Curriculum vitae

Sebastian Danicic

Contact Details

Sebastian Danicic
Department of Computing
Goldsmiths, University of London
London SE14 6NW

Tel: (+44) (0)20 7919 7171
Mobile: (+44) (0)7980 573 520
e-mail: s.danicic@gold.ac.uk
web: http://sebastian.doc.gold.ac.uk

Employment

Sept 2007–	Reader in Computer Science Department of Computing Goldsmiths College University of London London SE14 6NW
March 2000–Sept 2007	Lecturer B Department of Computing Goldsmiths College University of London London SE14 6NW
1985–2000	Senior Lecturer School of Computing University of North London Holloway Rd London N7 8DB
1981–1982	Programmer/Analyst Burroughs Corporation Feltham Middx UK
1977–1980	Programmer/Analyst NCR Corporation Dayton Ohio USA

Qualifications

1977	BSc	Pure Mathematics	II. 1	Queen Mary College, London
1981	MSc	Computation	-	Oxford University
1999	PhD	Computer Science	-	University of N. London

Research

Research Overview and Principal Contributions

My principal research contributions lie in the areas of source code analysis (notably slicing), program schemas and program dependence.

Schema Theory and its Applications

A schema (or scheme) represents a whole class of programs. Decidability of equivalence is the ability to automatically check whether two different schemas represent the same class of programs. Such problems are notoriously difficult to solve. I proved that equivalence of conservative, free, linear schemas is decidable [25] and later, strengthened this by proving that equivalence of liberal, free linear schemas is decidable in polynomial time [13]. As a result of the *Schemas project* (where I was the PI), these results were further strengthened [2, 5, 6]. This work represents significant progress in the field of schema theory and the first notable extension for decades. Furthermore, this work has important practical applications: it shows that for large classes of programs, dependence analysis and slicing algorithms can be developed which are more accurate than those allowed by previous technologies.

Software Watermarking, Slice Graphs and Dependence Communities

Software watermarking [7, 8] involves embedding a unique identifier within a piece of software, to discourage software theft. Watermarking does not prevent theft but instead discourages software thieves by providing a means to identify the owner of a piece of software and/or the origin of the stolen software. The hidden watermark can be extracted, at a later date, by the use of a recogniser to prove ownership of stolen software.

Recent work (see http://jameshamilton.eu/category/tags/research/software-watermarking) has successfully applied dependence analysis to uncover watermarks of Java programs. We have introduced various forms of *slice graph* and have shown that watermarks appear in their own *dependence communities* in such graphs. Future work will investigate how to hide such communities as well as researching into other applications of communities in graphical representations of software.

Control Dependence

Control dependence is the relationship that exists between two vertices of a control flow graph representing a program when one vertex determines whether or not the other can be executed. It is central to many program analysis and transformation techniques. I have introduced a semantics and generalisation of Control Dependence [1] which allows it to be applied to a much wider range of program structures than before. Future work will use these generalisations to apply control dependence to other structures including extended finite state machines and representations of concurrent programs.

Program Slicing and Semantics

Program slicing is a general term for a set of algorithms which transform a program into a smaller one whose effect is limited to a particular area of interest (for example, a subset of the program's variables). In order to improve the accuracy and efficiency of these algorithms it is first necessary to semantically define their desired behaviour. I have produced a number of papers which define the semantics preserved by slicing in its different forms [3, 14, 16, 17, 30, 31, 33].

In 1997 I introduced Amorphous Slicing [24], a way of reducing program size by focussing upon a point of interest within a program. Amorphous slicing has been studied and developed by many universities and research institutes and has been incorporated into industrial tools. The work originated in the EPSRC project GUSTT for which I was the CI and which was rated outstanding on completion. Work on amorphous slicing has since been funded by other bodies and companies including the NSF in the USA.

High Impact Industrial Collaboration

Ongoing industrial collaboration resulted from a very high impact KTP project with @UK PLC. The main output of this project was a spend-analysis system called SpendInsight. SpendInsight is a software system which enables organisations to efficiently re-order basic supplies based on analysis of what they already use. It employs new artificial intelligence algorithms developed during the KTP project, that automate the previously manual spend analysis process, enabling analysis to be performed with unprecedented speed and detail which have been used to highlight potentially huge savings in the organisations. SpendInsight is already being deployed at a number of organisations, bringing essential revenue to @UK. In particular, the service has already been sold to Basingstoke and North Hants NHS Foundation Trust and the NHS Share Business Service (with 128 NHS Trusts). However the

most significant impact of this research has resulted from its use by the National Audit Office. It noted in a recent report that the procurement of medical and other supplies consumables by NHS hospitals is essential to the quality of patient care and successful treatment outcomes. (http://www.nao.org.uk/publications/1011/nhs_procurement.aspx). Using *SpendInsight*, the National Audit Office have estimated that if hospital trusts were to amalgamate small, ad-hoc orders into larger, less frequent ones, rationalise and standardise product choices and strike committed volume deals across multiple trusts, they could make overall savings of at least £500 million, around 10 per cent of the total NHS consumables expenditure of £4.6 billion.

As a result of this project I have developed a good working relationship with the company and am still actively involved in collaborative research [1, 3] with @UK PLC and further projects are being developed. Future work will apply dependence analysis techniques to restructure the company's code-base.

Funded Research Projects

Date and Duration	Awarding Body	Project Title	My rôle	Value
2006 - Three years	EPSRC	Schemas: Linear Schemas for Program Dependence Ref: EP/E002919/1	**Principal Investigator** Other named participants: Mark Harman (King's) (CI) Rob Hierons (Brunel) (CI) Industrial Collaborators: Elaine Weyuker (AT &T) John Howroyd @UK PLC	£357,267.31
2006 - Three years	DTI	KTP Partnership with @UK PLC To develop a product aware web spider as part of an integrated Internet search engine. KTP Code: 1575	**Academic Supervisor** Other named participants: Mark Bishop (Goldsmiths) (Lead Academic)	£118, 309
2003 - Three years	EPSRC	TeTra Evolutionary test data generation	**Co-investigator** Other named participants: Mark Harman (Brunel) (PI) Chris Fox (Essex) (CI) Rob Hierons (CI))	£269, 522
2000 - Six Months	EPSRC	GUSST Guided Slicing and Transformation	**Co-investigator** Other named participants: Mark Harman (Goldsmiths) (PI) Jasna Kuljis (Goldsmiths) (CI) Chris Fox (Goldsmiths) (CI) Rob Hierons (Goldsmiths) (CI)	£135, 111

Research supervision

Succesfully completed:

PhD	2008-	James Hamilton (Goldsmiths)	Watermarking and Dependence Analysis
MPhil	2004-2009	Kostas Adamopoulos (King's)	Mutation Testing
PhD	2001-2006	Mohammed Daoudi (Goldsmiths)	Conditioned Slicing
PhD	2001-2005	Lahcen Ouarbya (Goldsmiths)	Semantics of Program Slicing
PhD	2001-2004	Michael Laurence (Goldsmiths)	Decidability Results in Schema Theory
PhD	1997-2003	Yoga Sivagurunathan (UNL)	Modeling Dynamic Memory using Slicing
PhD	1990-1994	George Karakitsos(UNL)	Compiler Compilers
PhD	1988-1992	Mark Harman (UNL)	Functional Models of Procedural Programs

Publications

Recent 4* Publications. (These are all papers, whose quality, to use HEFCE terminology, is world-leading in terms of originality, significance and rigour appearing in top journals. These paper are all eligible for inclusion in HEFCE REF 2014)

[1] Sebastian Danicic, Richard Barraclough, Mark Harman, John Howroyd, Akos Kiss, and Mike Laurence. A unifying theory of control dependence and its application to arbitrary program structures. *Theoretical Computer Science*, October 2011.

[2] Sebastian Danicic, Robert Hierons, and Michael Laurence. Complexity of data dependence problems for program schemas with concurrency. *ACM Transactions on Computation and Logic*, 2011. to appear.

[3] Richard Barraclough, David Binkley, Sebastian Danicic, Mark Harman, Robert Hierons, Ákos Kiss, and Michael Laurence. A trajectory-based strict semantics for program slicing. *Theoretical Computer Science*, 411:1372–1386, March 2010.

[4] Sebastian Danicic, Robert Hierons, and Michael Laurence. On the computational complexity of dynamic slicing problems for program schemas. *Mathematical Structures in Computer Science*. (Accepted May 2010. To appear in 2011).

[5] Sebastian Danicic, Robert Hierons, and Michael Laurence. Characterizing minimal semantics-preserving slices of predicate-linear, free, liberal program schemas. *Journal of Logic and Algebraic Programming*, May 2011.

[6] Sebastian Danicic, Robert Hierons, and Michael Laurence. Decidability of strong equivalence for slices of linear, free, near-liberal program schemas. *Journal of Logic and Algebraic Programming*. (Available online from 11 Aug 2010).

All Other Papers

[7] James Hamilton and Sebastian Danicic. An Evaluation of the Resilience of Static Java Bytecode Watermarks Against Distortive Attacks. *IAENG International Journal of Computer Science*, 38(1):1–15, 2011.

[8] James Hamilton and Sebastian Danicic. A Survey of Static Software Watermarking. In *Proceedings of the World Congress on Internet Security 2011*, London, 2011. IEEE.

[9] James Hamilton and Sebastian Danicic. An Evaluation of Static Java Bytecode Watermarking. In *Proceedings of the International Conference on Computer Science and Applications (ICCSA10), The World Congress on Engineering and Computer Science (WCECS10)*, volume 1, pages 1 – 8, San Francisco, USA, October 2010.

[10] James Hamilton and Sebastian Danicic. An evaluation of current java bytecode decompilers. In $9^{th.}$ *IEEE International Working Conferenece on Source Code Analysis and Maniputlation*, Edmonton, Canada, 2009.

[11] Sebastian Danicic, Robert Hierons, , and Mike Laurence. Weisers's algorithm computes minimal path-faithful slices of function-linear, free program schemas. In $4^{th.}$ *International Workshop on Programming Language Interference and Dependence*, Velncia, Spain, 2008.

[12] Sebastian Danicic, Robert Hierons, , and Mike Laurence. On the computational complexity of dynamic slicing problems for program schemas. In $3^{rd.}$ *International Workshop on Programming Language Interference and Dependence*, Lyngby, Denmark, 2007.

[13] 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.

[14] 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.

[15] David Clark, Sebastian Danicic, and Roberto Giacobazzi. Special issue on theory and foundations of programming language interference and dependence. *Logic and Algebraic Programming*, 72:123–240, July-August 2007.

[16] David Binkley, Sebastian Danicic, 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.

[17] 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.

[18] David Wendell Binkley, Sebastian Danicic, Mark Harman, John Howroyd, and Lahcen Ouarbya. A formal relationship between program slicing and partial evaluation. *Formal Aspects of Computing*, 18(2):103–119, 2006.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

[26] 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.

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

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

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

[30] David Binkley, Sebastian Danicic, Tibor Gyimóthy, Mark Harman, Ákos Kiss, and Bogdan Korel. Minimal slicing and the relationships between forms of slicing. In 5^{th} *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.

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

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

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

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

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

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

[42] Chris Fox, Mark Harman, Rob Mark Hierons, and Sebastian Danicic. Backward conditioning: a new program specialisation technique and its application to program comprehension. In 9^{th} *IEEE International Workshop on Program Comprenhesion*, pages 89–97, Los Alamitos, California, USA, May 2001. IEEE Computer Society Press.

[43] 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.

[44] 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.

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

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

[47] 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.

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

[49] Yoga Sivagurunathan, Mark Harman, and Sebastian Danicic. Slicing, I/O and the implicit state. In Mariam Kamkar, editor, 3^{rd} *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.

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

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

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

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

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

Other Research Activity

I regularly act as referee for papers appearing top quality journals and conferences and I have refereed several funding bids for large grants. I am and have been on a number of conference program committees and have been guest editor for journals. I was a founder member of the *VASTT* research group which has many collaborators at other universities in London, UK, Europe and America. The VASTT group became internationally recognised. Its work is concerned with extraction, modification and verification of systems and their components, primarily using techniques associated with program slicing and transformation. I am a founder member of Goldsmiths' Program Transformation and Analysis group and a regular attendee at the open workshops at the Centre for Research on Evolution Search and Testing at UCL.

I am currently contributing and collaborating on the Java.net project. Java.net is the realization of a vision of a diverse group of engineers, researchers and technologists at Sun Microsystems, Inc. to provide a common area for innovative development projects related to Java technology I am the leader of an open source Java software project called $jabstract$.
(See https://jabstract.dev.java.net/ for further information about the project).

I am regularly asked to examine PhDs in other institutions.

Teaching

I have taught (and supervised projects) in a wide range of subjects to BSc, MSc and PhD students. I have also developed syllabuses and developed complete new degree programmes and accreditation and TQA documents. I have produced distance learning material for various courses and pioneered new teaching methods. I am a regular examiner on the University of London External Programme. In 2003, I received a Centre For Excellence in Learning and Teaching Award at Goldsmiths'. I have developed syllabuses and taught on the following courses:

2011–	Distributed and Network Programming
2011–	Client-Side Web Technologies and Programming
2010-2011	Functional Programming
2008-2011	Object Oriented Programming in Java
2001	Language Design and Implementation (Compilers)
2000–2009	Introduction to Programming in Java
2001	Language Design and Implementation
1998	Maths for Computer Science
1996	Prolog Programming
1995	Communicating Sequential Processes (CSP)
1995	Compiler Techniques
1992	Concurrent Programming
1987	Formal Specification in Z
1985	Data Structures

Subject Guides:

[1] S.Danicic. Introduction to Java and object–oriented programming(Volume 1). *University of London External Program*, *University of London Press*, 2006.

[2] S.Danicic. Introduction to Java and object–oriented programming(Volume 2). *University of London External Program*, *University of London Press*, 2006.

Administration

I am currently the **Director of Postgraduate Research** and the **Programme Leader for BSc Computer Science** in the Department of Computing at Goldsmiths. These are two very important administrative roles with overall responsibility for a postgraduate research supervision and undergraduate teaching respectively. I am chair of the Postgraduate Research Committee and a serving member of various other college committees.

I have had many other administrative duties in the past including: Grant Workshop Organiser, Line Manager for the Departmental Systems Administrator First Year Teaching Team Leader, CIS BSc Final Year Project Co–ordinator, Deputy Director of Post Graduate Studies Departmental Web Page Organiser Departmental Labs and Equipment Officer Departmental Teaching and Learning Officer Chair Lab User Group and Open Day Organiser.

Referees

1. Professor Mark Harman
   Professor of Software Engineering,
   CREST
   Software Systems Engineering Group,
   Department of Computer Science,
   University College London,
   Malet Place, London, WC1E 6BT, UK.
   Office: GS4.01
   Tel: +44 (0)20 7679 1067 (Direct Dial)
   M.Harman@cs.ucl.ac.uk

2. Professor John Darlington,
   Director London e-Science Centre
   Imperial College London
   South Kensington Campus
   180 Queen's Gate
   London SW7 2AZ
   U.K.
   +44 (0)20 7594 8360
   j.darlington@imperial.ac.uk

3. Professor Rob Hierons,
   School of Information Systems, Computing and Mathematics,
   Brunel University,
   Uxbridge,
   Middlesex,
   UK, UB8 3PH,
   rob.hierons@brunel.ac.uk,
   phone +44 (0) 1895 266002