DAASE: Dynamic Adaptive Automated Software Engineering

Keywords: Search Based Software Engineering, Hyper-Heuristics.

Principal Investigator: Prof. Mark Harman, Head of Software Systems Engineering, UCL.
Lead Partners: Prof. Edmund Burke, Deputy Principal for Research, University of Stirling.
Prof. John A. Clark, Deputy Head (Research), Department of Computer Science, University of York.
Prof. Xin Yao, Director of CERCIA, University of Birmingham.

DAASE integrates two teams of researchers from the Operational Research and Search Based Software Engineering communities. Both groups of researchers are widely regarded as world leading, having pioneered the fields of Hyper-Heuristics and Search Based Software Engineering (SBSE); the two key fields that DAASE brings together. The four lead investigators' work in these two fields continues to have a major impact on research and practice: they have many 'Essential Science Indicators' (ESI) citation levels within the top 1% (and two within the top 0.01%) of all papers in Computer Science. Furthermore, the outcomes of their research work are used by many organisations including ABB, Daimler, DSTL, Ericsson, Google, Honda, IBM, KLM, Microsoft, Motorola and Park Air Systems. The four investigators have given a combined total of 104 keynotes and received 28 best paper awards. This international leadership is backed by extensive experience on large projects, including three platform grants, a science and innovation award and two previous programme grants.

Mark Harman is Professor of Software Engineering at UCL and director of the Centre for Research in Evolution Search and Testing (CREST) within the Department of Computer Science at UCL. Prof. Harman is a recognised world leader in the area of Search Based Software Engineering (SBSE). He has given 18 keynotes on SBSE at international academic conferences and workshops, 2 keynotes at industry conferences and an invited paper and presentation at the world-leading software engineering conference, ICSE. He has published widely on SBSE across a variety of software engineering applications from Requirements Analysis and Management to Maintenance and Testing. Many of his SBSE papers are in the top 1% of all cited papers and he has received 8 best paper awards. He has served on 137 program committees and as the guest editor for 14 journal special issues. His SBSE work has been published in many leading conferences and journals on software engineering. For example, in the last year alone, he has published in TSE, TOSEM, ASE, FSE and ICSE; the two top journals and three top conferences.

Prof. Harman is the director of the large EPSRC project SEBASE (Software Engineering by Automated Search), a collaboration on widespread application of SBSE theory, for which Prof. Clark and Prof. Yao are co-PIs at York and Birmingham. Prof. Harman's work on SBSE has been funded by the EPSRC, EU and industry with total funding as PI of £5.5M. In 2009 he was awarded a platform grant for the CREST centre. He has extensive experience of successful project management for large and small projects. Of the 20 grants Harman submitted to EPSRC as PI, 15 were funded, of which, 4 were ranked number one by the prioritisation panels that considered them.

Edmund Burke will take up the position of Deputy Principal for Research at the University of Stirling on December 1st, 2011. Prof. Burke is currently Dean of the Faculty of Science at the University of Nottingham, a post he has held since 2008. From 2005 to 2008 he was Head of the School of Computer Science at Nottingham. Throughout this period of senior management positions, he also led the Automated Scheduling, Optimisation and Planning (ASAP) group at Nottingham and was PI on two platform grants and many other grants on meta- and hyper-heuristics, continuing to actively contribute and manage the group's research portfolio. With the move of Prof. Burke and others from the ASAP group to Stirling, ASAP will become a two-site research group, with Prof. Petrovic leading the Nottingham team and Prof. Burke leading the Stirling team. The 2-page work plan document explains in more detail how the ASAP group will work on DAASE.

Prof. Burke is widely recognised as one of the leading scientists in the world on decision support methodologies and is specifically listed in ISI Essential Science Indicators as a highly cited author. He is editor-in-chief of the Journal of Scheduling and is the Combinatorial Optimisation Area Editor of the Journal of Heuristics. He has given 24 keynotes, edited/authored 14 books and has published over 230 refereed papers. He has pioneered the field of hyper-heuristics and has nine papers in the top 1% for citations. His total funding as PI is £8.5M.

John Clark has been PI for four EPSRC grants (total £1.34M) since becoming a professor in 2005, including the SEBASE project for which he is the PI at York, with over £500K over the period 2006-2011 from other sources. He has won 8 awards, including the Best Genetic Programming Paper Prize at GECCO 2004, a $1000 Human Competitive Results Prize at GECCO 2005, the ICSEA 2009 Prize for Testing, best paper prize at SBST 2011 and best SBSE paper prize at GECCO 2011. He has chaired conferences on Search Based Software Engineering (SSBSE 2010), on Testing (Mutation 2007) and on Security (Security in Pervasive Computing 2006).

Xin Yao is a world-leading researcher in Evolutionary Computation and Birmingham PI of the SEBASE project. He won the IEEE Donald G. Fink Prize Paper Award, an IEEE-wide award (in 2001) and outstanding paper awards from IEEE Transactions on Evolutionary Computation (in 2010) and IEEE Transactions on Neural Networks (in 2011). He was the editor-in-chief (2003-08) of IEEE Transactions on Evolutionary Computation (which lies within the top 1% of all SCI-indexed Computer Science journals in the world). He has been invited to give 61 keynote or plenary speeches at international conferences, holds four honorary visiting professorships at overseas universities and has published more than 370 refereed papers (including 167 in journals). Two of his papers are in the top 0.01% for citations in Computer Science and many others are in the top 1%. His total funding as PI is £8.5M, from a range of sources including 11 EPSRC grants.

The 6 DAASE Named RAs: DAASE will hit the ground running with the appointment of 6 named RAs: Yuri Bykov and Gabriela Ochoa (Stirling); Yuanyuan Zhang and Yue Jia (UCL); Simon Poulding (York) and Leandro Minku (Birmingham). As their 2-page CVs (included with this proposal) demonstrate, all 6 have excellent track records and expertise, making them ideal initial appointments for DAASE.

Public Engagement Leaders Dr. Peter Bentley and Dr. Sue Black: The two DAASE Public Engagement Leaders will lead DAASE advocacy for science and engineering as explained in Section 6. Dr. Black won the PepsiCo Women's Inspiration Award (WIN) award in 2011, while Dr. Bentley is well known for his popular science books, each of which has sales in the tens of thousands. They are, of course, also outstanding researchers as well as leaders in public understanding of science: between them they have given 13 keynotes and have 5 best paper awards.

The DAASE Industrial Partners: As their letters of support show, DAASE has a balanced portfolio of ten industrial partners that will provide case studies and part funding for PhD studentships and will host visits. Their staff will also visit the 4 DAASE university sites and assist with the evaluation of DAASE results. More details can be found in the 'Pathways to Impact' document.

Other CIs: The 4 universities that are collaborating to form the DAASE project have committed significant additional strategic funding of their own. This funding, described in their letters of support and summarised in the 'Justification of Resources' document, includes 26 PhD studentships and 6 additional lecturer appointments, each of whom will become an additional DAASE CI. UCL will also start with one existing CI, Dr. Krinke, a senior lecturer at UCL and also deputy director of the CREST centre for which Prof. Harman is the director. Dr. Krinke will focus on ways in which source code analysis can both contribute to and benefit from Adaptive Automated Software Engineering.

The DAASE Academic Partner Outreach Programme: DAASE will support a rolling programme of visiting fellows and sabbatical placements from both the national and international research and practitioner communities (see Sections **??** and **??** for more details). This will bring in other leaders in the national and international Software Engineering, SBSE, Operational Research and Computational Search Communities.

1 Vision and Ambition

DAASE will create a new approach to software engineering, placing adaptive automation at the heart of the development process and the products it creates.

Current Software Engineering: Current software development processes are expensive, laborious and error prone. They achieve adaptivity at only a glacial pace, largely through enormous human effort, forcing highly skilled engineers to waste significant time adapting many tedious implementation details. Often, the resulting software is equally inflexible, forcing users to also rely on their innate *human adaptivity* to find 'workarounds'. As the letters of support from the DAASE industrial partners demonstrate, this creates a pressing need for greater automation and adaptivity.

Future DAASE Software Engineering: Suppose we automate large parts of the development process using computational search: requirements engineering, project planning and testing now become unified into a single automated activity. As requirements change, the project plans and associated tests are adapted to best suit the changes. Now suppose we further embed this adaptivity within the software product itself: smaller changes to the operating environment can now be handled automatically by in-situ adaptation. Feedback from the operating environment to the development process will also speed up adaption to larger changes that cannot be handled by in-situ adaptation alone.

This is the new approach to software engineering DAASE seeks to create. It places computational search at the heart of the process and the software it creates, embedding adaptivity into both. DAASE will also create an array of new development processes, methods, techniques and tools for this new kind of software engineering, radically transforming theory and practice. We do not underestimate the challenges this research agenda poses.

What is 'Adaptive Automation'?: The DAASE meaning of adaptive automation is three-fold:

1. Adaptive processes 2. Adaptive products & 3. Adaptive product-process cycles.

That is, in order to increase software development automation, we need greater adaptivity to cater for different development activities. This requires a form of adaptivity that DAASE will seek to embed into both the software development process and the software systems themselves. Most ambitiously of all, DAASE will exploit information from the deployed software to create a 'virtuous cycle' of adaptivity between an ever-increasingly automated and adaptive process and the products it creates. This is what we mean by 'Adaptive Automation'.

DAASE Hyper-Heuristic Approach: DAASE will develop a hyper-heuristic approach to adaptive automation. A hyper-heuristic is a methodology for selecting or generating heuristics. Most heuristic methods in the literature operate on a search space of potential solutions to a particular problem. However, a hyper-heuristic operates on a search space of heuristics. This research agenda will raise fundamental questions. For example: how best do we draw the dividing line between adaptive automation for small changes and human intervention to invoke more fundamental adaption and to provide oversight and decision making? The Technical Annex (Sections **??**, **??**, **??** and **??**) provides more detail, explaining how we propose to carry out this programme.

Context: The DAASE approach should be understood in the context of Search Based Software Engineering (SBSE), a rapidly developing new approach to software engineering automation in which computational search techniques are used to automate instances of software engineering problems.

Why Can't SBSE Already Provide Adaptive Automated Software Engineering? Current SBSE automates specific *instances* of problems, rather than *whole classes* of problems. A dramatic increase in the breadth of automation is required to achieve SBSE's full potential. Current SBSE techniques are also insufficiently adaptive; they are not designed to handle dynamically changing software development environments, nor do they currently create software products that are, themselves, adaptive.

2 Background and Context

Since the term SBSE was coined by Harman and Jones [14], there has been an explosion of activity, creating a body of work that is sufficiently large to support several surveys and analyses of the literature and includes diverse software engineering topics such as requirements [11, 27], predictive modelling [1, 13], design [20, 21], non-functional properties [2, 23] and testing [4, 15, 18, 19]. Figure 1 shows the growth in publications on SBSE from 1999 to 2010.

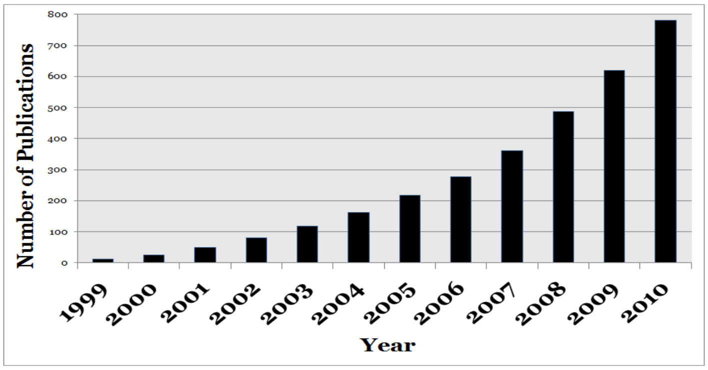

Figure 1: **Growth in SBSE publications**

SBSE has also found its way into industrial practice. One of the earliest industrial examples of the application of SBSE was the Daimler evolutionary testing system [22]. Joachim Wegener, one of the DAASE industrial partners, led the team that implemented this system, before moving to Berner&Mattner. More recently, Microsoft used SBSE to handle floating point computation in its Pex software testing tool [8, 16], while Google incorporated SBSE regression test optimisation into its test process [25]. NASA [10], Motorola [5] and Ericsson [26] have all also experimented with SBSE for requirements analysis and optimisation, while Ericsson [3] has also used genetic programming for fault prediction. As the citations indicate, much of this industrial uptake has been led by the DAASE proposers.

The more general growth in SBSE activity has also been led by the wider UK research community, through the EPSRC SEMINAL network (1999-2002) and the large programme grant SEBASE (2006-2011), both of which were led by Prof. Harman, the PI of the DAASE project. The DAASE CIs Clark and Yao were also co-PIs of the SEBASE project and Clark was a co-founder of the SEMINAL network with Harman [9]. A recent ten year retrospective [12] listed Harman, Yao and Clark as the top three authors in SBSE, with 5 papers in the top 10 by citation. Figure 1 demonstrates that SBSE authorship has grown quadratically, while Figure 2 shows that the UK currently leads.

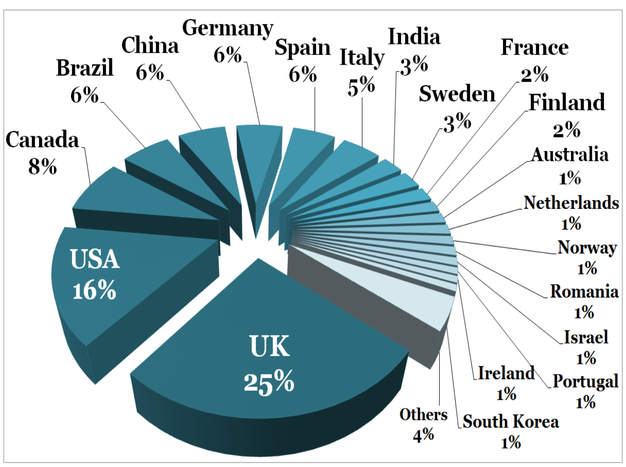

Figure 2: **Current SBSE authorship by country**

DAASE will bring to the SBSE community and the wider software engineering community, exciting new developments in hyper-heuristics [7]. Over the past five years, the field of Hyper-Heuristics has also grown rapidly. In 2006 a Google search on 'hyper-heuristics' yielded approximately 1,000 hits. In September 2011, the same search yielded more than 3,370,000 hits. In the intervening time, there have been, dedicated to hyper-heuristics, 10 Workshops, 12 Special Sessions, 7 Tutorials, and 3 Journal Special Issues.

Timeliness and Relevance: The growth of SBSE has demonstrated that it is possible to effect a significant change of perspective in the theory and practice of software engineering. The UK is providing international leadership in this change of thinking. In the field of computational search too, UK research is at the forefront of hyper-heuristics. The DASSE agenda of adaptive automation thus comes at a time when the research and practitioner communities are receptive to these ideas. DAASE will ensure that the UK extends this world leadership in software engineering automation and will drive forward this research agenda to address the new challenge of adaptive automation.

3 Importance and Strategic Fit

This section addresses the questions about 'importance' listed on Page 3 (of 10) of the referee form.

Underpinning other areas: DAASE brings together two aspects of the EPSRC portfolio (Software Engineering and Artificial Intelligence Technologies) which are not currently linked in the EPSRC graph of Information and Communications Technology (ICT) capacity relationships[1]. At the broadest level, it also creates strong connections between Operational Research (OR) and Computer Science. Since OR lies partly within the Mathematics programme and Computer Science within the ICT programme, this creates cross–programme linkages. In recognition of this inter-disciplinarity, the Mathematics and ICT EPSRC programmes have agreed to share DAASE costs, if DAASE is funded.

The Success of UK Economy and Emerging Industry: Increased software development automation can give a strong competitive edge to smaller companies, increasing margins and reducing time to market. Adaptivity is also a key success driver in a rapidly changing market place, such as that typified by the software development industry. This is why UCL has committed to fund a full time Business Development Manager to allow DAASE to interact with an order of magnitude more SMEs than would be possible were the research team themselves to be managing and acquiring these relationships.

EPSRC Challenge Themes and ICT Programme Priorities: DAASE is directly relevant to the Digital Economy EPSRC Challenge Theme, since cost-effective, adaptive software is critical to so many applications that underpin the digital economy. DAASE also addresses 3 of the 5 ICT Programme Priorities (no project is expected to address all 5). DAASE will develop automated decision support for engineers faced with large scale complex software engineering decision making for multiple conflicting and competing objectives. This is directly relevant to the ICT Programme Priority '**Towards an intelligent information infrastructure (TI3)**'. DAASE will be both a user of multicore architectures and a producer of adaptive multicore deployment techniques. The cross cutting engineering challenge of 'Scalability' described in the Technical Annex (Section **??**), will exploit our recent (award winning) work on GPGPU for SBSE scalability [24], while the problem of balancing the competing concerns of multicore deployment will be addressed by the 'multi-objective' cross cutting concern (Section **??**). In this way DAASE will also be highly relevant to the ICT Programme Priority '**Many-core architectures and concurrency in distributed and embedded systems**'. Naturally, DAASE's strengthening of ties between Software Engineering and Artificial Intelligence Technologies and between the Operational Research and Computer Science will also address the ICT Programme Priority '**Working together**'.

4 Research Programme and Methodology

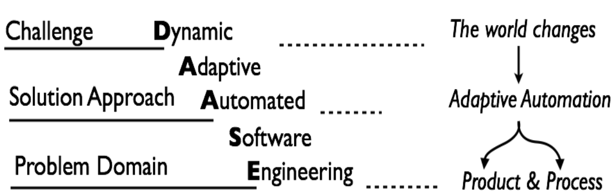

Figure 3: **The DAASE Acronym Explained**

The acronym DAASE is intended to capture the problem (a 'dynamic' world), the two inter-related planks of our solution (adaptivity and automation) and our target application area (software engineering). This is depicted in Figure 3, which explains the way the acronym captures problem, solution approach and the application domain we seek to transform through the DAASE programme. In the context of DAASE, 'dynamic' means that change is so rapid that it may happen *during* the execution of computational search algorithm.

4.1 Research Aims and Objectives

DAASE will develop theories, algorithms, methods, techniques and tools for automated and adaptive software engineering, able to handle dynamically changing development processes and dynamically changing operating environments. The project has three primary objectives, each of which maps onto one of DAASE's three overarching strands of research.

[1] http://www.epsrc.ac.uk/ourportfolio/themes/ict/Pages/default.aspx

These three primary DAASE objectives are:

1. **Underpinning Theory Strand:** DAASE will develop a theory of adaptive automated software engineering that characterises worst case and typical case performance. This will help us to understand which computational search techniques will work best, when and why.

2. **Algorithmic Strand:** DAASE will develop hyper-heuristic computational search algorithms for adaptive automated software engineering. This will lead to a radically new approach to software development processes. Computational search will be a central driving force for automation and adaptive responses to dynamic environments. Adaptivity will be built into the development processes and the products it produces.

3. **Cross Cutting Engineering Principles Strand:** DAASE will develop software engineering methods that generalise and integrate across software development phases, activities and domains to address the engineering concerns that cut across all aspects of software engineering, such as scalability and the need to simultaneously handle multiple objectives.

4.2 The Three Strand Project Structure

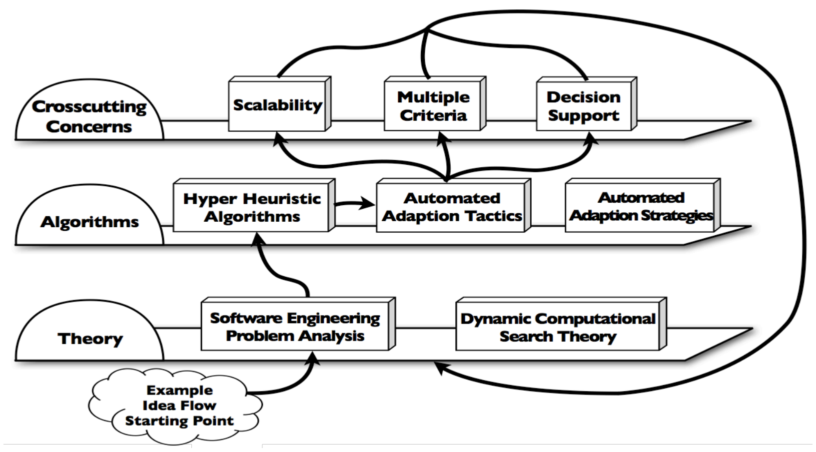

Figure 4: **DAASE's overall three strand project architecture:** The arrows give just one illustrative example of a typical DAASE knowledge flow. These flows cycle round the three research strands.

DAASE will develop fundamental insights into hyper-heuristic search and optimisation, applied to a wide and diverse range of real world software systems, that will provide motivation, relevance, evaluation and a testing ground for new ideas. The project is formed of three research strands (depicted in Figure 4). We shall seek truly deep and generalisable insights, with research into each problem domain contributing to different cross cutting themes. The two page work plan and the Technical Annex explain how the project will address these three research strands. This section describes the DAASE objectives and gives an overview of these three core research strands, explaining how they allow for the flexibility needed by the 'programme grant' style funding mechanism.

The DAASE plans for visits, workshops and group working will be designed to encourage and facilitate the knowledge flows that will cycle round the three research strands. As these 'knowledge flows' cycle through the three research strands, increasing engineering generalisation and theoretical applicability will become possible, encompassing adaptive automation of an ever-wider spectrum of domains and problem scenarios. Our aim is that these cyclic flows should resonate, amplify and re-enforce one another throughout the project.

One example of such a cycle of knowledge flow and collaboration is illustrated by the arrows in Figure 4. In this example flow, let us suppose we start with a theoretical analysis of software test data selection problems, leading to hyper-heuristic algorithms for test case selection that can adapt to different test objectives. This leads to the construction of automated adaptive tactics for test case selection. We then address the cross-cutting issue of how to scale the selection to very large test sets. We also consider how to use our multiple test objectives as multiple criteria for decision support. This leads us to define engineering principles that will underpin our future work on scalable, multi-objective decision support for test case selection.

Of course, our new approaches to test case selection raise new theoretical questions about the analysis of multiple objectives for decision support, thereby cycling back new challenges to the theoretical foundations research strand. This is merely an illustrative example: flows can start from any point in the project and may cycle several times round the three strands, drawing in different themes and researchers as they ripple through the project.

This three tiered project structure is designed to be flexible and resilient. As the project progresses there will be new engineering insights and these will create novel foundational scientific questions, resulting in changes to the composition of the strands, but not to the need to have three strands, nor to the cycles of information flow throughout the project. It is also probable that new algorithms and opportunities to evaluate the project will arise. The algorithmic strand of the overall project architecture can also accommodate these. The DAASE project structure thus has built-in flexibility. We anticipate, indeed we hope, that new cross cutting issues will emerge during the lifetime of the project. However, as Figure 4 shows, any such emergent cross cutting issues can be incorporated quite naturally without disrupting the overall project structure.

5 Added Value

This section explains why DAASE will be more effective by being funded as a single large programme grant, rather than as a series of smaller disconnected projects and how this 'programme grant' funding mode will bring added value.

The Need for 'Programme Grant' Mode Funding: In order to realise the vision of adaptive automation *across the spectrum* of widely diverse software systems, we need to simultaneously attack a broad range of software domains, facilitating the generalisation that comes from an overarching analysis of the full set of application domains. The DAASE project will be a single, co-ordinated programme of integrated research in which all sub-projects will complement one another, free from the overlaps, omissions and discontinuities that may result from piecemeal funding. Our methodology will also adapt and incorporate new challenges as they emerge. The DAASE project architecture (depicted in Figure 4) is specifically designed to support the insertion, re-focussing and curtailment of sub-projects so that we can rebalance the work between the overall research strands.

Added Value through Synergy: The DAASE project architecture brings together theoretical foundations and engineering principles to fundamentally change the way in which software is engineered. One cannot have theory in isolation, nor algorithmic development without sound theoretical foundations. Even when one has both theory and algorithms in concert, widespread impact is best maximised by generalising the engineering principles that emerge through algorithmic application to a wide breadth of sub-domains. DAASE not only brings these three strands together, but it also weaves them together. Information flows through the project in a cycle from theoretical foundations through algorithmic development to engineering principles and back around to foundations, with cross cutting engineering principles posing new theoretical challenges.

Added Value through Highly Transferable Research Skill Training: Computational search has applications across a broad spectrum of industrial environments and academic disciplines. DAASE will establish a comprehensive research training programme that will provide an inter-disciplinary pool of highly trained research staff who will have computational search and optimisation expertise that can be applied in fields and industries across the EPSRC's remit.

Added Value through Accretion: Our previous larger projects acted as 'funding accretion discs', drawing in substantial additional funding. The exceptionally high level of institutional support for the DAASE project demonstrates that this powerful accretion engine has already started. The combination of critical mass EPSRC funding, backed by significant strategic institutional support will ensure that we can achieve the major scientific breakthrough of Dynamic Adaptive Automated Software Engineering.

6 Advocacy for Engineering and the Physical Sciences

We shall advocate for physical sciences and engineering through professional publications, popular science, print and broadcast media and with the provision of novel and innovative 'edutainment' apps for iPhones and other app-enabled devices. The 'Pathways to Impact' Document explains how DAASE will seek to maximise impact on all the key sectors that stand to benefit from the DAASE programme. This section focuses on our specific plans to ensure that DAASE will play a leading role in the advocating for engineering and the physical sciences.

UCL's Peter Bentley and Sue Black will lead this advocacy activity. Both are ideally suited to this role: Dr. Bentley publishes regularly in the New Scientist, WIRED and broadsheet newspapers. He has written 4 popular science books and 3 text books on Evolutionary Computation, while his appearances on BBC TV and radio, CNN, NBC and Fox News have reached a global audience numbered in the hundreds of millions. His stethoscope iPhone app has been downloaded over 3 million times and was, for a while, the world's best-selling iPhone app (out of all 250,000), demonstrating the potential to quickly generate unprecedented levels of public interest.

Dr. Black and Dr. Bentley are also ideally suited to this role from an intellectual and technical point of view. Dr. Black is well known in the software measurement and maintenance communities for her work on measurement and analysis of ripple effects [6]. Dr. Bentley is well known for his work on evolutionary computation and, of particular relevance to DAASE, his work on dynamic evolutionary computation [17]. Thus Drs. Bentley and Black combine public engagement success with complementary expertise in Software Engineering and Computational Search.

DAASE will develop a series of apps that bring to life the basic concepts of computational search and adaption, by exploiting social media interaction. Dr. Bentley will lead on the development and dissemination of DAASE apps, with Dr. Black leading on social media. These apps will be used as an educational tool to disseminate and explain the core underlying concepts of the project. For example, we can use multi-objective computational search to help friends find mutually acceptable meeting places at pubs, bars and restaurants. Similar algorithms can help users to select apps for their smart phones, balancing price, features and friends' ratings as dynamically changing multiple objectives. The use of apps in this 'demonstrative' mode has enormous potential. It can facilitate public *understanding* of science and engineering, complementing the mere *awareness* which is so often the limit achievable through traditional press releases and other forms of traditional media engagement.

Dr. Black has been a leading campaigner for public understanding and appreciation of digital technologies and has been very active in promoting and widening the participation of women in Computer Science. She is also an expert in the use of social media for awareness raising and funding accretion. Her blog reaches a wide public (with over 42,000 views) while Klout, the social media influence assessor, lists her as a 'thought leader' with a Klout score of 72. She is well known for her leading role in helping to secure a future for Bletchley Park, drawing in significant funding through outreach to traditional print and broadcast media, but also through innovative use of social networking technologies.

Dr. Black will lead a team that will organise public outreach events and activities centred on the theme of adaptivity in software engineering and automation of the software development process. We shall use our apps and case study materials from the industrial partners to illustrate the way adaptivity can lead to the next generation of products in the telecoms and automotive industries.

The public outreach events DAASE will organise will raise the level of public awareness and understanding of both the possibilities and limitations of software development. Greater public understanding is very much needed in this area; unlike other engineering domains, software is perhaps less well understood precisely because it is so intangible. However, public misconceptions about software capabilities are a cause of concern because they impinge increasingly on daily experience. The DAASE public outreach events will thus benefit the software engineering research and practitioner community as a whole.

References

[1] W. Afzal and R. Torkar. On the Application of Genetic Programming for Software Engineering Predictive Modeling: A Systematic Review. *Expert Systems Applications*, 38(9):11984–11997, September 2011.

[2] W. Afzal, R. Torkar, and R. Feldt. A Systematic Review of Search-Based Testing for Non-Functional System Properties. *Information and Software Technology*, 51(6):957–976, June 2009.

[3] W. Afzal, R. Torkar, R. Feldt, and G. Wikstrand. Search-based Prediction of Fault-slip-through in Large Software Projects. In *Proceedings of the 2nd International Symposium on Search Based Software Engineering (SSBSE '10)*, pages 79–88, Benevento, Italy, 7-9 September 2010. IEEE.

[4] S. Ali, L. C. Briand, H. Hemmati, and R. K. Panesar-Walawege. A Systematic Review of the Application and Empirical Investigation of Search-Based Test-Case Generation. *IEEE Transactions on Software Engineering*, 36(6):742–762, November/December 2010.

[5] P. Baker, M. Harman, K. Steinhöfel, and A. Skaliotis. Search Based Approaches to Component Selection and Prioritization for the Next Release Problem. In *Proceedings of the 22nd International Conference on Software Maintenance (ICSM '06)*, pages 176–185, Philadelphia, Pennsylvania, USA, 24-27 September 2006. IEEE.

[6] S. E. Black. Computing ripple effect for software maintenance. *Journal of Software Maintenance and Evolution: Research and Practice*, 13:263–279, 2001.

[7] E. K. Burke, B. McCollum, A. Meisels, S. Petrovic, and R. Qu. A Graph-Based Hyper-Heuristic for Timetabling Problems. *European Journal of Operational Research*, 176(1):177–192, 2007.

[8] C. Cadar, P. Godefroid, S. Khurshid, C. S. Păsăreanu, K. Sen, N. Tillmann, and W. Visser. Symbolic Execution for Software Testing in Practice: Preliminary Assessment. In *Proceedings of the 33rd International Conference on Software Engineering (ICSE '11)*, pages 1066–1071, Hawaii, USA, 21-28 May 2011. ACM.

[9] J. Clark, J. J. Dolado, M. Harman, R. M. Hierons, B. Jones, M. Lumkin, B. Mitchell, S. Mancoridis, K. Rees, M. Roper, and M. Shepperd. Reformulating Software Engineering as a Search Problem. *IEE Proceedings — Software*, 150(3):161–175, 2003.

[10] S. L. Cornford, M. S. Feather, J. R. Dunphy, J. Salcedo, and T. Menzies. Optimizing Spacecraft Design - Optimization Engine Development: Progress and Plans. In *Proceedings of the IEEE Aerospace Conference*, pages 3681–3690, Big Sky, Montana, March 2003.

[11] A. Finkelstein, M. Harman, A. Mansouri, J. Ren, and Y. 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.

[12] F. G. Freitas and J. T. Souza. Ten Years of Search Based Software Engineering: A Bibliometric Analysis. In *Proceedings of the 3rd International Symposium on Search based Software Engineering (SSBSE '11)*, volume 6956 of *LNCS*, pages 18–32, Szeged, Hungary, 10-12 September 2011. Springer.

[13] M. Harman. The Relationship between Search Based Software Engineering and Predictive Modeling. In *6th International Conference on Predictive Models in Software Engineering*, Timisoara, Romania, 2010. Keynote paper.

[14] M. Harman and B. F. Jones. Search Based Software Engineering. *Information and Software Technology*, 43(14):833–839, December 2001.

[15] M. Harman and P. 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.

[16] K. Lakhotia, N. Tillmann, M. Harman, and J. de Halleux. FloPSy - Search-Based Floating Point Constraint Solving for Symbolic Execution. In *Proceedings of the 22nd IFIP International Conference on Testing Software and Systems (ICTSS '10)*, volume 6435 of *LNCS*, pages 142–157, Natal, Brazil, November 2010. Springer.

[17] E. L. Martelot and P. J. Bentley. On-line systemic computation visualisation of dynamic complex systems. In *Proceedings of the 2009 International Conference on Modeling, Simulation & Visualization Methods (MSV 2009)*, pages 10–16, 2009.

[18] P. McMinn. Search-based Software Test Data Generation: A Survey. *Software Testing, Verification and Reliability*, 14(2):105–156, June 2004.

[19] S. Poulding and J. Clark. Efficient Software Verification: Statistical Testing Using Automated Search. *IEEE Transactions on Software Engineering*, 36(6):763–777, 2010.

[20] K. Praditwong, M. Harman, and X. Yao. Software Module Clustering as a Multi-Objective Search Problem. *IEEE Transactions on Software Engineering*, 37(2):264–282, March/April 2011.

[21] O. Räihä. A Survey on Search-Based Software Design. *Computer Science Review*, 4(4):203–249, November 2010.

[22] J. Wegener, A. Baresel, and H. Sthamer. Evolutionary Test Environment for Automatic Structural Testing. *Information and Software Technology*, 43(14):841–854, December 2001.

[23] D. R. White, J. A. Clark, J. Jacob, and S. M. Poulding. Searching for Resource-Efficient Programs: Low-Power Pseudorandom Number Generators. In M. Keijzer, editor, *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO '08)*, pages 1775–1782, Atlanta, GA, USA, 12-16 July 2008. ACM.

[24] S. Yoo, M. Harman, and S. Ur. Highly Scalable Multi-Objective Test Suite Minimisation using Graphics Cards. In *Proceedings of the 3rd International Symposium on Search based Software Engineering (SSBSE '11)*, volume 6956 of *LNCS*, pages 219–236, Szeged, Hungary, 10-12 September 2011. Springer.

[25] S. Yoo, R. Nilsson, and M. Harman. Faster Fault Finding at Google using Multi Objective Regression Test Optimisation. In *Proceedings of the 8th European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE 11)*, Szeged, Hungary, 5-9 September 2011. ACM. Industry Track.

[26] Y. Zhang, E. Alba, J. J. Durillo, S. Eldh, and M. Harman. Today/Future Importance Analysis. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (GECCO '10)*, pages 1357–1364, Portland, Oregon, USA, 7-11 July 2010. ACM.

[27] Y. Zhang, A. Finkelstein, and M. Harman. Search Based Requirements Optimisation: Existing Work and Challenges. In *Proceedings of International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ '08)*, volume 5025 of *LNCS*, pages 88–94, Montpellier, France, 16-17 June 2008. Springer.