems*,

32(1), October 2009. Article 1.

Yue Jia and Mark Harman. Higher order mutation testing. *Journal of Information and Software Technology*, 51(10):1379–1393, 2009.

Anthony Finkelstein, Mark Harman, Afshin Mansouri, Jian Ren, and Yuanyuan Zhang. A search based approach to fairness analysis in requirements assignments to aid negotiation, mediation and decision making. *Requirements Engineering*, 14(4):231–245, 2009.

Phil McMinn, David Binkley, and Mark Harman. Empirical evaluation of a nesting testability transformation for evolutionary testing. *ACM Transactions on Software Engineering and Methodology*, 18(3), May 2009. Article 11.

Rob Hierons, Kirill Bogdanov, Jonathan Bowen, Rance Cleaveland, John Derrick, Jeremy Dick, Marian Gheorghe, Mark Harman, Kalpesh Kapoor, Paul Krause, Gerald Luetzgen, Tony Simons, Sergiy Vilkomir, Martin Woodward, and Hussein Zedan. Using formal methods to support testing. *ACM Computing Surveys*, 41(2), February 2009. Article 9.

David Binkley, Nicolas Gold, Mark Harman, Zheng Li, and Kiarash Mahdavi. An empirical study of the relationship between the concepts expressed in source code and dependence. *Journal of Systems and Software*, 81(12):2287–2298, 2008.

Tao Jiang, Nicolas Gold, Mark Harman, and Zheng Li. Locating dependence structures using search based slicing. *Information and Software Technology*, 50(12):1189–1209, 2008.

Sebastian Danicic, , Mark Harman, John Howroyd, and Lahcen Ouarbya. A non-standard semantics for program slicing and dependence analysis. *Logic and Algebraic Programming, Special Issue on Theory and Foundations of Programming Language Interference and Dependence*, 72:123–240, July-August 2007.

David Wendell Binkley, Mark Harman, and Jens Krinke. Empirical study of optimization techniques for massive slicing. *ACM Transactions on Programming Languages and Systems*, 30:3:1–3:33, 2007.

Sebastian Danicic, Mark Harman, Robert Mark Hierons, John Howroyd, and Mike Laurence. Equivalence of linear, free, liberal, structured program schemas is decidable in polynomial time. *Theoretical Computer Science*, 373:1–18, March 2007.

Zheng Li, Mark Harman, and Rob Hierons. Meta-heuristic search algorithms for regression test case prioritization. *IEEE Transactions on Software Engineering*, 33(4):225–237, 2007.

David Wendell Binkley, Nicolas Gold, and Mark Harman. An empirical study of static program slice size. *ACM Transactions on Software Engineering and Methodology*, 16(2):1–32, 2007.

Qiang guo, Robert Hierons, Mark Harman, and Karnig Derderian. Heuristics for fault diagnosing when testing from finite state machines. *Software Testing, Verification and Reliability*, 17(1):41–57, 2007.

Qiang Guo, Rob Hierons, Mark Harman, and Karnig Derderian. Improved test quality using robust unique input/output circuit sequences (UIOCs). *Information and Software Technology*, 48:696–707.

David Wendell Binkley, Sebastian Danicic, Mark Harman, John Howroyd, and Lahcen Ouar-

- bya. A formal relationship between program slicing and partial evaluation. *Formal Aspects of Computing*, 18(2):103–119, 2006.
- Mark Harman, Arun Lakhotia, and David Wendell Binkley. A framework for static slicers of unstructured programs. *Information and Software Technology*, 48(7):549–565, 2006.
- David Binkley, Sebastian Danicic, Tibor Gyimóthy, Mark Harman, Ákos Kiss, and Bogdan Korel. Theoretical foundations of dynamic program slicing. *Theoretical Computer Science*, 360(1):23–41, 2006.
- Sebastian Danicic, David Binkley, Tibor Gyimóthy, Mark Harman, Ákos Kiss, and Bogdan Korel. A formalisation of the relationship between forms of program slicing. *Science of Computer Programming*, 62(3):228–252, 2006.
- David Binkley, Mariano Ceccato, Mark Harman, Filippo Ricca, and Paolo Tonella. Tool-supported refactoring of existing object-oriented code into aspects. *IEEE Transactions on Software Engineering*, 32(9):698–717, 2006.
- Karnig Derderian, Robert Hierons, Mark Harman, and Qiang Guo. Automated Unique Input Output sequence generation for conformance testing of FSMs. *The computer Journal*, 49(3):331–344, 2006.
- Sebastian Danicic, Chris Fox, Mark Harman, Robert Mark Hierons, John Howroyd, and Mike Laurence. Slicing algorithms are minimal for programs which can be expressed as linear, free, liberal schemas. *The computer Journal*, 48(6):737–748, 2005.
- Qiang Guo, Robert Mark Hierons, Mark Harman, and Karnig Derderian. Constructing multiple unique input/output sequences using evolutionary optimisation techniques. *IEE Proceedings — Software*, 152(3):127–140, 2005.
- Nicolas Gold, Mark Harman, David Wendell Binkley, and Robert Mark Hierons. Unifying program slicing and concept assignment for higher-level executable source code extraction. *Software Practice and Experience*, 35(10):977–1006, 2005.
- Sebastian Danicic, Mohammed Daoudi, Chris Fox, Mark Harman, Robert Mark Hierons, John Howroyd, Lahcen Ouarbya, and Martin Ward. Consus: A lightweight program conditioner. *Journal of Systems and Software*, 77(3):241–262, 2005.
- Robert Hierons, Mark Harman, and Chris Fox. Branch-coverage testability transformation for unstructured programs. *The computer Journal*, 48(4):421–436, 2005.
- David Wendell Binkley and Mark Harman. Analysis and visualization of predicate dependence on formal parameters and global variables. *IEEE Transactions on Software Engineering*, 30(11):715–735, 2004.
- Robert Mark Hierons and Mark Harman. Testing conformance of a deterministic implementation against a non-deterministic specification. *Theoretical Computer Science*, 323(1-3):191–233, 2004.
- David Wendell Binkley and Mark Harman. A survey of empirical results on program slicing. *Advances in Computers*, 62:105–178, 2004.
- Chris Fox, Sebastian Danicic, Mark Harman, and Robert Mark Hierons. ConSIT: a fully automated conditioned program slicer. *Software—Practice and Experience*, 34:15–46, 2004.

Published online 26th November 2003.

Mark Harman, Lin Hu, Robert Mark Hierons, Joachim Wegener, Harmen Sthamer, Andr Baresel, and Marc Roper. Testability transformation. *IEEE Transactions on Software Engineering*, 30(1):3–16, January 2004.

Mark Harman, Lin Hu, Malcolm Munro, Xingyuan Zhang, David Wendell Binkley, Sebastian Danicic, Mohammed Daoudi, and Lahcen Ouarbya. Syntax-directed amorphous slicing. *Journal of Automated Software Engineering*, 11(1):27–61, January 2004.

Keith Brian Gallagher, Mark Harman, and Sebastian Danicic. Guaranteed inconsistency avoidance during software evolution. *Journal of Software Maintenance and Evolution*, 15(6):393–416, Nov/Dec 2003.

José Javier Dolado, Mark Harman, Mari Carmen Otero, and Lin Hu. An empirical investigation of the influence of a type of side effects on program comprehension. *IEEE Transactions on Software Engineering*, 29(7):665–670, 2003.

John Clark, José Javier Dolado, Mark Harman, Robert Mark Hierons, Bryan Jones, Mary Lumkin, Brian Mitchell, Spiros Mancoridis, Kearton Rees, Marc Roper, and Martin Shepherd. Reformulating software engineering as a search problem. *IEE Proceedings — Software*, 150(3):161–175, 2003.

Mark Harman, David Wendell Binkley, and Sebastian Danicic. Amorphous program slicing. *Journal of Systems and Software*, 68(1):45–64, October 2003.

Michael R. Laurence, Sebastian Danicic, Mark Harman, Rob Hierons, and John Howroyd. Equivalence of conservative, free, linear program schemas is decidable. *Theoretical Computer Science*, 290:831–862, January 2003.

Mark Harman, Malcolm Munro, Lin Hu, and Xingyuan Zhang. Source code analysis and manipulation. *Information and Software Technology*, 44(13):717–720, October 2002.

Robert Mark Hierons, Mark Harman, Chris Fox, Lahcen Ouarbya, and Mohammed Daoudi. Conditioned slicing supports partition testing. *Software Testing, Verification and Reliability*, 12:23–28, March 2002.

Mark Harman and Bryan F. Jones. The seminal workshop: Reformulating software engineering as a metaheuristic search problem. *Software Engineering Notes*, 26(6):62–66, November 2001.

Mark Harman and Robert Mark Hierons. An overview of program slicing. *Software Focus*, 2(3):85–92, 2001.

Mark Harman and Bryan F. Jones. Search based software engineering. *Information and Software Technology*, 43(14):833–839, December 2001.

Mark Harman and Bryan F. Jones. Software engineering using metaheuristic innovative algorithms. *Information and Software Technology*, 43(14):905–907, December 2001.

Robert Mark Hierons and Mark Harman. Testing against non-deterministic stream X-machines. *Formal Aspects of Computing*, 12:423–442, 2000.

Robert Mark Hierons, Mark Harman, and Sebastian Danicic. Using program slicing to

assist in the detection of equivalent mutants. *Software Testing, Verification and Reliability*, 9(4):233–262, 1999.

Mark Harman and Keith Brian Gallagher. Program slicing. *Information and Software Technology*, 40(11 and 12):577–581, November 1998.

Mark Harman and Sebastian Danicic. A new algorithm for slicing unstructured programs. *Journal of Software Maintenance and Evolution*, 10(6):415–441, 1998.

Mark Harman, Dan Simpson, and Sebastian Danicic. Slicing programs in the presence of errors. *Formal Aspects of Computing*, 8(4):490–497, 1996.

Sebastian Danicic, Mark Harman, and Yogasundary Sivagurunathan. A parallel algorithm for static program slicing. *Information Processing Letters*, 56(6):307–313, December 1995.

Mark Harman and Sebastian Danicic. Using program slicing to simplify testing. *Software Testing, Verification and Reliability*, 5(3):143–162, September 1995.

Journal Special Issues Edited (17)

Mark Harman and Mauro Pezze. Special issue of extended papers from ISSTA 2013. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 2015. to appear.

Mark Harman and Francisco Chicano. Search based software engineering (SBSE). *Journal of Systems and Software*, 103:266, 2015.

Mark Harman and Bogdan Korel. Editorial for special issue of STVR on software testing, verification, and validation - volume 1 (extended selected papers from ICST 2011). *Software Testing, Verification and Reliability*, 23(6):437, 2013.

Mark Harman, Yue Jia, and Mercedes G. Merayo. Special issue on mutation testing. *Software Testing, Verification and Reliability*, 2015. to appear.

Mark Harman and Afshin Mansouri. Special issue on search based software engineering (SBSE). *IEEE Transactions on Software Engineering*, 36(6), November – December 2010.

Giulio Antoniol, Massimiliano Di Penta, and Mark Harman. Special issue on search based software maintenance and evolution. *Journal of Software Maintenance and Evolution: Research and Practice*, 20(5), September – October 2008.

John Clark, Mark Harman, Phil McMinn, and Jeff Offutt. Special issue of extended papers from the TAIC PART 2007 and Mutation 2007. *Journal of Systems and Software*, 2009.

Mark Harman, Zheng Li, and Phil McMinn. Special issue of extended papers from the taic part 2006. *Software Testing, Verification and Reliability*, 2007.

Walter Gutjahr and Mark Harman. Focussed issue on search based software engineering. *Journal Computers and Operations Research*, 2006. To appear.

Mark Harman, Bogdan Korel, and Panos Linos. Special issue on best papers from ICSM 04. *IEEE Transactions on Software Engineering*, 31(10), October 2006.

Mark Harman, Bogdan Korel, Audris Mockus, and Martin Shepperd. Special issue on

maintenance and metrics. *Empirical Software Engineering*, 11(3), September 2006.

Mark Harman, Rainer Koschke, and Michael Van De Vanter. Special issue on source code analysis and manipulation. *Journal of Systems and Software*, 79(9), September 2006.

John Derrick, Mark Harman, and Robert Hierons. Special issue on formal approaches to software testing. *Formal Aspects of Computing*, 18(1), 2006.

Mark Harman, David Binkley, Paolo Tonella, and Liz Burd. Special issue of extended papers from SCAM'03 workshop. *Software Quality Journal*, 12(4), December 2004.

John Clark, Mark Harman, and Robert Hierons. Special issue of papers from UK Testing Workshop. *Software Testing, Verification and Reliability*, 14(3), September 2004.

Mark Harman, Malcolm Munro, Lin Hu, and Xingyuan Zhang. Special issue on source code analysis and manipulation. *Information and Software Technology*, 44(13), October 2002.

Mark Harman and Keith Gallagher. Special issue on program slicing. *Information and Software Technology*, 40(11 and 12), November 1998.

Keynotes and Invited Talks (38)

David Erb, Xinbo Gao, Mark Harman, Yue Jia, Ke Mao, Alexander Mols, Don Stewart, and Taijin Tei. Deploying search based software engineering with sapienz at facebook (keynote paper). In 10th *International Symposium on Search Based Software Engineering (SSBSE 2018)*, page 10, Montpellier, France, September 8th-10th 2018. To Appear.

Mark Harman and Peter O'Hearn. From start-ups to scale-ups: Open problems, challenges and myths in static and dynamic program analysis for testing and verification (keynote paper). In 18th *IEEE International Working Conference on Source Code Analysis and Manipulation (SCAM 2018)*, page 10, Madrid, Spain, 2018. To Appear.

Mark Harman. We need a formal semantics for testability transformation. In 16th *International Conference on Software Engineering and Formal Methods (SEFM 2018)*, Toulouse, France, 2018.

Mark Harman, Yue Jia, and Yuanyuan Zhang. Achievements, open problems and challenges for search based software testing (keynote). In 8th *IEEE International Conference on Software Testing, Verification and Validation (ICST 2015)*, Graz, Austria, April 2015.

Mark Harman, Yue Jia, Jens Krinke, Bill Langdon, Justyna Petke, and Yuanyuan Zhang. Search based software engineering for software product line engineering: a survey and directions for future work (keynote paper). In 18th *International Software Product Line Conference (SPLC 14)*, pages 5–18, Florence, Italy, September 2014.

Mark Harman. Applications of sbse to software testing and self adaptive systems (keynote). In *Quality Assurance for Self-adaptive, Self-organising Systems (QA4SASO 2014)*, London, UK, September 2014.

Mark Harman, Yue Jia, William B. Langdon, Justyna Petke, Iman Hemati Moghadam, Shin Yoo, and Fan Wu. Genetic improvement for adaptive software engineering (keynote).

In 9th *International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2014)*, pages 1–4, New York, NY, USA, 2014. ACM.

Mark Harman, William B. Langdon, and Westley Weimer. Genetic programming for reverse engineering (keynote paper). In Rocco Oliveto and Romain Robbes, editors, 20th *Working Conference on Reverse Engineering (WCRE 2013)*, Koblenz, Germany, 14-17 October 2013. IEEE.

Mark Harman. Software engineering: An ideal set of challenges for evolutionary computation (keynote). In 22nd *Genetic and Evolutionary Computation Conference (GECCO 2013)*, Amsterdam, Netherlands, 6th - 10th July 2013.

Mark Harman, John Clark, and Mel Ó Cinnéidie. Dynamic adaptive search based software engineering needs fast approximate metrics (keynote paper). In 4th *International Workshop on Emerging Trends in Software Metrics (WeTSOM 2013)*, San Francisco, USA, May 2013.

Mark Harman, William B. Langdon, Yue Jia, David R. White, Andrea Arcuri, and John A. Clark. The GISMOE challenge: Constructing the pareto program surface using genetic programming to find better programs (keynote paper). In 27th *IEEE/ACM International Conference on Automated Software Engineering (ASE 2012)*, pages 1–14, Essen, Germany, September 2012.

Mark Harman. Dynamic adaptive search based software engineering (keynote paper). In 6th *IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM 2012)*, Lund, Sweden, 2012.

Mark Harman. Search based software engineering (keynote). In 6th *IEEE International Symposium on Theoretical Aspects of Software Engineering (TASE 2012)*, Beijing, China, 2012.

Mark Harman. The role of artificial intelligence in software engineering (keynote paper). In 1st *International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE 2012)*, Zurich, Switzerland, 2012.

Mark Harman. The law of tendency to executability and its implications (keynote). In 11th *International Conference on Quality Software (QSIC 2010)*, Madrid, Spain, 2011.

Mark Harman. Refactoring as testability transformation (keynote paper). In 1st *Refactoring and Testing Workshop (RefTest 2011)*, Berlin, Germany, 2011.

Mark Harman. Making the case for MORTO: Multi objective regression test optimization (invited position paper). In 1st *International Workshop on Regression Testing (Regression 2011)*, Berlin, Germany, 2011.

Mark Harman. How sbse can support construction and analysis of predictive models (keynote). In 6th *International Conference on Predictive Models in Software Engineering*, Timisoara, Romania, 2010.

Mark Harman. Why source code analysis and manipulation will always be important (keynote). In 10th *IEEE International Working Conference on Source Code Analysis and Manipulation*, Timisoara, Romania, 2010.

Mark Harman, Yue Jia, and William B. Langdon. How higher order mutation helps mutation testing (keynote). In 5th *International Workshop on Mutation Analysis (Mutation 2010)*, Paris, France, 2010.

- Mark Harman. Search based software engineering (keynote). In *13th International Conference on Fundamental Approaches to Software Engineering (FASE 2010)*, Paphos, Cyprus, 2010.
- Mark Harman. An evolutionary approach to evolution: Applications of sbse to software analysis, refactoring, evolution and re-engineering (keynote). In *8th Belgian-Netherlands software eVOLution seminar (BENEVOL 2009)*, Universite Catholique de Louvain, Belgium, 2009.
- Mark Harman. The sbse approach to automated optimization of verification and testing (keynote). In *5th IBM Haifa Verification Conference (HVC 09)*, 2009.
- Mark Harman. Search based optimization is machine assisted reasoning? (invited talk). In *Reasoning '09*, Microsoft Research, Cambridge, UK, 2009.
- Mark Harman. Recent trends in metaheuristic optimization for search based software engineering (keynote). In *Metaheuristics and bioinspired algorithms*, Malaga, Spain, 2009.
- Mark Harman. Open problems in testability transformation (keynote). In *1st International Workshop on Search Based Testing (SBT 2008)*, Lillehammer, Norway, 2008.
- Mark Harman. The current state and future of search based software engineering (invited paper). In *29th International Conference on Software Engineering (ICSE 2007), Future of Software Engineering (FoSE)*, Minneapolis, USA, 2007.
- Mark Harman. Automated test data generation using search based software engineering (keynote). In *2nd Workshop on Automation of Software Test (AST 07) at the 29th International Conference on Software Engineering (ICSE 2007)*, Minneapolis, USA, 2007.
- Mark Harman. Search based software engineering for program comprehension (keynote). In *15th International Conference on Program Comprehension (ICPC 2007)*, Banff, Canada, 2007.
- Mark Harman. The use of measurement in search based software engineering (keynote). In *International Conference on Software Process and Product Measurement (Mensura 2006)*, Cádiz, Spain, 2006.
- Mark Harman. Search based testing (keynote). In *Software & Systems Quality Conference (SQS-UK 2006, industry conference)*, London, UK, 2006.
- Mark Harman. Search based software engineering (keynote). In *Workshop on Computational Science in Software Engineering (CSSE 2006), LNCS 3994.*, pages 740–747, Reading, UK, 2006.
- Mark Harman. Search based testing (keynote). In *TestExpo industrial conference*, London, UK, 2006.
- Mark Harman. Search-based software engineering for maintenance and reengineering (keynote). In *10th European Conference on Software Maintenance and Reengineering (CSMR 2006)*, Bari, Italy March 22-24, 2006.
- Mark Harman and Joachim Wegener. Search based testing (keynote). In *6th Metaheuristics International Conference (MIC 2005)*, Vienna, Austria, August 2005.

Mark Harman. Side-effects considered harmful (keynote). In *14th Annual Psychology of Programming Workshop (PPIG)*, London, April 2002.

Sebastian Danicic and Mark Harman. A simultaneous slicing theory and derived program slicer (keynote). In *4th RIMS Workshop in Computing*, Kyoto University, Kyoto, Japan, July 1996.

Mark Harman and Sebastian Danicic (invited talk). Towards the measurement of objects. In Martin Shepperd, editor, *1st Bournemouth Metrics Workshop*, Bournemouth University, UK, April 1996.

Book Chapters (3)

Filomena Ferrucci, Mark Harman, and Federica Sarro. Search based software project management. In Guenther Ruhe and Claes Wohlin, editors, *Software Project Management in a Changing World*. Springer, 2014. To appear.

Mark Harman, Phil McMinn, Jerffeson Teixeira de Souza, and Shin Yoo. Search based software engineering: Techniques, taxonomy, tutorial. In Bertrand Meyer and Martin Nordio, editors, *Empirical software engineering and verification: LASER 2009-2010*. Springer, 2012. to appear.

Mark Harman, André Baresel, David Binkley, Robert Hierons, Lin Hu, Bogdan Korel, Philip McMinn, and Marc Roper. Testability transformation — program transformation to improve testability. In Robert Hierons, Jonathan Bowen, and Mark Harman, editors, *Formal Methods and Testing LNCS 4949*, chapter 11. Springer, 2008.

Fully Refereed Conference Papers (192)

Mark Harman. Search based software testing for android (keynote). In *10th IEEE/ACM International Workshop on Search-Based Software Testing, SBST@ICSE 2017, Buenos Aires, Argentina, May 22-23, 2017*, page 2, 2017.

Thierry Titchou Chekam, Mike Papadakis, Yves Le Traon, and Mark Harman. An empirical study on mutation, statement and branch coverage fault revelation that avoids the unreliable clean program assumption. In *Proceedings of the 39th International Conference on Software Engineering, ICSE 2017, Buenos Aires, Argentina, May 20-28, 2017*, pages 597–608, 2017.

Ke Mao, Mark Harman, and Yue Jia. Crowd intelligence enhances automated mobile testing. In *Proceedings of the 32Nd IEEE/ACM International Conference on Automated Software Engineering, ASE 2017*, pages 16–26, 2017.

Matheus Paixao, Jens Krinke, DongGyun Han, and Mark Harman. CROP: Linking code reviews to source code changes. In *International Conference on Mining Software Repositories (MSR 2018)*, 2018.

Matheus Paixão, Jens Krinke, DongGyun Han, Chaiyong Ragkhitwetsagul, and Mark Harman. Are developers aware of the architectural impact of their changes? In *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering, ASE 2017, Urbana, IL, USA, October 30 - November 03, 2017*, pages 95–105, 2017.

Nicolas E. Gold, David Binkley, Mark Harman, Syed S. Islam, Jens Krinke, and Shin Yoo.

Generalized observational slicing for tree-represented modelling languages. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, ESEC/FSE 2017, Paderborn, Germany, September 4-8, 2017*, pages 547–558, 2017.

William Martin, Federica Sarro, and Mark Harman. Causal impact analysis for app releases in Google Play. In *24th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE 2016)*, pages 435–446, Seattle, WA, USA, November 2016.

Fan Wu, Mark Harman, Yue Jia, and Jens Krinke. HOMI: searching higher order mutants for software improvement. In *Search Based Software Engineering - 8th International Symposium, SSBSE 2016, Raleigh, NC, USA, October 8-10, 2016, Proceedings*, pages 18–33, 2016.

William B. Langdon, David R. White, Mark Harman, Yue Jia, and Justyna Petke. Api-constrained genetic improvement. In *Search Based Software Engineering - 8th International Symposium, SSBSE 2016, Raleigh, NC, USA, October 8-10, 2016, Proceedings*, pages 224–230, 2016.

Yibiao Yang, Mark Harman, Jens Krinke, Syed S. Islam, David Binkley, Yuming Zhou, and Baowen Xu. An empirical study on dependence clusters for effort-aware fault-proneness prediction. In *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering, ASE 2016, Singapore, September 3-7, 2016*, pages 296–307, 2016.

Federica Sarro, Alessio Petrozziello, and Mark Harman. Multi-objective software effort estimation. In *Proceedings of the 38th International Conference on Software Engineering, ICSE 2016, Austin, TX, USA, May 14-22, 2016*, pages 619–630, 2016.

Christopher Henard, Mike Papadakis, Mark Harman, Yue Jia, and Yves Le Traon. Comparing white-box and black-box test prioritization. In *Proceedings of the 38th International Conference on Software Engineering, ICSE 2016, Austin, TX, USA, May 14-22, 2016*, pages 523–534, 2016.

Afnan A. Al-Subaihini, Federica Sarro, Sue Black, L. Capra, Mark Harman, Yue Jia, and Yuanyuan Zhang. Clustering mobile apps based on mined textual features. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2016, Ciudad Real, Spain, September 8-9, 2016*, pages 38:1–38:10, 2016.

Ke Mao, Mark Harman, and Yue Jia. Sapienz: Multi-objective automated testing for Android applications. In *ISSTA*, pages 94–105, 2016.

David Bowes, Tracy Hall, Mark Harman, Yue Jia, Federica Sarro, and Fan Wu. Mutation-aware fault prediction. In *ISSTA*, pages 330–341, 2016.

Mike Papadakis, Christopher Henard, Mark Harman, Yue Jia, and Yves Le Traon. Threats to construct validity of mutation-based test assessment. In *ISSTA*, pages 354–365, 2016.

Gunel Jahangirova, David Clark, Mark Harman, and Paolo Tonella. Test oracle assessment and improvement. In *ISSTA*, pages 247–258, 2016.

Federica Sarro, Afnan AlSubaihini, Mark Harman, Yue Jia, William Martin, and Yuanyuan Zhang. Feature lifecycles as they spread, migrate, remain and die in app stores. In *Requirements Engineering (RE'15)*, Ottawa, Canada, August 2015. To appear.

Christopher Henard, Mike Papadakis, Mark Harman, and Yves Le Traon. Combining multi-objective search and constraint solving for configuring large software product lines. In *37th International Conference on Software Engineering (ICSE 2015)*, Florence, Italy, 2015. To appear.

Yue Jia, Myra B. Cohen, Mark Harman, and Justyna Petke. Learning combinatorial interaction test generation strategies using hyperheuristic search. In *37th International Conference on Software Engineering (ICSE 2015)*, Florence, Italy, 2015. To appear.

Mike Papadakis, Yue Jia, Mark Harman, and Yves Le Traon. Trivial compiler equivalence: A large scale empirical study of a simple, fast and effective equivalent mutant detection technique. In *37th International Conference on Software Engineering (ICSE 2015)*, Florence, Italy, 2015. To appear.

William Martin, Mark Harman, Yue Jia, Federica Sarro, and Yuanyuan Zhang. The app sampling problem for app store mining. In *Mining Software Repositories (MSR'15)*, Florence, Italy, May 2015.

Geoffrey Neumann, Mark Harman, and Simon M. Poulding. Transformed vargha-delaney effect size. In *7th International Symposium on Search-Based Software Engineering (SSBSE)*, pages 318–324, Bergamo, Italy, September 2015.

Yi Bian, Serkan Kirbas, Mark Harman, Yue Jia, and Zheng Li. Regression test case prioritisation for guava. In *7th International Symposium on Search-Based Software Engineering (SSBSE)— Challenge Track*, pages 221–227, Bergamo, Italy, September 2015.

Alexandru Marginean, Earl T. Barr, Mark Harman, and Yue Jia. Automated transplantation of call graph and layout features into kate. In *7th International Symposium on Search-Based Software Engineering (SSBSE)— Challenge Track*, pages 262–268, Bergamo, Italy, September 2015.

Yue Jia, Mark Harman, William B. Langdon, and Alexandru Marginean. Grow and serve: Growing django citation services using SBSE. In *7th International Symposium on Search-Based Software Engineering (SSBSE)— Challenge Track*, pages 269–275, Bergamo, Italy, September 2015.

Matheus Paixão, Mark Harman, and Yuanyuan Zhang. Multi-objective module clustering for kate. In *7th International Symposium on Search-Based Software Engineering (SSBSE)— Challenge Track*, pages 282–288, Bergamo, Italy, September 2015.

Lingbo Li, Mark Harman, Fan Wu, and Yuanyuan Zhang. Sselector: Search based component selection for budget hardware. In *7th International Symposium on Search-Based Software Engineering (SSBSE)— Challenge Track*, pages 289–294, Bergamo, Italy, September 2015.

Yuanyuan Zhang, Mark Harman, Yue Jia, and Federica Sarro. Inferring test models from kate's bug reports using multi-objective search. In *7th International Symposium on Search-Based Software Engineering (SSBSE)— Challenge Track*, pages 301–307, Bergamo, Italy, September 2015.

Earl T. Barr, Mark Harman, Yue Jia, Alexandru Marginean, and Justyna Petke. Automated software transplantation. In *Proceedings of the 2015 International Symposium on Software Testing and Analysis, ISSTA 2015, Baltimore, MD, USA, July 12-17, 2015*, pages 257–269, 2015.

Michael G. Epitropakis, Shin Yoo, Mark Harman, and Edmund K. Burke. Empirical evaluation of pareto efficient multi-objective regression test case prioritisation. In *Proceedings of the 2015 International Symposium on Software Testing and Analysis, ISSTA 2015, Baltimore, MD, USA, July 12-17, 2015*, pages 234–245, 2015.

Mark Harman and Justyna Petke. GI4GI: improving genetic improvement fitness functions. In *Genetic and Evolutionary Computation Conference, GECCO 2015, Madrid, Spain, July 11-15, 2015, Companion Material Proceedings*, pages 793–794, 2015.

Yue Jia, Fan Wu, Mark Harman, and Jens Krinke. Genetic improvement using higher order mutation. In *Genetic and Evolutionary Computation Conference, GECCO 2015, Madrid, Spain, July 11-15, 2015, Companion Material Proceedings*, pages 803–804, 2015.

William B. Langdon and Mark Harman. Grow and graft a better CUDA pknotsrg for RNA pseudoknot free energy calculation. In *Genetic and Evolutionary Computation Conference, GECCO 2015, Madrid, Spain, July 11-15, 2015, Companion Material Proceedings*, pages 805–810, 2015.

Fan Wu, Mark Harman, Yue Jia, Jens Krinke, and Westley Weimer. Deep parameter optimisation. In *Genetic and evolutionary computation conference (GECCO 2015)*, pages 1375–1382, Madrid, Spain, July 2015.

William B. Langdon, Brian Lam, Justyna Petke, and Mark Harman. Improving cuda dna analysis software with genetic programming. In *Genetic and evolutionary computation conference (GECCO 2015)*, pages 1063–1070, Madrid, Spain, July 2015.

Bobby Bruce, Justyna Petke, and Mark Harman. Reducing energy consumption using genetic improvement. In *Genetic and evolutionary computation conference (GECCO 2015)*, pages 1327–1334, Madrid, Spain, July 2015.

Mark Harman, Yue Jia, and Yuanyuan Zhang. Achievements, open problems and challenges for search based software testing. In *8th IEEE International Conference on Software Testing, Verification and Validation (ICST 2015)*, pages 1–12, Graz, Austria, April 2015.

Ke Mao, Ye Yang, Qing Wang, Yue Jia, and Mark Harman. Developer recommendation for crowdsourced software development tasks. In *2015 IEEE Symposium on Service-Oriented System Engineering, SOSE 2015, San Francisco Bay, CA, USA, March 30 - April 3, 2015*, pages 347–356, 2015.

Jay Nanavati, Fan Wu, Mark Harman, Yue Jia, and Jens Krinke. Mutation testing of memory-related operators. In *Eighth IEEE International Conference on Software Testing, Verification and Validation, ICST 2015 Workshops, Graz, Austria, April 13-17, 2015*, pages 1–10, 2015.

Mark Harman, Yue Jia, Pedro Reales Mateo, and Macario Polo. Angels and monsters: an empirical investigation of potential test effectiveness and efficiency improvement from strongly subsuming higher order mutation. In *ACM/IEEE International Conference on Automated Software Engineering (ASE '14)*, pages 397–408. ACM, 2014.

Mark Harman, Yue Jia, Jens Krinke, Bill Langdon, Justyna Petke, and Yuanyuan Zhang. Search based software engineering for software product line engineering: a survey and directions for future work. In *18th International Software Product Line Conference (SPLC 14)*, pages 5–18, Florence, Italy, September 2014.

David Binkley, Nicolas Gold, Mark Harman, Syed Islam, Jens Krinke, and Shin Yoo. ORBS: Language-independent program slicing. In *22nd ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE 2014)*, pages 109–120, Hong Kong, China, November 2014.

Earl T. Barr, Yuriy Brun, Premkumar Devanbu, Mark Harman, and Federica Sarro. The plastic surgery hypothesis. In *22nd ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE 2014)*, pages 306–317, Hong Kong, China, November 2014.

Mark Harman, Yue Jia, William B. Langdon, Justyna Petke, Iman Hemati Moghadam, Shin Yoo, and Fan Wu. Genetic improvement for adaptive software engineering. In *9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2014)*, pages 1–4, New York, NY, USA, 2014. ACM.

Mark Harman, Syed Islam, Yue Jia, Leandro Minku, Federica Sarro, and Komsan Srivisut. Less is more: Temporal fault predictive performance over multiple hadoop releases. In *6th Symposium on Search Based Software Engineering (SSBSE 2014)*, pages 240–246, Fortaleza, Brazil, August 2014. Springer LNCS.

Mark Harman, William B. Langdon, and Yue Jia. Babel pidgin: SBSE can grow and graft entirely new functionality into a real world system (winner of SBSE Challenge 2014 at SSBSE 2014). In *6th Symposium on Search Based Software Engineering (SSBSE 2014)*, pages 247–252, Fortaleza, Brazil, August 2014. Springer LNCS.

Nadia Alshahwan and Mark Harman. Coverage and fault detection of the output-uniqueness test selection criteria. In *International Symposium on Software Testing and Analysis (ISTA '14)*, pages 181–192. ACM, 2014.

William B. Langdon, Marc Modat, Justyna Petke, and Mark Harman. Improving 3D medical image registration CUDA software with genetic programming. In *Genetic and Evolutionary Computation Conference (GECCO '14), 2014*, pages 951–958. ACM, 2014.

Lingbo Li, Mark Harman, Emmanuel Letier, and Yuanyuan Zhang. Robust next release problem: handling uncertainty during optimization. In *Genetic and Evolutionary Computation Conference (GECCO '14)*, pages 1247–1254. ACM, 2014.

Mark Harman, Xiangjuan Yao, and Yue Jia. A study of equivalent and stubborn mutation operators using human analysis of equivalence. In *36th International Conference on Software Engineering (ICSE 2014)*, pages 919–930, Hyderabad, India, June 2014.

Kelly Androutsopoulos, David Clark, Haitao Dan, Mark Harman, and Robert Hierons. An analysis of the relationship between conditional entropy and failed error propagation in software testing. In *36th International Conference on Software Engineering (ICSE 2014)*, pages 573–583, Hyderabad, India, June 2014.

Justyna Petke, Mark Harman, William B. Langdon, and Westley Weimer. Using genetic improvement & code transplants to specialise a C++ program to a problem class (silver medal winner at GECCO 2014 human competitive results competition — The HUMIES). In *17th European Conference on Genetic Programming (EuroGP)*, pages 132–143, Granada, Spain, April 2014.

William B. Langdon and Mark Harman. Genetically improved CUDA C++ software.

In 17th *European Conference on Genetic Programming (EuroGP)*, pages 84–95, Granada, Spain, April 2014.

Mark Harman, William B. Langdon, and Westley Weimer. Genetic programming for reverse engineering. In Rocco Oliveto and Romain Robbes, editors, *20th Working Conference on Reverse Engineering (WCRE 2013)*, Koblenz, Germany, 14-17 October 2013. IEEE.

Tiantian Wang, Mark Harman, Yue Jia, and Jens Krinke. Searching for better configurations: a rigorous approach to clone evaluation. In *European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE'13*, pages 455–465, Saint Petersburg, Russian Federation, August 2013. ACM.

Justyna Petke, Myra B. Cohen, Mark Harman, and Shin Yoo. Efficiency and early fault detection with lower and higher strength combinatorial interaction testing. In *European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE'13*, pages 26–36, Saint Petersburg, Russian Federation, August 2013. ACM.

Bertrand Meyer, Harald Gall, Mark Harman, and Giancarlo Succi. Empirical answers to fundamental software engineering problems (panel paper). In *European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE'13*, pages 14–18, Saint Petersburg, Russian Federation, August 2013. ACM.

Xiaoyuan Xie, Fei-Ching Kuo, Tsong Yueh Chen, Shin Yoo, and Mark Harman. Provably optimal and human-competitive results in SBSE for spectrum based fault localisation. In *Proceedings of the 5th International Symposium on Search Based Software Engineering (SSBSE '13)*, volume 8084, pages 224–238, St. Petersburg, Russia, 24-26 August 2013. Springer.

Justyna Petke, William B. Langdon, and Mark Harman. Applying genetic improvement to minisat. In *Proceedings of the 5th International Symposium on Search Based Software Engineering (SSBSE '13)*, volume 8084, pages 257–262, St. Petersburg, Russia, 24-26 August 2013. Springer.

Filomena Ferrucci, Mark Harman, Jian Ren, and Federica Sarro. Not going to take this anymore: Multi-objective overtime planning for software engineering projects. In 35th *ACM/IEEE International Conference on Software Engineering (ICSE 2013)*, San Francisco, USA, 2013. to appear.

Mustafa Bozkurt and Mark Harman. Optimised realistic test input generation using web services. In 4th *International Symposium on Search Based Software Engineering (SSBSE 2012)*, pages 105–120, Riva del Garda, Italy, 2012.

Mel O Cinneide, Laurie Tratt, Mark Harman, Steve Counsell, and Iman Hemati Moghadam. Experimental assessment of software metrics using automated refactoring. In 6th *IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM 2012)*, pages 49–58, Lund, Sweden, September 2012.

Mark Harman, Edmund Burke, John A. Clark, and Xin Yao. Dynamic adaptive search based software engineering. In 6th *IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM 2012)*, pages 1–8, Lund, Sweden, September 2012.

Mark Harman, William B. Langdon, Yue Jia, David R. White, Andrea Arcuri, and John A. Clark. The GISMOE challenge: Constructing the pareto program surface using genetic programming to find better programs (keynote paper). In *27th IEEE/ACM International Conference on Automated Software Engineering (ASE 2012)*, Essen, Germany, September 2012.

Mark Harman. Search based software engineering. In *6th IEEE International Symposium on Theoretical Aspects of Software Engineering (TASE 2012)*, Beijing, China, 2012.

Nadia Alshahwan and Mark Harman. State aware test case regeneration for improving web application test suite coverage and fault detection. In *International Symposium on Software Testing and Analysis (ISSTA 2012)*, pages 45–55, Minneapolis, MN, USA, 2012.

Nadia Alshahwan and Mark Harman. Augmenting test suites effectiveness by increasing output diversity. In *34th International Conference on Software Engineering New Ideas and Emerging Results Track (NIER 2012)*, pages 1345–1348, 2012.

Mark Harman. The role of artificial intelligence in software engineering. In *1st International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE 2012)*, Zurich, Switzerland, 2012.

Mark Harman, Yue Jia, and Yuanyuan Zhang. App store mining and analysis: MSR for App Stores. In *Proceedings of the 9th Working Conference on Mining Software Repositories (MSR 2012)*, Zurich, Switzerland, 2-3 June 2012.

Paolo Tonella, Alessandro Marchetto, Duy Cu Nguyen, Yue Jia, Kiran Lakhotia, and Mark Harman. Finding the optimal balance between over and under approximation of models inferred from execution logs. In *5th International Conference on Software Testing, Verification and Validation (ICST 2012)*, pages 21–30, Montreal, Canada, 2012.

Nadia Alshahwan, Mark Harman, Alessandro Marchetto, Roberto Tiella, and Paolo Tonella. Crawlability metrics for web applications. In *5th International Conference on Software Testing, Verification and Validation (ICST 2012)*, pages 21–30, Montreal, Canada, 2012.

Tanja E. J. Vos, Paolo Tonella, Joachim Wegener, Mark Harman, Wishnu Prasetya, Elisa Puoskari, and Yarden Nir-Buchbinder. Future internet testing with FITTEST. In *15th IEEE European Conference on Software Maintenance and Reengineering (CSMR 2011)*, pages 355–358, 2011.

Arthur Baars, Mark Harman, Youssef Hassoun, Kiran Lakhotia, Phil McMinn, Paolo Tonella, and Tanja Vos. Symbolic search-based testing. In *26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011)*, pages 53 – 62, Lawrence, Kansas, USA, 6th - 10th November 2011.

Nadia Alshahwan and Mark Harman. Automated web application testing using search based software engineering. In *26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011)*, pages 3 – 12, Lawrence, Kansas, USA, 6th - 10th November 2011.

Shin Yoo, Mark Harman, and Shmuel Ur. Highly scalable multi-objective test suite minimisation using graphics cards. In *3rd International Symposium on Search based Software Engineering (SSBSE 2011)*, pages 219–236, 10th - 12th September 2011. LNCS Volume 6956.

Jian Ren, Mark Harman, and Massimiliano Di Penta. Cooperative co-evolutionary optimization on software project staff assignments and job scheduling. In *3rd International Symposium on Search based Software Engineering (SSBSE 2011)*, pages 127–141, 10th - 12th September 2011. LNCS Volume 6956.

Shin Yoo, Robert Nilsson, and Mark Harman. Faster fault finding at Google using multi objective regression test optimisation. In *8th European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE '11)*, Szeged, Hungary, September 5th - 9th 2011. Industry Track.

Mark Harman, Yue Jia, and Bill Langdon. Strong higher order mutation-based test data generation. In *8th European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE '11)*, pages 212–222, New York, NY, USA, September 5th - 9th 2011. ACM.

Nicolas Gold, Jens Krinke, Mark Harman, and David Binkley. Cloning in Max/MSP patches. In *Proceedings of International Computer Music Conference (ICMC 2011)*, Huddersfield, UK, 31st July - 5th August 2011.

Jungsup Oh, Mark Harman, and Shin Yoo. Transition coverage testing for Simulink/Stateflow models using messy genetic algorithms. In *Genetic Algorithms and Evolutionary Computation Conference (GECCO 2011)*, pages 1851–1858, Dublin, Ireland, 2011.

Kelly Androutsopoulos, David Binkley, David Clark, Nicolas Gold, Mark Harman, Kevin Lano, and Zheng Li. Model projection: Simplifying models in response to restricting the environment. In *33rd International Conference on Software Engineering (ICSE '11)*, pages 291–300, New York, NY, USA, 2011. ACM.

Mark Harman. Making the case for MORTO: Multi objective regression test optimization. In *1st International Workshop on Regression Testing (Regression 2011)*, Berlin, Germany, March 2011.

Mark Harman. Refactoring as testability transformation. In *Refactoring and Testing Workshop (RefTest 2011)*, Berlin, Germany, March 2011.

Kiran Lakhota, Nikolai Tillmann, Mark Harman, and Jonathan de Halleux. FloPSy — Search-based floating point constraint solving for symbolic execution. In *22nd IFIP International Conference on Testing Software and Systems (ICTSS 2010)*, pages 142–157, Natal, Brazil, November 2010. LNCS Volume 6435.

Kiran Lakhota, Mark Harman, and Hamilton Gross. AUSTIN: A tool for search based software testing for the C language and its evaluation on deployed automotive systems. In *2nd International Symposium on Search Based Software Engineering (SSBSE 2010)*, pages 101 – 110, Benevento, Italy, September 2010.

Yuanyuan Zhang and Mark Harman. Search based optimization of requirements interaction management. In *2nd International Symposium on Search Based Software Engineering (SSBSE 2010)*, pages 47 – 56, Benevento, Italy, September 2010.

Philip McMinn, Mark Stevenson, and Mark Harman. Reducing qualitative human oracle costs associated with automatically generated test data. In *1st International Workshop on Software Test Output Validation (STOV 2010)*, pages 1–4, Trento, Italy, July 2010.

Mark Harman. The relationship between search based software engineering and predictive

modeling. In *6th International Conference on Predictive Models in Software Engineering*, page Article Number 1, Timisoara, Romania, 2010. keynote paper.

Mark Harman. Why source code analysis and manipulation will always be important. In *10th IEEE International Working Conference on Source Code Analysis and Manipulation*, pages 7–19, Timisoara, Romania, 2010. keynote paper.

William B. Langdon and Mark Harman. Evolving `gzip` matches kernel from an nVidia CUDA template. In *IEEE World Congress on Computational Intelligence (WCCI 2010)*, pages 2326–2383, 2010.

Yuanyuan Zhang, Enrique Alba, Juan J. Durillo, Sigrid Eldh, and Mark Harman. Today/future importance analysis. In *ACM Genetic and Evolutionary Computation Conference (GECCO 2010)*, pages 1357–1364, Portland Oregon, USA, 7th–11th July 2010.

Syed S. Islam, Jens Krinke, David Binkley, and Mark Harman. Coherent dependence clusters. In *9th ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering (PASTE 2010)*, pages 53–60, Toronto, Canada, 5th–6th June 2010.

Mark Harman, Yue Jia, and William B. Langdon. A manifesto for higher order mutation testing. In *5th International Workshop on Mutation Analysis (Mutation 2010)*, Paris, France, 2010.

Ruilian Zhao, Mark Harman, and Zheng Li. Empirical study on the efficiency of search based test generation for EFSM models. In *3rd International Workshop on Search-Based Software Testing (SBST 2010)*, Paris, France, April 2010.

Mark Harman, Sung Gon Kim, Kiran Lakhotia, Phil McMinn, and Shin Yoo. Optimizing for the number of tests generated in search based test data generation with an application to the oracle cost problem. In *3rd International Workshop on Search-Based Software Testing (SBST 2010)*, Paris, France, April 2010.

Mark Harman. Why the virtual nature of software makes it ideal for search based optimization. In *13th International Conference on Fundamental Approaches to Software Engineering (FASE 2010)*, pages 1–12, Paphos, Cyprus, March 2010.

Haoying Connie Bao, Nicolas Gold, and Mark Harman. Maintaining ws-bpel workflows using aspects. In *Business Process and Services Computing (BPSC 2009)*, Leipzig, Germany, 23rd–25th March 2009.

Nadia Alshahwan, Mark Harman, Alessandro Marchetto, and Paolo Tonella. Improving web application testing using testability measures. In *11th IEEE International Symposium on Web Systems Evolution (WSE 2009)*, pages 49–58, Edmonton, Alberta, Canada, 25th–26th September 2009.

Kelly Androutsopoulos, Nicolas Gold, Mark Harman, Zheng Li, and Laurie Tratt. A theoretical and empirical study of EFSM dependence. In *25th IEEE International Conference on Software Maintenance (ICSM 2009)*, pages 287–296, Edmonton, Alberta, Canada, 23rd–26th September 2009.

David Binkley and Mark Harman. Identifying ‘linchpin vertices’ that cause large dependence clusters. In *9th International Working Conference on Source Code Analysis and Manipulation (SCAM’09)*, pages 89–98, Edmonton, Alberta, Canada, 20th–21st September 2009.

Kiran Lakhotia, Phil McMinn, and Mark Harman. Automated test data generation for coverage: Haven't we solved this problem yet? In *4th Testing Academia and Industry Conference — Practice And Research Techniques (TAIC PART'09)*, pages 95–104, Windsor, UK, 4th–6th September 2009.

William B. Langdon, Mark Harman, and Yue Jia. Multi objective mutation testing with genetic programming. In *4th Testing Academia and Industry Conference — Practice And Research Techniques (TAIC PART'09)*, pages 21–29, Windsor, UK, 4th–6th September 2009.

Shin Yoo, Mark Harman, Paolo Tonella, and Angelo Susi. Clustering test cases to achieve effective and scalable prioritisation incorporating expert knowledge. In *ACM International Conference on Software Testing and Analysis (ISSTA 09)*, pages 201–212, Chicago, Illinois, USA, 19th – 23rd July 2009.

Giulio Antoniol, Stefan Gueorguiev, and Mark Harman. Software project planning for robustness and completion time in the presence of uncertainty using multi objective search based software engineering ('best at GECCO' award winner). In *ACM Genetic and Evolutionary Computation Conference (GECCO 2009)*, pages 1673–1680, Montreal, Canada, 8th – 12th July 2009.

Mark Harman, Jens Krinke, Jian Ren, and Shin Yoo. Search based data sensitivity analysis applied to requirement engineering. In *ACM Genetic and Evolutionary Computation Conference (GECCO 2009)*, pages 1681–1688, Montreal, Canada, 8th – 12th July 2009.

Shin Yoo, Mark Harman, and Shmuel Ur. Measuring an improving latency to avoid test suite wear out (best paper award winner). In *2nd IEEE International Workshop on Search-Based Software Testing (SBST 09)*, pages 101–110, Denver, Colorado, 1st April 2009.

Cu Nguyen, Anna Perini, Paolo Tonella, Simon Miles, Mark Harman, and Michael Luck. Evolutionary testing of autonomous software agents. In *8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pages 521–528, Budapest, Hungary, May 2009.

Kelly Androutsopoulos, David Clark, Mark Harman, Zheng Li, and Laurie Tratt. Control dependence for extended finite state machines (best theory paper award winner). In *Fundamental Approaches to Software Engineering (FASE '09)*, volume 5503, pages 216–230, York, UK, March 2009. Springer LNCS.

Mark Harman, Fayezin Islam, Tao Xie, and Stefan Wappler. Automated test data generation for aspect-oriented programs. In *8th International Conference on Aspect-Oriented Software Development (AOSD '09)*, pages 185–196, Charlottesville, Virginia, USA, March 2009.

Kiran Lakhotia, Mark Harman, and Phil McMinn. Handling dynamic data structures in search based testing. In *GECCO 2008 Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 1759 – 1766, Atlanta, USA, July 2008. ACM Press.

Tao Jiang, Mark Harman, and Youssef Hassoun. Analysis of procedure splitability. In *15th Working Conference on Reverse Engineering (WCRE'08)*, pages 247–256, Antwerp, Belgium, October 2008.

David Binkley, Nicolas Gold, Mark Harman, Zheng Li, Kiarash Mahdavi, and Joachim Wegener. Dependence anti patterns. In *4th International ERCIM Workshop on Software Evolution and Evolvability (Evol'08)*, pages 25–34, L'Aquila, Italy, September 2008.

Yue Jia and Mark Harman. Milu: A customizable, runtime-optimized higher order mutation testing tool for the full C language. In *3rd Testing Academia and Industry Conference - Practice and Research Techniques (TAIC PART'08)*, pages 94–98, Windsor, UK, August 2008.

Yue Jia and Mark Harman. Constructing subtle faults using higher order mutation testing (best paper award winner). In *8th International Working Conference on Source Code Analysis and Manipulation (SCAM'08)*, pages 249–258, Beijing, China, 2008. IEEE Computer Society.

David Binkley, Nicolas Gold, Mark Harman, Zheng Li, and Kiarash Mahdavi. Evaluating key statements analysis. In *8th International Working Conference on Source Code Analysis and Manipulation (SCAM'08)*, pages 121–130, Beijing, China, 2008. IEEE Computer Society.

Yuanyuan Zhang, Anthony Finkelstein, and Mark Harman. Search based requirements optimisation: Existing work and challenges. In *International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'08)*, volume 5025, pages 88–94, Montpellier, France, 2008. Springer LNCS.

Anthony Finkelstein, Mark Harman, Afshin Mansouri, Jian Ren, and Yuanyuan Zhang. Fairness analysis in requirements assignments. In *16th IEEE International Requirements Engineering Conference*, pages 115–124, Los Alamitos, California, USA, September 2008. IEEE Computer Society Press.

Nadia Alshahwan and Mark Harman. Automated session data repair for web application regression testing. In *1st IEEE International conference on Software Testing (ICST'08)*, pages 298–307, Lillehammer, Norway, 2008. IEEE Computer Society.

Giulio Antoniol, Massimiliano Di Penta, Mark Harman, and Fahim Qureshi. The effect of communication overhead on software maintenance project staffing: a search-based approach. In *23rd IEEE International Conference on Software Maintenance (ICSM 2007)*, pages 315–324, Los Alamitos, California, USA, October 2007. IEEE Computer Society Press.

Mark Harman, Youssef Hassoun, Kiran Lakhotia, Philip McMinn, and Joachim Wegener. The impact of input domain reduction on search-based test data generation. In *ACM Symposium on the Foundations of Software Engineering (FSE '07)*, pages 155–164, Dubrovnik, Croatia, September 2007. Association for Computer Machinery.

Mark Harman and Philip McMinn. A theoretical and empirical analysis of evolutionary testing and hill climbing for structural test data generation. In *International Symposium on Software Testing and Analysis (ISSTA'07)*, pages 73 – 83, London, United Kingdom, July 2007. Association for Computer Machinery.

Mark Harman. Automated test data generation using search based software engineering. In *2nd International Workshop on Automation of Software Test (AST 07)*, page 2, Minneapolis, USA, May 2007. IEEE Computer Society Press.

Mark Harman. Search based software engineering for program comprehension. In *15th International Conference on Program Comprehension (ICPC 07)*, pages 3–13, Banff,

Canada, 2007. IEEE Computer Society Press.

Mark Harman and Laurence Tratt. Pareto optimal search-based refactoring at the design level. In *GECCO 2007: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1106 – 1113, London, UK, July 2007. ACM Press.

Yunayuan Zhang, Mark Harman, and Afshin Mansouri. The multi-objective next release problem (‘best at GECCO’ award winner). In *GECCO 2007: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1129 – 1137, London, UK, July 2007. ACM Press.

Mark Harman, Kiran Lakhota, and Philip McMinn. A multi-objective approach to search-based test data generation. In *GECCO 2007: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1098 – 1105, London, UK, July 2007. ACM Press.

Mark Harman. The current state and future of search based software engineering. In Lionel Briand and Alexander Wolf, editors, *Future of Software Engineering 2007*, pages 342–357, Los Alamitos, California, USA, 2007. IEEE Computer Society Press.

Nicolas Gold, Mark Harman, Zheng Li, and Kiarash Mahdavi. An empirical study of executable concept slice size. In *13th Working Conference on Reverse Engineering (WCRE 06)*, pages 103–114, Benevento, Italy, October 2006.

David Binkley, Mark Harman, and Jens Krinke. Animated visualisation of static analysis: Characterising, explaining and exploiting the approximate nature of static analysis. In *6th International Workshop on Source Code Analysis and Manipulation (SCAM 06)*, pages 43–52, Philadelphia, Pennsylvania, USA, September 2006.

Keith Brian Gallagher, David Binkley, and Mark Harman. Stop-list slicing. In *6th International Workshop on Source Code Analysis and Manipulation (SCAM 06)*, pages 11–20, Philadelphia, Pennsylvania, USA, September 2006.

Nicolas Gold, Mark Harman, Zheng Li, and Kiarash Mahdavi. A search based approach to overlapping concept boundaries. In *22nd International Conference on Software Maintenance (ICSM 06)*, pages 310–319, Philadelphia, Pennsylvania, USA, September 2006.

Mark Harman, Kathleen Steinhöfel, and Alexandros Skaliotis. Search based approaches to component selection and prioritization for the next release problem. In *22nd International Conference on Software Maintenance (ICSM 06)*, pages 176–185, Philadelphia, Pennsylvania, USA, September 2006.

Phil McMinn, Mark Harman, David Binkley, and Paolo Tonella. The species per path approach to search-based test data generation. In *International Symposium on Software Testing and Analysis (ISSTA 06)*, pages 13–24, Portland, Maine, USA., 2006.

Bogdan Korel, Mark Harman, S. Chung, P. Apirukvorapinit, and R. Gupta. Data dependence based testability transformation in automated test generation. In *16th International Symposium on Software Reliability Engineering (ISSRE 05)*, pages 245–254, Chicago, Illinois, USA, November 2005.

Deji Fatiregun, Mark Harman, and Rob Hierons. Search-based amorphous slicing. In *12th International Working Conference on Reverse Engineering (WCRE 05)*, pages 3–12, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, November 2005.

David Binkley and Mark Harman. Forward slices are smaller than backward slices. In *5th IEEE International Workshop on Source Code Analysis and Manipulation*, pages 15–24, Los Alamitos, California, USA, 2005. IEEE Computer Society Press.

Sebastian Danicic, David Binkley, Tibor Gyimóthy, Mark Harman, Ákos Kiss, and Bogdan Korel. Minimal slicing and the relationships between forms of slicing. In *5th IEEE International Workshop on Source Code Analysis and Manipulation (SCAM 05)*, pages 45–54, Los Alamitos, California, USA, 2005. IEEE Computer Society Press. Best paper award winner.

David Binkley and Mark Harman. Locating dependence clusters and dependence pollution. In *21st IEEE International Conference on Software Maintenance*, pages 177–186, Los Alamitos, California, USA, 2005. IEEE Computer Society Press.

Giulio Antoniol, Massimiliano Di Penta, and Mark Harman. Search-based techniques applied to optimization of project planning for a massive maintenance project. In *21st IEEE International Conference on Software Maintenance*, pages 240–249, Los Alamitos, California, USA, 2005. IEEE Computer Society Press.

David Binkley, Mariano Ceccato, Mark Harman, Filippo Ricca, and Paolo Tonella. Automated refactoring of object oriented code into aspects. In *21st IEEE International Conference on Software Maintenance*, pages 27–36, Los Alamitos, California, USA, 2005. IEEE Computer Society Press.

Bogdan Korel, Luay Tahat, and Mark Harman. Test prioritization using system models. In *21st IEEE International Conference on Software Maintenance*, pages 559–568, Los Alamitos, California, USA, 2005. IEEE Computer Society Press.

Phil McMinn, David Binkley, and Mark Harman. Testability transformation for efficient automated test data search in the presence of nesting. In *UK Software Testing Workshop (UK Test 2005)*, Sheffield, UK, September 2005.

Mark Harman, Stephen Swift, and Kiarash Mahdavi. An empirical study of the robustness of two module clustering fitness functions. In *Genetic and Evolutionary Computation Conference (GECCO 2005)*, pages 1029–1036, Washington DC, USA, June 2005. Association for Computer Machinery.

Lin Hu, Mark Harman, David Binkley, and Rob Hierons. Loop squashing transformations for amorphous slicing. In *11th IEEE Working Conference on Reverse Engineering*, pages 152–160, Los Alamitos, California, USA, November 2004. IEEE Computer Society Press.

Dave Binkley, Sebastian Danicic, Tibor Gyimóthy, Mark Harman, Ákos Kiss, and Lahcen Ouarbya. Formalizing executable dynamic and forward slicing. In *4th International Workshop on Source Code Analysis and Manipulation (SCAM 04)*, pages 43–52, Los Alamitos, California, USA, September 2004. IEEE Computer Society Press.

Deji Fatiregun, Mark Harman, and Rob Hierons. Evolving transformation sequences using genetic algorithms. In *4th International Workshop on Source Code Analysis and Manipulation (SCAM 04)*, pages 65–74, Los Alamitos, California, USA, September 2004. IEEE Computer Society Press.

Mark Harman, Dave Binkley, Ranjit Singh, and Rob Hierons. Amorphous procedure extraction. In *4th International Workshop on Source Code Analysis and Manipulation*

(*SCAM 04*), pages 85–94, Los Alamitos, California, USA, September 2004. IEEE Computer Society Press.

Mark Harman and John Clark. Metrics are fitness functions too. In *10th International Software Metrics Symposium (METRICS 2004)*, pages 58–69, Los Alamitos, California, USA, September 2004. IEEE Computer Society Press.

Giulio Antoniol, Massimiliano Di Penta, and Mark Harman. A robust search-based approach to project management in the presence of abandonment, rework, error and uncertainty. In *10th International Software Metrics Symposium (METRICS 2004)*, pages 172–183, Los Alamitos, California, USA, September 2004. IEEE Computer Society Press.

Sebastian Danicic, Andrea De Lucia, and Mark Harman. Building executable union slices using conditioned slicing. In *12th International Workshop on Program Comprehension*, pages 89–97, Los Alamitos, California, USA, June 2004. IEEE Computer Society Press.

Konstantinos Adamopoulos, Mark Harman, and Robert Mark Hierons. Mutation testing using genetic algorithms: A co-evolution approach. In *Genetic and Evolutionary Computation Conference (GECCO 2004), LNCS 3103*, pages 1338–1349, Seattle, Washington, USA, June 2004. Springer.

Karnig Derderian, Qiang Quo, Mark Harman, and Robert Hierons. Computing unique input/output sequences using genetic algorithms. In *3rd International Workshop on Formal Approaches to Testing of Software (FATES 2003)*, pages 164–177, Montral, Canada, 2003. LNCS 2931.

André Baresel, David Wendell Binkley, Mark Harman, and Bogdan Korel. Evolutionary testing in the presence of loop-assigned s: A testability transformation approach. In *International Symposium on Software Testing and Analysis (ISSTA 2004)*, pages 108–118, Omni Parker House Hotel, Boston, Massachusetts, July 2004. Appears in *Software Engineering Notes*, Volume 29, Number 4.

Kiarash Mahdavi, Mark Harman, and Robert Mark Hierons. A multiple hill climbing approach to software module clustering. In *IEEE International Conference on Software Maintenance*, pages 315–324, Los Alamitos, California, USA, September 2003. IEEE Computer Society Press.

David Wendell Binkley and Mark Harman. A large-scale empirical study of forward and backward static slice size and context sensitivity. In *IEEE International Conference on Software Maintenance*, pages 44–53, Los Alamitos, California, USA, September 2003. IEEE Computer Society Press.

David Wendell Binkley and Mark Harman. Results from a large-scale study of performance optimization techniques for source code analyses based on graph reachability algorithms. In *IEEE International Workshop on Source Code Analysis and Manipulation (SCAM 2003)*, pages 203–212, Los Alamitos, California, USA, September 2003. IEEE Computer Society Press.

Rob Hierons, Mark Harman, and Harbhajan Singh. Automatically generating information from a Z specification to support the classification tree method. In *3rd International Conference of B and Z Users (ZB2003)*, pages 388–407, Turku, Finland, June 2003. Springer. LNCS 2651.

David Wendell Binkley and Mark Harman. An empirical study of predicate dependence

levels and trends. In 25th *IEEE International Conference and Software Engineering (ICSE 2003)*, pages 330–339, Los Alamitos, California, USA, May 2003. IEEE Computer Society Press.

Xingyuan Zhang, Malcolm Munro, Mark Harman, and Lin Hu. Formal verification of communication protocol using type theory. In *International Conference on Communications Technology (ICCT 2003)*, pages 1585 – 1593, Beijing, China, April 2003. Beijing University of Posts and Telecommunications Press.

Andrea De Lucia, Mark Harman, Rob Hierons, and Jens Krinke. Unions of slices are not slices. In 7th *IEEE European Conference on Software Maintenance and Reengineering (CSMR 2003)*, pages 363 – 367, Los Alamitos, California, USA, March 2003. IEEE Computer Society Press.

Mark Harman, Nicolas Gold, Robert Mark Hierons, and David Wendell Binkley. Code extraction algorithms which unify slicing and concept assignment. In *IEEE Working Conference on Reverse Engineering (WCRE 2002)*, pages 11 – 21, Los Alamitos, California, USA, October 2002. IEEE Computer Society Press.

Mohammed Daoudi, Sebastian Danicic, John Howroyd, Mark Harman, Chris Fox, Lahcen Ouarbya, and Martin Ward. ConSUS: A scalable approach to conditioned slicing. In *IEEE Working Conference on Reverse Engineering (WCRE 2002)*, pages 109 – 118, Los Alamitos, California, USA, October 2002. IEEE Computer Society Press. Invited for special issue of the Journal of Systems and Software as best paper from WCRE 2002.

Lahcen Ouarbya, Sebastian Danicic, Dave (Mohammed) Daoudi, Mark Harman, and Chris Fox. A denotational interprocedural program slicer. In *IEEE Working Conference on Reverse Engineering (WCRE 2002)*, pages 181 – 189, Los Alamitos, California, USA, October 2002. IEEE Computer Society Press.

Mark Harman, Lin Hu, Robert Mark Hierons, Xingyuan Zhang, Malcolm Munro, José Javier Dolado, Mari Carmen Otero, and Joachim Wegener. A post-placement side-effect removal algorithm. In *IEEE International Conference on Software Maintenance*, pages 2–11, Los Alamitos, California, USA, October 2002. IEEE Computer Society Press.

Mark Harman, Lin Hu, Robert Mark Hierons, Chris Fox, Sebastian Danicic, André Baresel, Harmen Sthamer, and Joachim Wegener. Evolutionary testing supported by slicing and transformation. In *IEEE International Conference on Software Maintenance*, page 285, Los Alamitos, California, USA, October 2002. IEEE Computer Society Press.

Xingyuan Zhang, Malcolm Munro, Mark Harman, and Lin Hu. Mechanized operational semantics of WSL. In *IEEE International Workshop on Source Code Analysis and Manipulation (SCAM 2002)*, pages 73–82, Los Alamitos, California, USA, October 2002. IEEE Computer Society Press.

Mark Harman, Chris Fox, Robert Mark Hierons, Lin Hu, Sebastian Danicic, and Joachim Wegener. Vada: A transformation-based system for variable dependence analysis. In *IEEE International Workshop on Source Code Analysis and Manipulation (SCAM 2002)*, pages 55–64, Los Alamitos, California, USA, October 2002. IEEE Computer Society Press.

Mark Harman, Lin Hu, Xingyuan Zhang, Malcolm Munro, Sebastian Danicic, Mohammed Daoudi, and Lahcen Ouarbya. An interprocedural amorphous slicer for WSL. In *IEEE International Workshop on Source Code Analysis and Manipulation (SCAM 2002)*, pages 105–114, Los Alamitos, California, USA, October 2002. IEEE Computer Society Press.

Selected for consideration for the special issue of the Journal of Automated Software Engineering.

Xingyuan Zhang, Malcolm Munro, Mark Harman, and Lin Hu. Weakest precondition for general recursive programs formalized in Coq. In 15th *International Conference on Theorem Proving in Higher Order Logics (TPHOLs 2002)*, pages 332–348, Hampton, Virginia, USA, August 2002. Springer Verlag. LNCS 2410.

Jonathan Bowen, Kirill Bogdanov, John Clark, Mark Harman, Robert Mark Hierons, and Paul Krause. Fortest: Formal methods and testing (joint panel paper). In 26th *IEEE Annual Computer Software and Applications Conference (COMPSAC 2002)*, pages 91–101, Los Alamitos, California, USA, August 2002. IEEE Computer Society Press.

Yoga Sivagurunathan, Mark Harman, and Bala Sivagurunathan. Slice-based dynamic memory modelling – a case study. In 26th *IEEE Annual Computer Software and Applications Conference (COMPSAC 2002)*, pages 351–356, Los Alamitos, California, USA, August 2002. IEEE Computer Society Press.

Mark Harman, Lin Hu, Robert Hierons, Andr Baresel, and Harmen Sthamer. Improving evolutionary testing by flag removal (‘best at GECCO’ award winner). In *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1359–1366, San Francisco, CA 94104, USA, 9-13 July 2002. Morgan Kaufmann Publishers.

Mark Harman, Robert Hierons, and Mark Proctor. A new representation and crossover operator for search-based optimization of software modularization. In *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1351–1358, San Francisco, CA 94104, USA, 9-13 July 2002. Morgan Kaufmann Publishers.

Mark Harman, Rob Mark Hierons, Sebastian Danicic, John Howroyd, and Chris Fox. Pre/post conditioned slicing. In *IEEE International Conference on Software Maintenance (ICSM’01)*, pages 138–147, Los Alamitos, California, USA, November 2001. IEEE Computer Society Press.

Mark Harman, Lin Hu, Xingyuan Zhang, and Malcolm Munro. GUSTT: An amorphous slicing system which combines slicing and transformation. In 1st *Workshop on Analysis, Slicing, and Transformation (AST 2001)*, pages 271–280, Los Alamitos, California, USA, October 2001. IEEE Computer Society Press.

Mark Harman, Rob Mark Hierons, Sebastian Danicic, John Howroyd, Mike Laurence, and Chris Fox. Node coarsening calculi for program slicing. In 8th *Working Conference on Reverse Engineering*, pages 25–34, Los Alamitos, California, USA, October 2001. IEEE Computer Society Press.

Mark Harman and Bryan Jones. SEMINAL: Software engineering using metaheuristic innovative algorithms. In 23rd *International Conference on Software Engineering (ICSE 2001)*, pages 762–763, Los Alamitos, California, USA, May 2001. IEEE Computer Society Press.

Mark Harman, Lin Hu, Xingyuan Zhang, and Malcolm Munro. Side-effect removal transformation. In 9th *IEEE International Workshop on Program Comprehension*, pages 310–319, Los Alamitos, California, USA, May 2001. IEEE Computer Society Press.

Chris Fox, Mark Harman, Rob Mark Hierons, and Sebastian Danicic. Backward conditioning: a new program specialisation technique and its application to program comprehension.

In 9th *IEEE International Workshop on Program Comprehension*, pages 89–97, Los Alamitos, California, USA, May 2001. IEEE Computer Society Press.

Mark Harman, Rob Mark Hierons, and Sebastian Danicic. The relationship between program dependence and mutation analysis. In W. Eric Wong, editor, *Mutation Testing for the New Century (proceedings of Mutation 2000)*, pages 5–13, San Jose, California, USA, October 2001. Kluwer.

Sebastian Danicic, Chris Fox, Mark Harman, and Rob Mark Hierons. ConSIT: A conditioned program slicer. In *IEEE International Conference on Software Maintenance (ICSM'00)*, pages 216–226, Los Alamitos, California, USA, October 2000. IEEE Computer Society Press.

David Wendell Binkley, Mark Harman, L. Ross Raszewski, and Christopher Smith. An empirical study of amorphous slicing as a program comprehension support tool. In 8th *IEEE International Workshop on Program Comprehension*, pages 161–170, Los Alamitos, California, USA, June 2000. IEEE Computer Society Press.

Rob Mark Hierons and Mark Harman. Program analysis and test hypotheses complement. In *IEEE ICSE International Workshop on Automated Program Analysis, Testing and Verification*, pages 32–39, Limerick, Ireland, June 2000.

Sebastian Danicic and Mark Harman. Espresso: A slicer generator. In *ACM Symposium on Applied Computing, (SAC'00)*, pages 831–839, Como, Italy, March 2000.

Mark Harman, Chris Fox, Rob Mark Hierons, David Wendell Binkley, and Sebastian Danicic. Program simplification as a means of approximating undecidable propositions. In 7th *IEEE International Workshop on Program Comprehension (IWPC'99)*, pages 208–217, Los Alamitos, California, USA, May 1999. IEEE Computer Society Press.

Mark Harman, Yoga Sivagurunathan, and Sebastian Danicic. Analysis of dynamic memory access using amorphous slicing. In *IEEE International Conference on Software Maintenance (ICSM'98)*, pages 336–345, Los Alamitos, California, USA, November 1998. IEEE Computer Society Press.

Mark Harman, Margaret Okunlawon, Bala Sivagurunathan, and Sebastian Danicic. Slice-based measurement of coupling. In Rachel Harrison, editor, 19th *ICSE, Workshop on Process Modelling and Empirical Studies of Software Evolution*, Boston, Massachusetts, USA, May 1997.

Yoga Sivagurunathan, Mark Harman, and Sebastian Danicic. Slicing, I/O and the implicit state. In Mariam Kamkar, editor, 3rd *International Workshop on Automated Debugging (AADEBUG'97)*, volume 2 of *Linköping Electronic Articles in Computer and Information Science*, pages 59–65, Linköping, Sweden, May 1997.

Mark Harman and Sebastian Danicic. Amorphous program slicing. In 5th *IEEE International Workshop on Program Comprehension (IWPC'97)*, pages 70–79, Los Alamitos, California, USA, May 1997. IEEE Computer Society Press.

Book (1)

Mark Harman and Ray Jones. *First Course in C++: A Gentle Introduction*. McGraw-Hill, 1996.

Refereed Tutorial Presentations (3)

Mark Harman and Joachim Wegener. Getting results with search-based software engineering: Tutorial. In *26th IEEE International Conference and Software Engineering (ICSE 2004)*, pages 728–729, Los Alamitos, California, USA, 2004. IEEE Computer Society Press.

Mark Harman and Joachim Wegener. Evolutionary testing: Tutorial. In *Genetic and Evolutionary Computation (GECCO)*, Chicago, July 2003.

Mark Harman and Joachim Wegener. Evolutionary testing. In *Genetic and Evolutionary Computation (GECCO)*, New York, July 2002.

Refereed Poster papers (4)

Giulio Antoniol, Massimiliano Di Penta, and Mark Harman. Search-based techniques for optimizing software project resource allocation. In *Genetic and Evolutionary Computation Conference (GECCO 2004)*, pages 1425–1426, Seattle, Washington, USA, June 2004. LNCS 3103.

Karnig Dederian, Robert Mark Hierons, Mark Harman, and Qiang Quo. Input sequence generation for testing of communicating finite state machines (CFSMs). In *Genetic and Evolutionary Computation Conference (GECCO 2004)*, pages 1429–1430, Seattle, Washington, USA, June 2004. LNCS 3103.

Deji Fatiregun, Mark Harman, and Robert Hierons. Search based transformations. In *Genetic and Evolutionary Computation – GECCO-2003*, volume 2724 of LNCS, pages 2511–2512, Berlin, 12-16 July 2003. Springer-Verlag.

Kiarash Mahdavi, Mark Harman, and Robert Hierons. Finding building blocks for software clustering. In *Genetic and Evolutionary Computation – GECCO-2003*, volume 2724 of LNCS, pages 2513–2514, Berlin, 12-16 July 2003. Springer-Verlag.

Lightly Refereed Conferences, Workshops and Journals (13)

Sebastian Danicic, Mark Harman, Robert Hierons, John Howroyd, and Mike Laurence. Applications of linear program schematology in dependence analysis. In *1st. International Workshop on Programming Language Interference and Dependence*, Verona, Italy, August 2004.

Sebastian Danicic, Mark Harman, John Howroyd, and Lahcen Ouarbya. A lazy semantics for program slicing. In *1st. International Workshop on Programming Language Interference and Dependence*, Verona, Italy, August 2004.

Stuart Reid, Mark Harman, Rob Mark Hierons, Mike Holcombe, Bryan Jones, Marc Roper, and Martin Woodward. A framework for measurement in software testing. In *7th European Conference Software Testing Analysis & Review (EuroSTAR'99)*, Princessa Sophia Intercontinental Hotel, Barcelona, Spain, November 1999.

Mark Harman, Rob Mark Hierons, Mike Holcombe, Bryan Jones, Stuart Reid, Marc Roper, and Martin Woodward. Towards a maturity model for empirical studies of software testing. In *5th IEEE Workshop on Empirical Studies of Software Maintenance (WESS'99)*, Keble College, Oxford, UK, September 1999.

Mark Harman, Sebastian Danicic, Yogasundary Sivagurunathan, and Dan Simpson. The next 700 slicing criteria. In Malcolm Munro, editor, *2nd UK workshop on program comprehension*, Durham University, UK, July 1996.

Mark Harman, Sebastian Danicic, and Yogasundary Sivagurunathan. Program comprehension assisted by slicing and transformation. In Malcolm Munro, editor, *1st UK workshop on program comprehension*, Durham University, UK, July 1995.

Mark Harman, Sebastian Danicic, Balasubramaniam Sivagurunathan, Barry Jones, and Yogasundary Sivagurunathan. Cohesion metrics. In *8th International Quality Week*, pages Paper 3-T-2, pp 1-14, San Francisco, May 1995.

Mark Harman and Sebastian Danicic. Using an interpreter to teach introductory programming. In Staples and Brebbia, editors, *1st International Conference on Software Engineering in Higher Education (SEHE'94)*, November 1994.

Mark Harman and Sebastian Danicic. Using a functional language to teach elements of formal methods. In Staples and Brebbia, editors, *1st International Conference on Software Engineering in Higher Education (SEHE'94)*, November 1994.

Mark Harman and Sebastian Danicic. Using program slicing to simplify testing. In *2nd EuroSTAR Conference on Software Testing Analysis and Review*, Brussels, October 1994.

Mark Harman and Sebastian Danicic. Practical subprogram verification – An approach which uses slicing, metrics and axiomatic verification. In *Proceedings of the 2nd BCS Conference on Software Quality Management*, pages 26-28, Edinburgh, Scotland, July 1994.

Mark Harman and Sebastian Danicic. A new approach to program slicing. In *7th International Quality Week*, pages Paper 4-T-4, pp 1-14, San Francisco, May 17 – 20 1994.

Mark Harman and Sebastian Danicic. Projecting functional models of imperative programs. *ACM SIGPLAN Notices*, 28(11):42-52, November 1993.

PhD Thesis

Mark Harman. *Functional Models of Procedural Programs*. PhD thesis, University of North London, 1992.

Trade Press Articles (13)

Mark Harman. How small is a bit?: The uncertain world of quantum computing. *EXE*, pages 36-41, September 1999.

Mark Harman. Sifting through the wreckage. *EXE*, page 5, March 1999. Editorial.

Mark Harman. Project estimation: How long is this going to take? *EXE*, pages 20-24, November 1998.

Mark Harman. Testing techniques: Advanced techniques. *EXE*, pages 29-34, August 1998.

Mark Harman. Testing in black and white. *EXE*, pages 18-26, July 1998.

- Mark Harman. DIY software agents. *EXE*, pages 18–24, December 1997.
- Mark Harman. \LaTeX : typesetting as a programming language. *EXE*, pages 31–38, October 1997.
- Mark Harman. Program transformation: New programs for old. *EXE*, pages 25–30, July 1997.
- Mark Harman and Chris Kopec. The mating game. *EXE*, 11(12):39–42, May 1997.
- Mark Harman and Chris Kopec. Genetic algorithms: Growing your own code. *EXE*, 11(11):17–22, April 1997.
- Mark Harman. Cleaving together: Program cohesion with slices. *EXE*, 11(8):35–42, January 1997.
- Mark Harman. Carving up bugs. *EXE*, 11(6):39–42, November 1996.
- Mark Harman. A piece of cake: Slicing as a debugging technique. *EXE*, 11(5):25–28, October 1996.