

# BCS<sup>TM</sup> Requireonautics Quarterly



The Newsletter of the  
**Requirements Engineering Specialist Group**  
of the British Computer Society

©2004, BCS RESG

<http://www.resg.org.uk>

Issue 33 (September 2004)

## Contents

<i>RE-Soundings</i>	1	<i>RE-Readings</i>	5
From the Editor	1	Annual General Meeting and audience with Anthony Finkelstein	5
Chairman's Message	2	The AGM and Lecture: a Personal View	8
<i>RE-Treats</i>	2	Twelfth IEEE International Requirements Engineering Conference (RE'04)	9
RE for Defence	2	RE'04: a Personal View	11
RE Books Event / Birds of a Feather Meeting	2	<i>RE-Papers</i>	12
PhD Symposium	2	Requirements and Formality	12
<i>RE-Calls</i>	3	Software Traceability: Past, Present, and Future	13
RE'05: 13 <sup>th</sup> IEEE International Requirements Engineering Conference	3	<i>RE-flections</i>	15
ICSE 2005: 27th International Conference on Software Engineering	4	Requirements for Fortifications Engineering	15
INCOSE UK Autumn Assembly	4	Necessity: Missed Key Requirements	15
ICSSEA 2004: 17th International Conference on Software & Systems Engineering and their Applications	4	<i>RESG Accounts</i>	16
ACM SIGSOFT 2004: 12th International Symposium on the Foundations of Software Engineering	4	<i>RE-Publications</i>	17
CaiSE'05	4	Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle	17
REFSQ'05	5	Book Review: Sandom & Harvey: Human Factors for Engineers	18
Early Aspects 2004: Aspect-Oriented Requirements Engineering and Architecture Design	5	Comparative Book Reviews: Mathematics Handbooks	19
International Journal on Software Engineering and Knowledge Engineering Organizations and Society in Information Systems (OASIS) 2004 Workshop	5	<i>RE-Sponses</i>	19
Call For Chapters: Rationale Management in Software Engineering	5	<i>RE-Sources</i>	20
		Books, Papers	20
		Mailing lists	20
		Requirements Engineering Journal	20
		<i>RE-Actors</i>	21
		The committee of the RESG	21

## RE-Soundings

### From the Editor

Since the last RQ came out, your committee has been entirely rearranged. You'll see that you have a new Editor; Pete Sawyer is now Chairman, and I've been 'promoted' from Reporter. I'll try to maintain Pete's high standards. You'll find some modest changes in the design and content of this RQ; comments on these, and your suggestions for further enhancements will be very welcome.

Unfortunately our roving Reporter, Elena Perez-Miñana, has temporarily left the committee, so we have an exciting job for the right person! So if you'd like to join the editorial team, let us know.

I would specially like to encourage requirements people in industry to contribute their thoughts, however preliminary and unacademic, to RQ: we're delighted to have an article on Requirements and Formality from John Warren, and a report on the AGM by Lindsay Smith in this issue. The big event in RE since the summer was of course RE'04 in the beautiful temple city of Kyoto. Read all about it in this issue.

*Ian Alexander,  
Scenario Plus*

### Chairman's Message

Regular readers of this column will know that RESG has just celebrated its tenth birthday. This was marked

by two notable events. The first, following the AGM on the 7th July, was a terrific talk by Anthony Finkelstein where he set out his personal vision of the key unsolved problems facing RE (see review later in this issue). There was a satisfying symmetry to this because Anthony gave the RESG's inaugural talk back in 1994. The second notable event was, of course, Bashar Nuseibeh's retirement as Chair. We've already sung Bashar's praises in RQ (see issue 32) and at the AGM so I won't embarrass him further here. However, you will be relieved to know that while he has passed on the Chair's baton, he remains a member of the executive as well as continuing to be a very influential RE researcher.

As usual, several other members of the executive have moved on this year. All have worked very hard indeed and we will miss them. Many thanks to Sofia Guerra, Efi Raili and Elena Perez Miñana (Industrial Liaison), Alessandra Russo (Vice-Chair) and Sebastian Uchitel (Publicity). In compensation, we are lucky to have Will Heaven who is our new publicity officer, and Alistair Mavin who joined the Industrial Liaison team mid-way through last year. Inevitably, there has also been a 'cabinet reshuffle' to ensure that all the main

posts are covered. Kathy Maitland has taken over as Vice Chair, Ian Alexander has taken over as Editor of RQ, Juan Ramil shoulders the Membership Secretary role, Steve Armstrong inherits the Regional Officer job from Kathy and I'm in the Chair. Anyone not mentioned here remains in the same role and deserves our thanks. See the RE-Actors section for roles and contact details.

The basic mission of the RESG remains unchanged, of course, and we have what we think is an exciting programme of events coming up for the coming year. It sounds clichéd to say that the RESG exists for you the members. Nevertheless, it does, so if there are events you would like us to organise, issues you think we should address, or constituencies you think we could serve better, please let us know. If you get the chance, do come and participate in our events and please support RQ with contributions. Best of all, if you are willing to help run the RESG, enthusiastic new members of the executive committee are always most welcome.

*Pete Sawyer*  
*Computing Department, Lancaster University*

---

## RE-Treats

*For further details of all events, see [www.resg.org.uk](http://www.resg.org.uk)  
Forthcoming events organised by the RESG:*

### RE for Defence

**Date:** 19<sup>th</sup> October, 1 pm

**Location:** Main Lecture Theatre, Central Facilities Building, MoD, Abbeywood, Bristol

**Programme:**

- Good and Bad Requirements - attributes and good practice for the development of requirements
- Human Factors Integration - how to ensure that a human being can actually use the capability you deliver
- Validation and Verification of Requirements - how to ensure that your Project can be accepted off contract

**To register contact:** Gill Cryer  
[PDGOM@dpa.mod.uk](mailto:PDGOM@dpa.mod.uk), Tel: 0117 91 34169

**Contact:** Gordon Woods  
([Gordon@cigitech.demon.co.uk](mailto:Gordon@cigitech.demon.co.uk))

### RE Books Event / Birds of a Feather Meeting

**Date:** 12<sup>th</sup> November at 6p.m.

**Location:** City University, London

**Programme:**

This evening event will be both an informal opportunity to meet other members of the RE

community and to meet the authors of new books on RE. In particular, Ian Alexander and Neil Maiden will be on hand to launch their new book *Scenarios, Stories, Use Cases* (John Wiley) and James Robertson will be available to talk about his new book on *Requirements-Led Project Management* (Addison-Wesley). Prof. Michael Jackson will be there with copies of his Problem Frames, and we hope several other authors and numerous interesting new titles. Free refreshments will be provided.

**Contact:** Ian Alexander ([ian@scenarioplus.org.uk](mailto:ian@scenarioplus.org.uk))

### PhD Symposium

**Date** 08 December 2004

**Venue** University College London

**Overview**

The PhD workshop aims to provide useful guidance for PhD students. This event will give students the opportunity to discuss their research in an informal environment and get advice from experts on how to conduct successful PhD research. During this one-day event there will be short student presentations, two keynote speakers, and a mock viva where students will observe workshop facilitators simulating a real viva.

**How to Participate**

Students interested to participate in this event should submit a research abstract consisting of:

- Thesis title
- Supervisor name
- The technical problem to be solved. You should justify the importance of the problem and explain why prior research has not solved the problem.

- The research hypothesis (claim).
- Sketch of the proposed solution
- The expected contributions of your research.

Explain how you plan to evaluate your results.

You should submit your abstract to Carina Alves at [c.alves@cs.ucl.ac.uk](mailto:c.alves@cs.ucl.ac.uk). The research abstract should be no longer than half a page of A4 (10pt Times please). Accepted students will have their research abstract published in the next edition of RQ. The number of

participants is limited to 12 students to ensure a highly interactive and collaborative event.

### Important dates:

November 10, 2004 Deadline for submissions  
 November 17, 2004 Notification of acceptance  
 December 08, 2004 PhD Workshop

Refreshments and lunch will be provided

## RE-Calls

*Recent Calls for Papers and Participation*

### RE'05: 13<sup>th</sup> IEEE International Requirements Engineering Conference

August 29th - September 2nd, 2005, Paris, France

<http://www.re05.org>

High-quality requirements are at the heart of successful systems engineering, in that a successful product is one that effectively satisfies some essential need. To engineer a successful product, we must build according to the goals for, functions of, and constraints on the product, with respect to the contexts in which the product will be used. This involves understanding the needs of users, customers, and other stakeholders in the product; specifying, modelling, and analyzing the stakeholders' requirements; and using these requirements to guide design decisions, to define acceptance criteria for evaluating the final product, and to constrain the product's evolution. Requirements encompass more than desired functionality - users increasingly demand systems that are usable, reliable, secure, and responsive, while product developers expect to be able to adapt and evolve products rapidly, in response to users' changing needs. As such, requirements activities are multi-disciplinary, drawing on research and experience in computer science, mathematics, engineering, human-computer interaction, and social and cognitive sciences.

RE' 05 will bring together researchers, practitioners, and students to exchange problems, solutions, and experiences concerning requirements. The conference will emphasize the crucial role that requirements play in the successful development and delivery of systems, products, and services. Topics of interest include, but are not restricted to :

- Requirements elicitation and identification
- Informal modelling of requirements
- Domain modelling
- Formal modelling of goals and requirements
- Specification languages
- Formal analysis and verification
- Multiple viewpoints, managing inconsistency
- Nonfunctional and quality requirements

- Prioritization, negotiation, and resolution of conflicting requirements
- Prototyping, animation, simulation
- Requirements validation
- Requirements evolution over time, across product families; variability requirements
- Requirements management, traceability, metrics
- Requirements methodologies (e.g., Agile methods)
- Social, cultural, and cognitive factors in requirements activities
- Aligning requirements to business goals and processes
- Relating requirements to system architecture, testing
- Requirements for COTS-based systems
- Requirements for interoperating, multi-organizational systems
- Domain-specific problems and solutions (e.g., high-assurance systems; secure systems; socio-technical systems; telecommunications and distributed systems; business and information systems)

### Papers and Evaluation

To provide a forum for exchanging practical experiences and challenges, as well as research results, we will run both researcher-oriented and practitioner-oriented technical tracks. We invite submissions of high-quality papers in the following categories:

- Technical research papers present original results from basic or applied research, including at least preliminary evaluation of results. Technical research papers will be reviewed for originality, soundness, and potential utility of the research to the state-of-the-art. Papers must not exceed 10 pages.
- Research evaluation papers present case studies, empirical studies, experiments, or analyses that evaluate scientifically the effectiveness and applicability of requirements techniques. Evaluation papers will be reviewed for quality and rigor of evaluation, and for significance and

generality of results. Papers must not exceed 10 pages.

- Reflection-on-practice papers introspectively evaluate industrial experience with using requirements techniques. Reflection papers should describe not only the requirements techniques employed, but also the setting in which the techniques were used, the results obtained, the lessons learned, and any conclusions that can be drawn. Reflection papers will be reviewed for significance of the insights presented. Papers must not exceed 10 pages.
- Industrial practice reports pose requirements problems or challenges encountered in practice, relate success or failure stories, or report on industrial practice. Industrial-practice reports must not exceed 5 pages. Authors who want to submit longer papers should consider submitting a reflection paper.

Papers must describe work that has not been submitted to or presented at another forum. Revised versions of a selection of best papers from the conference will appear in a special issue of the Requirements Engineering Journal. We seek also proposals for workshops, tutorials, panels, research tool demos, and poster presentations.

## Important dates

Paper abstracts (technical, evaluation, reflection papers): 07 February 2005

Paper submissions (technical, evaluation, reflection, practice papers): 14 February 2005

Notifications sent to authors: 22 April 2005

Camera-ready papers received: 03 June 2005

Workshop, tutorial, and panel proposal submissions: 11 March 2005

Doctoral symposium submissions: 28 April 2005

Poster and research demonstrations submissions: 28 April 2005

## ICSE 2005: 27th International Conference on Software Engineering

15-21 May 2005, St Louis, Missouri, USA

<http://www.icse-conferences.org/2005/>

[http://www.cs.wustl.edu/icse05/Downloads/ICSE05\\_CFP\\_General.pdf](http://www.cs.wustl.edu/icse05/Downloads/ICSE05_CFP_General.pdf)

We invite you to submit a paper, workshop or tutorial proposal, research demonstration and more to ICSE 2005—the premier forum for researchers, practitioners and educators to present and discuss the most recent ideas, innovations, trends, experiences, and concerns in the field of software engineering.

The theme of ICSE 2005 is Software Everywhere. It acknowledges the increasingly important role software plays in the life of our society through the technology that sustains it. The theme also highlights the growing level of responsibility our profession and its members are expected to assume. As such, an important goal of this meeting will be to reach out to other disciplines that have an impact upon or benefit from software engineering know-how.

General Chair: Gruia-Catalin Roman (Washington University in St. Louis, USA)

Program Chairs: William Griswold (University of California, San Diego, USA); Bashar Nuseibeh (The Open University, UK)

## DEADLINES

4 Oct 2004: Tutorial and workshop proposals

6 Dec 2004: Doctoral symposium, Research demonstrations

*Other events and calls likely to be of interest to RESG members:*

## INCOSE UK Autumn Assembly

will be held on 8-9 November 2004 at the Marriott Hotel, Pipers Way, Swindon, SN3 1SH, England.

The theme this year is “Risking It All”, looking back over 10 years of INCOSE in the UK.

<http://www.incose.org.uk>

## ICSSEA 2004: 17th International Conference on Software & Systems Engineering and their Applications

Paris - November 30 - December 3, 2004

[www.cnam.fr/CMSL](http://www.cnam.fr/CMSL)

## ACM SIGSOFT 2004: 12th International Symposium on the Foundations of Software Engineering

October 31 - November 6, 2004  
Hyatt Newporter, Newport Beach, California, USA  
<http://www.isr.uci.edu/FSE-12>

ACM SIGSOFT 2005 will be held on September 5-9, 2005 in Lisbon, Portugal, alongside the 10th European Software Engineering Conference.

## CaiSE'05

13-17 June 2005, Porto, Portugal

<http://www.fe.up.pt/caise2005/>

## REFSQ'05

will be held in connection with CaiSE'05.

<http://www.refsq.org>

**Early Aspects 2004: Aspect-Oriented Requirements Engineering and Architecture Design**

24<sup>th</sup> October 2004, Vancouver, Canada  
<http://www.early-aspects.net/events/oopsla04ws>

**International Journal on Software Engineering and Knowledge Engineering**

Special issue on software traceability

Contacts: Dr. George Spanoudakis  
[gespan@soi.city.ac.uk](mailto:gespan@soi.city.ac.uk),  
Dr. Andrea Zisman [a.zisman@soi.city.ac.uk](mailto:a.zisman@soi.city.ac.uk)

**Organizations and Society in Information Systems (OASIS) 2004 Workshop**

<http://www.ifipwg82.org/calls/oasis2004call.php3>

**Call For Chapters: Rationale Management in Software Engineering**

A book edited by Allen H. Dutoit, Raymond McCall, Ivan Mistrik, and Barbara Paech, to be published by Springer-Verlag/Computer Science Editorial.

Chapter proposals to Ivan Mistrik or Barbara Paech by 15<sup>th</sup> October 2004.

[mistrik@ipsi.fraunhofer.de](mailto:mistrik@ipsi.fraunhofer.de)  
[paech@informatik.uni-heidelberg.de](mailto:paech@informatik.uni-heidelberg.de)

---

**RE-Readings**

---

*Reviews of recent Requirements Engineering events.*

**Annual General Meeting and audience with Anthony Finkelstein**

**Date:** 7<sup>th</sup> July at 1.30pm

**Location:** Imperial College, London

**Bashar Nuseibeh** welcomed us to the 10<sup>th</sup> Annual AGM of the RESG, his swansong after a decade in the driving seat, and reported on the year's activities with his usual warmth and enthusiasm, as follows:

**Events (since last AGM, July 2003):**

Requirements Day (R-Day) 31<sup>st</sup> March 2004, Thistle City Barbican, London

Engineering Organisational Solutions from Human Information Requirements 12<sup>th</sup> May 2004, University College London

Requirements engineering training: the who, the how, the what 3<sup>rd</sup> December 2003, Imperial College London

Creativity – the path to innovative requirements 9<sup>th</sup> October 2003, Imperial College London

Planguage – a planning language for requirements and more, 17<sup>th</sup> September 2003, Imperial College London

Stakeholders without Tears: an exploration of project sociology, 16<sup>th</sup> July 2003, University College London

Distinguished Speaker Lecture 7<sup>th</sup> July 2004.

**Related/Co-sponsored events:**

Early Aspects Workshop at AOSD-04, March 2004, Lancaster

Mastering the Requirements Process, 3-day seminar/workshop, 6-8 Oct 03, London

Extending Requirements: A Practical Workshop, 2-day workshop, 17-18 Nov 03, London

RE'04, 6-10 September 2004, Kyoto, Japan

**Future Events (tbc)**

*(see RE-Treats for events coming up in 2004)*

RE for medical informatics, Lancaster, 26<sup>th</sup> Jan 2005.

i\* workshop, London, May 2005.

AGM & Distinguished Speaker Lecture, London, July 2005

**Newsletter: Requireonautics Quarterly (RQ)**

Editor: Pete Sawyer (until July 2004)

Reporter: Ian Alexander (until July 2004)

Four issues (RQ29-32) between August 2003 and July 2004.

Articles from readers always welcome!

No formal review process but editor will work with authors if editing is necessary.

Free books for contributions – while stocks last!!

Back issues on the RESG website – [www.resg.org.uk](http://www.resg.org.uk)

**Industrial Liaison**

Almost all RESG events bring together researchers and practitioners

**Industrial Liaison Team:**

David Bush (NATS) – RESG secretary  
Wolfgang Emmerich (UCL & Zülke)  
Alistair Mavin (Praxis Critical Systems)  
Elena Perez-Miñana (Philips)  
Gordon Woods (Independent)  
Suzanne Roberston (RESG Roving Ambassador!)  
Michael Jackson (RESG Patron)

**The RESG in the regions**

Yikes! Our regional event was moved from Manchester to London - apologies.

At least two events outside London planned for the coming year ...

Suggestions for regional venues welcome!

## Membership

Current membership: 294, of which Individuals, registered as BCS/IEE members: 83; Non BCS/IEE members: 32; Students: 36. Corporate – Industrial: 60 (in 5 companies); Academic: 83 (in 14 universities)

Last year's membership: 348; this means Individual membership stable, Academic numbers down.

Aims for 2004-5: Increase academic membership; try to recruit from other BCS groups; incorporate changes in BCS membership policy, where appropriate

For comments and suggestions please contact: j.f.ramil@open.ac.uk

## Publicity Report

Email distribution list has 982 addresses (up from 850 last year). Approximately 550 are non-members.

Policy: Email publicity used sparingly; some not so RESG related events are published on the web rather than forwarded.

Comments and suggestions contact Sebastian Uchitel (s.uchitel@imperial.ac.uk)

## Treasurer's Report 2004

Neil Maiden prepared the accounts (published in this RQ). They show a healthy and stable financial situation for the group.


## Election of RESG Executive Committee

Patron: Michael Jackson (Independent)  
 Chair: Peter Sawyer (Lancaster University)  
 Vice Chair: Kathy Maitland (UCE at Birmingham)  
 Secretary: David Bush (NATS)  
 Treasurer: Neil Maiden (City University)  
 Membership Sec: Juan Ramil (The Open University)  
 Publicity Officer: William Heaven (Imperial College)  
 Newsletter Editor: Ian Alexander (Scenario Plus)  
 Regional Officer: Stephen Armstrong (The Open University)  
 Student Officer: Carina Alves (UCL)  
 Immediate Past Chair: Bashar Nuseibeh (The Open University)  
 Industrial Liaison Team:  
 Wolfgang Emmerich (UCL & Zulke)  
 Alistair Mavin (Praxis Critical Systems)  
 Suzanne Robertson (Atlantic Systems Guild)  
 Gordon Woods (Independent)

**Everyone** applauded Sofia Guerra, Elena Perez-Miñana, Alessandra Russo and Sebastian Uchitel who are stepping down from the committee.

Anthony Finkelstein (UCL) proposed and Ken Mycock (SYScribe Consulting) seconded the election of the new committee. The motion was carried unanimously.

**Pete Sawyer**, our new Chairman, thanked Bashar, pointing out that the vote of thanks to departing committee members had omitted his own name. Without further ado, he presented him with an engraved plaque recording our thanks. **Kathy Maitland**, our new Vice-Chair, also thanked Bashar for his ten years of chairing the group and presented him with a large cake marked 'RESG'. Bashar was suitably embarrassed and hastily introduced Anthony Finkelstein.

—————  —————

**Anthony Finkelstein** (UCL) gave a talk on **Unsolved Problems in Requirements Engineering**. He thought an AGM a good opportunity to take a broader but entirely personal look at our field. Unsolved problems include those not worth solving, along with those that look harder than we first thought — some of which are thereby moving from the solved back to the unsolved pile! There are far more unsolved than solved; many choice plums that remain unsolved; and too many distinguished research efforts have solved problems not worth the solving.

Fred Brooks' 'No Silver Bullet' distinguished 'accident' from 'essence' in software development: we've done much more on the former than the latter.

Our problems are often not soluble within our domain: we can rightly pass some on to other disciplines. But we have done rather well and have good prospects.

There are some 'grand challenges': big problems of public appeal that can yield to community endeavour, and have clear success criteria — like the human genome project, for instance. But such things aren't the be-all-and-end-all of research.

We've solved the task of creating the components of a sound, industrial strength, RE process for small, medium and large scale developments: tools, modelling techniques, practices and standards. We are less sure how to put these bits together, and we have failed to get enough practitioners to adopt them. This may be because

- 1) we lack a clear 'economic theory' of RE to let us reason about its costs and risks
- 2) we lack proven training methods and materials to support RE knowledge transfer.

We've done well on formal modelling: we can build good-sized formal specifications and we can analyse them. But

- 1) theorem proving still requires very high skill (unlike model-checking)
- 2) we don't know specially well how to formulate properties to reason against.

We have developed/begged/borrowed/stolen a range of elicitation techniques, e.g. from HCI (e.g. the work of Gordon Rugg, Neil Maiden). But they largely remain research tools applied only on a small scale and only one at a time: we haven't stitched them into a process.

We have done very well expressing and reasoning about non-functional properties like reliability, security, and performance. But there's much further to go, and we run slap-bang into the really BIG unsolved problem in RE and software development: compositional reasoning. Jeff Kramer has helped to make this clear. Given that almost all systems evolve and get extended, this is a major headache.

Everyone pays lip-service to RE as a lifecycle-wide activity, yet most work implicitly assumes it's all up-front. We need to build RE processes that are properly integrated into the wider software development process. E.g. the USDP or the RUP are really very light on requirements.

We still don't understand the relationship between requirements and system architecture. We do know that requirements are the cornerstone of software development. He presented Bashar's 'twin peaks' drawing, in which a spiral of 'specification' loops through the two triangles of requirements and architecture as the level of detail increases: but we need to make it more than a drawing, part of our practice to inform design by thinking about requirements. (Applies to systems too —Ed.)

We don't understand well how to 'bound' systems, so we make ad hoc decisions that either set the scope too narrowly (over-constraining the solution) or too widely (wasting money on gathering incomplete requirements). Many projects fail for lack of proper bounding. The current NHS information system exercise looks conspicuously badly bounded and is at serious risk.

We're still very poor at estimating cost and effort needed for a system. If we're lucky we made a similar one earlier. Otherwise, function points are "precious little assistance". They mainly help with modelling schemes no longer widely used, and they demand masses of calibration data that are often unavailable.

We don't know how to scale down projects and their RE components when budgets get squeezed and deadlines get shortened. And we don't know how this affects project risk. (Or shouldn't we spend more on RE in such risky situations? —Ed.)

Requirements are rarely cut-and-dried. Rather there's a bunch of competing priorities, preferences and expectations, from a mixture of stakeholders playing different roles. We've fiddled about with this problem by nicking ideas from decision theory. But we need new ways of dealing with the concept of value (of needs to people) as a first-class object in RE. Perhaps only Kevin Sullivan's Option Theory has made any impression on this problem.

Change is a fundamental feature of all software development processes. Yet we understand requirements volatility very little. We must have accurate impact analysis procedures and tools. (This is a system problem too —Ed.)

Some 'new' kinds of system are specially hard for RE: adaptive systems; systems embedding COTS; systems involving large amounts of user scripting and 'plug-and-play' extensibility (e.g. large ERP systems, not to mention programming systems themselves). An up-front specification just isn't very helpful when the requirements constantly change as modules are added.

We'd like to make systems that can 'reflect' their own requirements and how far they are currently being satisfied — the requirements become run-time artefacts that systems can reason about. Run-time monitoring makes a small contribution to this. Steve Fickas and Martin Feather have contributed here.

The old story is that we need to preserve knowledge and experience in RE so we can reuse it: domain modelling, RE apprenticeship, requirements cliches, component-based modelling schemes are all partial attacks on the problem, and all are flawed. Michael Jackson's Problem Frames are the boldest attack on reuse but it's merely a sketch of a possible approach: it's not clear that it scales up, for instance.

Finally, we know that some programmers are much better than others, regardless of training and experience. Does the same apply to RE and to architecture? It would be good to understand this better. Perhaps we could get a cheap boost to productivity from this (whether by selecting the right people, or by improving training).

There's plenty to do. We need to face up to the big questions and step out of the existing conceptual frameworks of what the historian of science Thomas Kuhn (*The Structure of Scientific Revolutions*, 1962) called 'Normal Science' (and incidentally, we've done well to make RE into such a respectable discipline). We've been good at borrowing from ethnography, formal methods, psychology: that's good — if someone has done something before, you ought to use it! But these new problems won't, thinks Finkelstein, yield to borrowing. We need to devise our own approaches.

**Gordon Woods** asked if AF thought that engineering the requirements for software was different from doing those for systems. AF said yes, he did: each domain was different. Much is shared, but software has its own issues (like compositional reasoning).

**Jeff Kramer** thought the presentation very insightful, but he missed the question of involving clients, stakeholders themselves, in the process: e.g. David Harel's 'playing-in' technique (in his book *Come, Let's Play* —see the book review in RQ31). AF said yes, inevitably having (surrogate) people interposed between real requirements and systems the more attenuated the signal became. We do want to get people more directly involved.

**Tom Gilb** disagreed with some of the supposedly unsolved problems. E.g. scaling-back is handled well by prioritisation —as long as we don't try to make requirements into stand-alone 'shall' statements. AF

replied that the trouble was not prioritising requirements (artefacts) but on cutting back RE activities: how much risk the activity was buying you out of. Tom Gilb said impact analysis could be done well with his impact tables; and with evolutionary project management feedback tools. AF said that nothing would satisfy him more than crossing problems off his list. Tom Gilb finally said that storing RE knowledge for reuse was possible with his templates (which themselves capture stored knowledge — as do his books). Arguably, RE knowledge is already stored. AF agreed that important contributions had been made, but that the whole issue had not been addressed.

© Ian Alexander 2004

## The AGM and Lecture: a Personal View

by Lindsay Smith

**Bashar Nuseibeh** chaired the AGM part of the event and kept it to the allocated time, half an hour, at the start of the event. This in itself impressed me; in my experience of such meetings, this is unusual. It did occur to me that the absence of the treasurer, who was on holiday, might have stopped the ‘accounts’ from slowing down the proceedings. As a relatively new member of the group, I joined this year, I found the AGM gave me some insights into how the RESG is organised and the people doing the work.

This event was also the 10<sup>th</sup> Anniversary of the inception of the RESG and Bashar Nuseibah was chairing the AGM for the last time. He is ‘retiring’ from his role as chair after putting in effort for those 10 years to help run the RESG. I certainly got the impression that he is highly thought of. The presentation of card and cake by other committee members and the applause in recognition of his work for the RESG came under AOB that seemed somehow appropriate. Bashar appeared suitably surprised.

**Anthony Finkelstein**, who spoke at the first RESG meeting 10 years ago, looked at the areas in RE that still, in his opinion, need significant work. What followed was somewhat of a ‘tour de force’ of RE progress to date.

He approached this by way of a summary of RE problems into solved and unsolved categories. He also acknowledged the corollary to this is that both categories contain problems not worth solving. I found that interesting and expected to have some examples of what he thought was not worth solving, but to my mind, he covered the former rather than the latter in his talk. I found what he did have to say fascinating despite this perceived omission; I just *am* that ‘one’ you get in any audience.

On balance, the majority of the ‘solved’ category appeared on the technical side, e.g. tools, techniques, etc. In the ‘partially solved’ category was incorporation of these into the development process.

For example requirements effort is still ‘front loaded’ in the life cycle and needs to be ‘stitched into the fabric’ of the development process.

Unsolved issues tended to cluster around client, practitioner and system boundary. A client wants to know what they can get for their money, but practitioners do not have the means to answer that. Being able to predict project cost and resources reliably is still unresolved. Taken overall the presentation had quite a strong commercial/ economic theme.

Commercial factors affect practitioner take up of a new technique/tool. This is an example of a partially solved problem. A wide range of elicitation techniques have been developed but not applied except in research. There is need for an economic incentive, e.g. benefit gained to offset effort to solve this problem. The expansion of teaching and learning for practitioners of requirements needs similar incentive.

*Ad hoc* decisions on system boundaries can result in setting the scope of investigation too narrow. Therefore constrain a solution, else too wide a scope and it becomes too generic and expensive. Function points are not of much use here. Anthony Finkelstein suggested an economic theory needs developing to resolve issues such as how to achieve scale back on requirements due to time/cost constraints.

Apart from the commercial perspective and associated economic value placed on RE, Anthony Finkelstein spent some time on the unresolved problem of understanding the relationship between requirements and system architecture. This he posed as the ‘Twin peaks’ problem. This is apparently a cornerstone of software development where the detail is unknown.

Anthony Finkelstein suggested that RE needs a new way to value requirements, which are typically subject to competing priorities, negotiation and trade off. Connected to this is the unsolved problem of requirement volatility and how to understand the nature and consequence of change, e.g. impact analysis.

He identified unresolved problems due to system development not being static. For example, new classes of systems that can adapt to context and users can adapt by plug ability and scripting. In addition, component embedded in systems (COTs).

Inevitably, at the end of the talk, the contributions from the floor tended to fixate on specific areas of interest. This depended on whether or not the speaker in question agreed with Anthony Finkelstein’s observations.

I found the part of the discussion that focused on the system not just as a development artefact but also a run-time artefact of interest. A robust system needs to be able to deal with context changes. A possible requirement could be that a system should give a dynamic account at run-time.



It was also discussed how it is possible to preserve knowledge and experience in requirements engineering. Numerous proposed solutions have proved flawed. It was pointed out that the training and aptitude of individuals varies for programming as it does for RE.

The wide range of the issues reinforced my opinion that various perspectives are appropriate for investigation into requirements problems. Personally, I found this encouraging because it reflects the RESG's ability to acknowledge this as a group even if individual views do not entirely concur.

© Lindsay Smith 2004  
[thimsl@tiscali.co.uk](mailto:thimsl@tiscali.co.uk)

**Twelfth IEEE International Requirements Engineering Conference (RE'04)**

6<sup>th</sup> - 10<sup>th</sup> September 2004, Kyoto, Japan

<http://www.re04.org>

RE' 04 took place in Kyoto, Japan, for the first time breaking the Europe/North America cycle. It was very well attended with some 300 delegates.

The main conference consisted of a carefully-balanced mixture of keynote talks, research papers, industrial talks, mini-tutorials, vendor presentations, a poster session, and invited state-of-the-practice talks. (See the separate report by Carina Alves.)

In addition to the main programme, RE'04 had several associated workshops on specific themes:

- 2nd International Workshop on Comparative Evaluation in Requirements Engineering (<http://www.di.unipi.it/CERE04/>)
- 2nd International Workshop on Requirements Engineering for COTS Components (<http://www.lsi.upc.es/events/recots/>)
- International Workshop on Requirements Engineering Patterns (<http://rep04.desy.de/>)
- RHAS ` 04 - International Workshop on Requirements for High Assurance Systems (<http://www.sei.cmu.edu/community/rhas-workshop/>)
- International Workshop in Service-oriented Requirements Engineering (<http://www.elet.polimi.it/conferences/sore04/>)
- International Workshop on Automotive Requirements Engineering (<http://www.seto.nanzan-u.ac.jp/~amikio/NISE/AuRE/>)
- International Workshop on Principles of Software Evolution (<http://iwpsc04.wakayama-u.ac.jp/>)

There was also a full tutorial program, with half or full day tutorials on Requirements-driven product line development, Requirements-based product line

engineering, Developing practical scenarios, Software traceability, Requirements: the bridge between business and development, Requirements and creativity, Financially informed requirements prioritization, and in Japanese: Introduction to scenario analysis, and Requirements traceability methodology and its application to UML. All of these are notably practical, emphasizing the determination in the RE community to disseminate such knowledge as we have to a wider audience.

-----  
 As the conference proceedings contain far too much to discuss, let me just pick a few papers that interested me. Two questions that have been waiting for an answer were addressed in the conference, even if "full" solutions may take many years to arrive.

Soren Lauesen of the IT University, Copenhagen, contributed a paper on COTS Tenders and Integration Requirements that looked directly at the mismatch between traditional RE practices and COTS-based development. 'Requirements for integrating the new COTS system with other systems are particularly hard because suppliers may integrate in different ways and with different other systems.' In other words, such requirements are close to impossible to frame in a solution-independent way. 'A related problem is that once the new COTS system is purchased, the COTS supplier may have a de-facto monopoly.' Other awkward issues include the fact that 'mandatory requirements don't work': if the COTS products you are looking at just don't do what you want, you must either give up (and commission a costly bespoke solution) or accept that you won't get everything you want. You may have done everything by the book, but once you have chosen a supplier, many other decisions are silently committed to, and they are hard to reverse. Interoperability and tool migration are easy to dream about, but hard to achieve.

M. Hoffman and colleagues at DaimlerChrysler have put together a catalogue of requirements for RM tools, so that one day it might be possible to select tools rationally and to migrate data freely between them. As illustration, they compared DOORS and Caliber, and found that the concepts underlying these two tools are quite different, making migration surprisingly hard.

DOORS had 'no global types' and 'no inheritance' (not exactly true, but one sees what they meant); this led to workarounds based on 'fake types using user-defined attributes like "Object Type"'; its views were nice but only for specific modules; links were typed, but again only available locally; it had excellent baselining facilities.

Caliber had global types, but strictly segregated these by subtree, thus creating 'de facto modules'. The tool had no view concept, just visibility rights, and no real document-oriented view either; but it handled both internal and external (to other tools) traces the same way on the same graph, which was very nice. Its links

were untyped, which was limiting; and ‘link structure was not saved in baselines’, so that snapshots of a project’s requirements at different times were ‘seriously incomplete’.

Clearly, much of the work setting up a project in either of these tools would have to be redone to migrate the data to the other tool.

Hoffman’s vision is of a standard feature set that RM tool users could rely on – though it must be said that in other areas (word processing, spreadsheets, CAD/CAM, databases) agreement has been reached only by natural selection and the survival of the fittest products.

A. Padula of Hewlett-Packard looked at RM process selection: the deep and complex issue of how you pick the right process for different types of project. It would be too much to expect a general approach in one short paper; sensibly, he described instead two sorts of project carried out by his department, namely light, agile ones and long, outsourced, complex ones. The light ones did without formal traces which he said gave a poor return on investment in short projects (how often would anyone go back and use the traces to assess impact in such a project?). They also depended heavily on a pair of people working closely together: a business/domain expert and an analyst/requirements person. The heavier type of project used ‘vignettes’ to characterize customer needs, describing the ‘inner dialogue’ to drive the requirements. He agreed with a questioner that the two approaches would not be universally applicable as they depended largely on the business organisation, the available skills, the domain, the degree of risk, the size of problem and the timescale, and not least the corporate culture. What of course one would like to do would be to create a matrix on all those dimensions and identify the exact mix of techniques (and the project life-cycle) to use in each case.

J. Doerr, B. Paech and Mathias Koehler looked at RE process improvement based on an information model. The basic idea was to describe the structure of a project from the points of view of different audiences (customer, customer project, platform project) and at different levels of abstraction (marketing view, user view, system view, software view). Each cell of such a matrix contains one or more documents, such as a set of use cases, or a system specification. You’d also have a table giving details of roles, and another for documents, and a description of a change process. Dependencies between documents could then be made explicit; roles could be made clear; and processes could be flexible, without rigid sequencing. This made templates easy to create, and also showed which documents and roles were overloaded! In their case, the *Pflichtenheft*, a key document in many German organisations, turned out to have overlaps with many other documents, though the paper did not critique the *Pflichtenheft* concept or propose a better way of assembling it.

Speakers had to contend with the attractions of Kyoto itself, and the surrounding area (beautiful wooded hills, carefully preserved from the encroaching urban sprawl of Osaka, and the ancient cities of Nara and Kobe nearby). Somehow, despite the 26 World Heritage Sites including the Golden Pavilion, the Imperial Palace and the largest wooden hall in the world, people found the time to hear about the state-of-the-art in Scenarios (at 4:45pm, too – I’m grateful for their patience) or to discuss Future Challenges in RE.

It was impossible not to notice the distinctive character of Japan: crowded cities; very efficient services; polite people; and miniaturisation of trees, gardens, houses, and of course electronics. Somehow this set the requirements for tiny, shining, jewel-like handheld devices clearly into perspective.

Next year’s conference will be in Paris: easier to reach at least for Europeans, but offering just as many temptations. Let’s hope that the Paris conference shares ideas that are just as exciting as those at Kyoto.

© Ian Alexander 2004



*Carina Alves has kindly contributed the following personal report from the conference in Kyoto.*

## **RE’04: a Personal View**

RE’04 was a stimulating conference; our Japanese hosts showed competence and enthusiasm in organizing this event. One of the conference social events was the banquet where we had the opportunity to taste the exquisite and sophisticated Japanese cuisine. Conference participants were delighted to watch *maiko* (apprentice geishas) perform traditional music and dance. After the performances, most

participants took pictures with them to bring these memories back home.

The conference program was as exciting as this evening. In the first day, Prof. Nigel Cross gave an inspiring keynote presentation entitled 'How Creative Design Happens'. He discussed the challenges of creative thinking in design from the perspective of engineering processes. Frequently, product designers face the problem of creating new product ideas from inexistent requirements, as states this quote from Dewys Lasdon: 'Our job is to give the client not what he wants, but what he never wanted and when he gets it, he recognizes it as something he always wanted'. He looked at examples ranging from different domains, such as formula one racing cars, sewing machines and litter bins, where designers are challenged to create innovative solutions to improve the way people use these products. However these examples come from very diverse domains, Nigel showed some striking similarities between them in terms of the creative strategies taken by designers. This suggests that fundamental principles drive creative design. For instance, three key processes recur in all examples studied by Nigel: designing from "first principles" that drive the physical world; frame the problem from the requirements of a particular design situation which sometimes can be highly influenced by personal motivations; and finally take a wide view of the problem rather than relying on narrow, pre established concepts about the world.

A quite imaginative presentation was the mini-tutorial 'He's Just Making It All Up!' (or 15 Techniques to Make Requirements More Creative and Innovative) where Neil Maiden and troupe improvised a comic show, presenting techniques to create more inventive requirements. James Roberson, Don Gause, Suzanne Robertson and the hilarious Dan Berry were competing to present the most creative techniques. They asked the audience to create a problem to be solved. After few suggestions, the most voted problem was 'how to visit the best attractions in Kyoto', which I believe was a real problem faced by most attendees. Each presenter as well as the audience tried to come up with solutions; some of them were rather absurd like visiting the temples of Kyoto by kangaroo, so that in a single step he could transport you from temple to temple. This was a cheerful and stimulating session to show us how to think outside the box and encourage more creative thinking about requirements.

The best paper award was given to Soren Lauesen for the valuable paper 'COTS Tenders and Integration Requirements'. As Ian has discussed in his report, this paper addresses a key problem when developing systems from COTS products, that of specifying and negotiating requirements and how to conduct the tender process. Another award was given to the most influential paper presented at the 1<sup>st</sup> International Conference on Requirements Engineering, ten years ago. Orlena Gotel and Anthony Finkelstein were awarded with the seminal paper 'An Analysis of the

Requirements Traceability Problem' presented at RE'94. As Andrea Zisman and George Spanoudakis confirms with a paper published in this issue of RQ, the requirements traceability problem remains a relevant topic in the requirements engineering research agenda.

Axel van Lamsweerde gave an insightful keynote talk on the state of the practice of goal-oriented requirements engineering (GORE). Axel started his talk giving an overview on what has been done in the research arena of GORE, for instance, extensive work has been done in the areas of goal operationalization, conflict and obstacle management, and reuse of goal taxonomies and specifications. Much of these ideas have been successfully applied in industrial settings as he gives examples of industrial projects in a wide variety of domains that have applied the KAOS method. He highlighted that the success of these projects depends on effective tool support, which in their case is supported by the professional version of GRAIL prototype. Over the years the models, techniques and tools developed from research projects were continuously refined, simplified and improved to suit an industrial environment. A rich, well-structured goal graph has proved to be the ideal mechanism to allow communication between business managers and software engineers. Axel concluded his presentation by discussing challenges faced by researchers in order to transfer research results to industrial practice. One of the key barriers are the lack of reliable measurement of the benefits of using RE techniques as well as the uncertainty concerning the economics of the RE process.

Following the Japanese principles of harmony, this conference proved to be a good balance of motivating program, interesting attendees to share ideas and experiences and the beautiful city of Kyoto.

© 2004 Carina Alves

## RE-Papers

### Requirements and Formality

Too many software and information system projects fail, or at least are less successful than they might be. One survey<sup>1</sup> reports that many projects (31%) are cancelled before delivery, that many more projects (53%) contain not only less than their specified functionality (on average, 60%) but also exceed their timescale (by 220%) and/or cost (by 189%); and that only 16% of projects (one in six) deliver their specified requirements to time and budget. A separate survey<sup>2</sup> reported that only 13% of projects were successful and that in 75% of failing projects, the most important cause of failure was "unclear requirements". Requirements engineering is therefore a massively important discipline, for at least five out of six projects; the question arises: how do we ensure good requirements?

The objective of a requirements definition is to communicate the users' requirements, clearly and correctly, to those who will design, build, and test the proposed system. If these requirements are not both comprehensible and correct, then the document has failed to meet its objective. To achieve communication, the document must be written in some language; natural language, enhanced by diagrams and with computer-aided software engineering (CASE) tool support for the chosen approach, is most common.

It is crucially important that requirements be correct. Certainly a requirements definition should have other properties (it should be complete, consistent, concise, unambiguous...) but if it is correct, then it is likely that most of the other properties will follow; for example, an inconsistent definition cannot be correct. A correct specification is one that exactly captures the users' requirements. I assert that the most practical way to ensure correctness is to test the document in some way (animation or prototyping, for example) so that users can confirm that the recorded information correctly represents their needs.

An early example of the value of this type of testing occurred in an airline ticketing specification, where the stated requirement was: "To ticket passengers between any two airports". The very first animation provided a ticket from Heathrow to Heathrow and elicited the comment "When I said two airports, what I meant was two *different* airports!". Such situations are all too common.

If a requirement definition is to be demonstrably correct, the language in which it is written must be

sufficiently formal to support either direct execution or automated reasoning. Such languages tend to be mathematical and are not easily understood other than by experts. They are therefore generally not adequate for communication with users; a possible solution to this difficulty is to conceal the formality inside a toolset that interacts with users in a friendlier manner. The formality and mathematical reasoning can then be completely hidden; the benefits of formality are obtained without the need for expertise in the use of formal methods.

There are two major benefits that accrue from this approach. One is that requirement reasoning leads to a greatly improved understanding of the users' needs. A recent example of this occurred when I prepared the requirement definition for a database project; one of the business reasons for the new system was the unreliability of its predecessor. An analysis of that system showed exactly how the unreliability arose and I was able to write the implied requirement for that system (with which the client agreed). Subsequent animation showed exactly the condition under which a single resource could be simultaneously allocated to more than one user. The present definition contains a formal (mathematically provable) guarantee that such situations can never arise.

The other major benefit is that of rule-based reasoning. This allows large amounts of expertise to be captured within rules, so that the reasoning system can represent the combined skill of many experts. This leads to requirement documents that are of very high quality, because they offer guarantees that are otherwise unavailable.

It would be nice to say that such rules as are already widely known will be universally applied, but this is not so. I have twice stood as part of an Expert Witness team where a failing software project has eventually become subject to litigation. In both cases, reputable software companies had produced database designs that failed to conform not only to standard normalisation rules but even to the basic rules for entity and referential integrity.

I urge that requirement engineers consider all possible ways that may be open to them to ensure that the documents they develop are correct and comprehensible to all. I regard formal methods as important in this respect; though it may be necessary to conceal their use from users, the benefits are substantial. The more that formal reasoning can be applied to requirement definitions, and subsequently to specifications, the sooner we can start delivering systems that genuinely and reliably satisfy users' needs and deliver what the user really wanted.

© John H. Warren

([john.warren@precisiondesign.co.uk](mailto:john.warren@precisiondesign.co.uk))

<sup>1</sup> "The Chaos Report", <http://www.standishgroup.com>

<sup>2</sup> "The BCS Review 2001", <http://www.bcs.org.uk/review/2001/html/p061.htm>

## Software Traceability: Past, Present, and Future

Andrea Zisman and George Spanoudakis  
 Department of Computing, City University  
 Northampton Square, London, EC1V 0HB

Software traceability is the ability to intra- and inter-relate software artefacts created during the development of software systems in order to improve the overall quality of these systems. It supports various activities in the software system development process, including change impact analysis, maintenance and evolution of systems, validation and verification analysis, reuse of software systems components, testing and compliance of software systems, system inspection and acceptance, and clear and consistency system documentation.

For more than 10 years many researchers and practitioners have been studying and proposing approaches to tackle different issues of software traceability, as well as tools to support generation, maintenance, and deployment of traceability during software system development. However, despite its importance and the efforts of researchers and practitioners in proposing new approaches and tools, experience and empirical studies have demonstrated that software traceability is rarely established.

This phenomenon raises the main question “why is software traceability rarely established?”, as well as other questions like “what are the problems with the existing tools that cannot fully support software traceability?”, “can we trust the existing tools?”, “what are the real benefits of establishing software traceability?”, “what are the meanings of the different traceability relationships?”, “how can we make the best use of the traceability relations?”, “how can we verify the correctness and completeness of traceability relations?”. In this report we try to answer some of these questions by giving an overview of the state-of-the-art in software traceability. We also identify some open issues that require further development by researchers and practitioners. The overview and open issues are based on our study of the literature described in more detailed in [3][4], our experience of building systems to support software traceability, and open discussions with practitioners and researchers in international workshops on traceability [1][2].

### Overview of the State-of-the-Art

In recent years study, research, and approaches for software traceability have been related to four main aspects: (a) study and definition of different types of traceability relations, (b) support for traceability generation, (c) support for representation and maintenance of traceability relations, and (d) study of how to use traceability relations to support various software development activities.

### Types of Traceability Relations

Various classifications have been proposed for different types of traceability relations. In addition, many reference models and frameworks have also been suggested to support semantic interpretation of these relations. The various classifications for traceability relations are based on the types of artefacts being related and how the relations are used to assist with different software system development activities.

These classifications can range from *vertical* and *horizontal* relations, if the relations associate elements in the same or different artefacts, respectively; to the notion of *pre-* and *post-traceability* relations, if the relations are between requirement specifications and their sources, or requirements specifications and artefacts created in other stages of the development life-cycle, respectively; and to more specific types of relations denoting dependency, generalisation/refinement, evolution, satisfaction, overlap, conflict, rationalisation, and contribution among the artefacts.

However, despite the various classifications that have been proposed there is a lack of standard semantic definition for the different types of relations. Many tools and approaches for software traceability allow the representation of the different types of relations, but the interpretation of the semantics of the relations depends on the stakeholders. This lack of standard semantics causes confusion in the interpretation of the relations, does not provide confidence in the use of traceability techniques and support to assess accuracy of the relations, and makes the development of tools to allow (semi-) automatic generation of relations a difficult task.

### Generation of Traceability Relations

Existing support for generation of traceability relations can be classified as *manual*, when the elements to be traced and the relations among them are manually established by the users; *semi-automatic*, when new traceability relations are generated based on a set of manually user-defined relations or when traceability relations are generated as a by-product of the software development process (process-driven approaches); and *fully-automatic*, when traceability relations are automatically generated by the use of different techniques such as information retrieval, rules, and inference axioms.

Most of the existing requirements engineering and traceability tools offer only manual generation of traceability relations, based on the use of sophisticated visualisation, display, and navigability components, in which the users can visualise the artefacts and identify the elements to be related. However, manual generation of traceability relations is difficult, error-prone, time consuming, and requires significant effort from the user – in particular when dealing with large and complex artefacts.

In order to alleviate the issues associated with manual generation of traceability relations, approaches to support semi- and fully-automatic generation of traceability relations have been proposed. Although they provide an improvement when compared with the manual generation of traceability relations, these approaches have been implemented as prototype tools and, therefore, have not yet achieved a level of maturity required for large-scale industrial use. Moreover, the performance levels of these tools in generating traceability relations may be inadequate and the quality result rates (e.g. recall and precision rates) of the generated relations may not be trustworthy.

## Representation and Maintenance of Traceability Relations

Approaches to support representation and maintenance of traceability relations range from the use of centralised databases and software repositories, to open hypermedia architecture and mark-up based documents.

The use of centralised databases to represent and maintain traceability relations has been supported by the majority of industrial requirements management tools. They provide very good querying facilities for traceability relation retrieval. However, the different types of artefacts and traceability relations that can be supported may depend on the database being used. In addition, it is difficult to support maintenance of evolving artefacts that are imported in the database after being created by different tools.

Software repositories have been proposed to allow storage of a wide range of artefacts and traceability relations created by different tools. The software repository provides the use of application programming interfaces (APIs) for creating and accessing artefacts and their relations, but requires extensive effort for external tool integration. The use of open hypermedia architecture has been proposed to alleviate the problem of maintaining traceability relations in evolving artefacts without having to integrate external tools around a software repository. Both software repository and open hypermedia approaches have only been advocated by research prototypes.

In order to support interoperability of heterogeneous tools, some approaches have proposed the use of mark-up languages to represent traceability relations and artefacts. These approaches support extensive traceability relation types, but have limited querying facilities (they are based on mark-up queries techniques) and maintenance of traceability relations in modified artefacts is not easy.

## Support for Various Development Activities

Software traceability can be used in different stages of the software development life-cycle to support various activities. In change impact analysis, software traceability can be used to establish which artefacts will be affected when one directly or indirectly related

artefact is changed and to establish the cost of introducing the changes. Affected related artefacts can be discovered by querying traceability relations, computing transitive closures of traceability relations, or composing paths of traceability relations (*trace paths*) using regular expressions, scripts, or rules. Some approaches have also been proposed to allow the analysis of wide-effect changes by using simulation techniques and special traceability relations. However, very few approaches have been proposed to support change cost analysis.

Software traceability can also support system validation and verification in which traceability relations are used to demonstrate that the system fulfils given requirements, and to check for system conformance, correctness, and consistency. It is also possible to use traceability relations to support component reuse by identifying similarities between an existing system and a new system being developed, following the traceability relations to locate artefacts that realise the existing system, and analysing if these artefacts can be reused in the new system. This can be very useful when developing product family systems. Other uses of traceability relations are concerned with system testing, inspection, acceptance, and understanding.

## Open Issues

Despite the various approaches, tools, and techniques that have been proposed to support software traceability, it has been demonstrated that they do not provide sufficient support for full traceability establishment. There are still many open issues that prevent the use of traceability in industrial settings and that need to be addressed in the near future.

One of the main issues is the lack of adequate automatic support for traceability relation generation, in particular for artefacts that incorporate text statements and code specifications. Other important issues are concerned with the lack of standard semantics for the different types of traceability relations, mechanisms for guaranteeing and verifying correctness and completeness of traceability relations, mechanisms to support trust of generated traceability relations, and mechanisms to support dealing with the high number of possible traceability relations that may be generated.

There is no consensus in the industry of what should be considered satisfactory recall and precision rates for tools supporting software traceability. Moreover, there is also no clear evidence of the real benefits of deploying software traceability, nor consensus on how to measure these benefits. Furthermore, the adoption of software traceability can only be effective if traceability is adjusted to the needs of specific projects and organisations. However, no methods have been proposed to support the identification of these needs.

There is no question that the establishment of software traceability is very important for the software system development life-cycle. In this report we have

outlined existing approaches that have contributed to various aspects of software traceability. We have also presented some open issues that require further investigation by researchers and practitioners before software traceability can be widely adopted in industrial settings and benefit software system development.

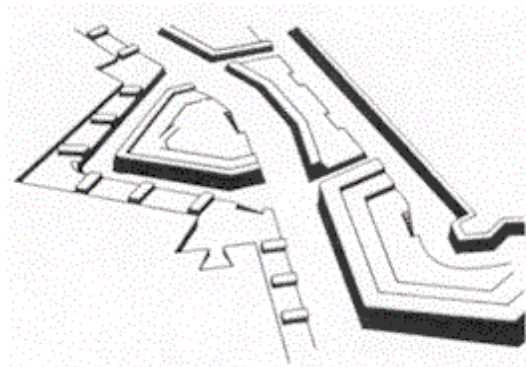
- [1] G. Spanoudakis, A. Zisman, E. Perez-Minana. Proceedings of the 1st International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE' 02), organised in conjunction with the 17th IEEE International Conference on Automated Software Engineering, September 2002.
- [2] G. Spanoudakis and A. Zisman. Proceedings of the 2nd International Workshop on Traceability in Emerging Forms of Software Engineering

(TESFSE' 03), organised in conjunction with the 18th IEEE International Conference on Automated Software Engineering, ISSN 1364-4009, October 2003.

- [3] G.Spanoudakis and A.Zisman. Software Traceability: A Roadmap, Handbook of Software Engineering and Knowledge Engineering, 3rd volume, (eds.) S.K. Chang, World Scientific Publishing Co., 2004 (to appear).
- [4] G.Spanoudakis and A.Zisman. Software Traceability. Tutorial notes in the 12th IEEE International Requirements Engineering Conference, Japan, September 2004; and 19th IEEE International Conference on Automated Software Engineering, Linz, September 2004.

## RE-flections

### Requirements for Fortifications Engineering



*“The art of fortifying does not consist in applying rules or following a procedure, but in good sense and experience.”*

Maréchal Sébastien le Prestre de **Vauban**  
(1633-1707, Military Engineer to King Louis XIV)

Vauban started his army career as an ordinary officer, involved in attacking rather than defending. He learnt what to look for: weak places not covered by artillery or protected by ditches. When he turned from poacher to gamekeeper, he put his knowledge of attackers' thinking to good use to mitigate the threats to a defensive position.

Artillery had rendered high stone walls useless, so his fortifications used thick earth embankments, smooth glacis to expose attackers to interlocking arcs of fire, confusing cross-walls to divide up the attack, and ditches to impede progress. Where available, Vauban incorporated natural features such as rivers and hills, so all his solutions are different. He was a great engineer with an intuitive grasp of the essential requirements.

### Necessity: Missed Key Requirements

Having kept a list of Requirements Proverbs for some years, it was with some surprise that I realised that

*‘Necessity is the mother of invention’*

is of course about Missed Key Requirements. How long does it take to design an air-to-air refuelling system for a new type of aircraft, for instance? Well, the answer during the Falklands/Malvinas war was ‘three weeks’. Hercules transport planes suddenly acquired an enormously extended range, meeting a critical military need in the process.

A Missed Key Requirement means a situation where you are prepared to sacrifice anything else to get the job done, even if it means you have to break the rules and ignore secondary considerations like maintainability or even safety – obviously this depends on your environment. If you really do absolutely have to have something, you feel the pain, and everything else falls by the wayside. There are never very many Key Requirements; Missed ones should be even rarer.

*‘Where there’s a will, there’s a way’*

is clearly talking about much the same thing, but perhaps I can give a more peaceful example of its application. At RE'04 in Kyoto, several of us were lucky enough to visit some charming small family restaurants. These were invariably run from tiny houses with no more than three or four tables, and a kitchen scarcely big enough for one (usually the husband). The wife would serve the food and clear away the plates: as there was generally no room to wash up while cooking was in progress, the plates would be discreetly stacked on one side. Somehow, the entrepreneur-couples managed to make space for each other to work quickly and efficiently in these tiny spaces, where we'd have been impossibly cramped. We found it challenging enough to eat sitting cross-legged and using chopsticks.

**RE-Creations**

To contribute to RQ please send contributions to Ian Alexander (ian@scenarioplus.org.uk).

Submissions must be in electronic form, preferably as plain ASCII text or rtf.

Deadline for next issue: 30<sup>th</sup> November 2004

**RESG Accounts**

SPECIALIST GROUP ACCOUNTS				
YEAR ENDED 30 APRIL 2004				
<u>Requirements Engineering</u>				
	B/FWD	MOVEMENT	2003/2004	C/FWD
	01/05/2003	DR	CR	30/04/2004
BALANCE B/FWD	24,131.17			22,272.39
SURPLUS / (DEFICIT)	(1,858.78)			(427.91)
<b>TOTAL</b>	<b>22,272.39</b>			<b>21,844.48</b>
FIXED ASSETS	0.00			0.00
CASH IN HAND	0.00			0.00
CURRENT ACCOUNT	2,161.72	5,023.78	892.64	6,292.86
DEPOSIT ACCOUNT	0.00			0.00
OTHER BANK ACCOUNTS	0.00			0.00
GOLD ACCOUNT	20,228.14	1,590.06	6,178.20	15,640.00
DEBTORS	0.00			0.00
CREDITORS	0.00			0.00
BCS	0.00			0.00
VAT	(117.47)	215.07	185.98	(88.38)
<b>TOTAL</b>	<b>22,272.39</b>			<b>21,844.48</b>
				0.00
<b>INCOME &amp; EXPENDITURE ACCOUNT</b>				
<b>INCOME</b>				
SGEC FUNDING				0.00
MSB DESIGNATED FUND				0.00
SUBSCRIPTIONS			2,990.00	2,990.00
EVENTS / CONFERENCE INCOME			2,622.77	2,622.77
INTEREST - GOLD ACCOUNT			690.06	690.06
INTEREST - LLOYDS ACCOUNTS			45.03	45.03
MEETING INCOME			80.00	80.00
SPONSORSHIP				0.00
PUBLICATIONS				0.00
SUNDRY INCOME				0.00
<b>TOTAL INCOME</b>	<b>0.00</b>	<b>0.00</b>	<b>6,427.86</b>	<b>6,427.86</b>
<b>EXPENDITURE</b>				
HQ SERVICES				0.00
MEETING EXPENDITURE		322.88		322.88
PRINTING & STATIONERY		689.45		689.45
POSTAGE / TELEPHONE				0.00
COMMITTEE EXPENDITURE				0.00
DEPRECIATION				0.00
BANK CHARGES				0.00
EVENTS / CONFERENCE		5,836.06		5,836.06
SPECIAL FUNDING				0.00
SUNDRY EXPENDITURE		7.38		7.38
<b>TOTAL EXPENDITURE</b>	<b>0.00</b>	<b>6,855.77</b>	<b>0.00</b>	<b>6,855.77</b>
<b>SURPLUS / (DEFICIT) FOR YEAR</b>				<b>(427.91)</b>



## RE-Publications

### Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle

*Ian Alexander and Neil Maiden (Editors)*  
 Wiley, 2004 ISBN 0-470-86194-0

This is a timely book that will serve as an excellent reference point for practitioners and researchers trying to resolve the relationship between use cases and scenarios. Alexander and Maiden have contributed significantly to the burgeoning popularity of scenarios in requirements engineering and, at first, I was slightly surprised that they'd restricted themselves to the role of editors (although they do contribute many of the chapters). In choosing this format, however, they have produced a book with a much wider scope than would otherwise have been possible. Despite the variety in tone and descriptive clarity inevitable in multi-author volumes, this is a book that is jam-packed with goodies. There's something tangible and useful here for everyone with an interest in the direction that modern systems development is heading.

Jacobsen first introduced the Use Case little over 10 years ago. Yet in that time it has both established itself in the main-stream, and been adopted by the radical wings of the development community. Few innovations in system development can have achieved this so quickly so it's perhaps unsurprising that a certain amount of confusion still exists about what Use Cases are and how they're used. Use Cases don't just establish which actors in a system's environment it interacts with, of course. To be useful, they also need to describe the nature of the interaction. To do that, they use scenarios.

Scenario has become a seriously overloaded term. For many (most?) people exposed to UML, scenarios have become closely identified with Use Cases. A side-effect of this is that scenarios have come to mean sequences of events (or even more concretely, sequences of method invocations). Look in most modern dictionaries and something along these lines will be one of the definitions of scenario. However, they will also include the original meaning - an outline for a play. Quite possibly, they will also include a looser form of the sequence of event definition in which scenario can mean some imagined set of circumstances.

This book reclaims all these types of scenario from Use Cases. Of course, many of the chapters are concerned with elaborating the relationship between scenarios and Use Cases and of that between scenarios and requirements. But others describe other applications of scenario, in all their forms, in systems engineering. These range from envisioning changes to systems' business environment to using storyboards for exploring how systems work to the validation of simulations for the Athens Olympics.

This diversity is mostly a good thing. The book represents a timely stock-take of both the state-of-the-practice and the state-of-the-art, with excellent chapters that clearly explain (for example) mechanisms for generating requirements from scenarios, using workshops to generate scenarios, managing scenarios through the life-cycle and how scenarios fit into different project and process types. There are honest appraisals from the editors on what they are good and not so good for and, perhaps as a side-effect, a balanced and informed assessment of the current agile vs. 'traditional' process model debate.

The only down-side of the book's diversity is that the reader (or this reader, at any rate) is sometimes left confused by all the different uses and life-cycle contexts. In particular, it becomes clear that the requirements engineering and human-computer interaction communities have evolved related but subtly different concepts of how and where to use scenarios.

At the core of the book are two sections. The first comprises chapters describing tools, methods and approaches in a variety of domains and process types. Despite the range of approaches described and some convincing arguments for their use in (for example) testing, it is clear that the utility of scenarios is heavily weighted to the early stages of the development process; the ones that in, conventional models at least, result in a requirements specification document.

The second core section is all about case studies to establish the different approaches' credentials and help the reader learn from real practical experience. As with the earlier section, a skew is revealed in that most of the examples are drawn from large systems engineering projects. These are all fascinating but the over-representation of large heavy-duty projects is slightly unfortunate because the earlier section included interesting and entrenched chapters on scenarios in RUP and agile development. It would have been good to learn more about how scenarios help realise the type of project that these approaches are optimised for.

However, these are minor defects that only slightly detract from the book's value. That it works so well overall is a credit to the editors who, in addition to their own specialist contributions, have taken great trouble to pull the different strands together and help the reader understand and classify the different contributions. When the inclusion of a set of really useful resources (scenario templates, for example) is taken into account, the book represents a substantial contribution to the maturation of scenarios in system development.

© Pete Sawyer 2004

## Book Review: Sandom & Harvey: Human Factors for Engineers

*Carl Sandom and Roger S Harvey (Editors)*  
*IEE (www.iee.org), 2004*

Human Factors, Ergonomics, HCI, MMI... there is a constellation of terms (the first two essentially synonymous, but from opposite sides of the Atlantic) for some of the ways researchers and increasingly engineers look at the human-facing aspects of systems of all kinds. These have inevitably tended to attract practitioners who prefer to deal with people rather than spokes and wheels, so HF has to labour under the burden of the caricature of the plump academic young lady who breezes into the workshop and starts asking questions about lighting and posture. Sandom mentions some other myths: "that it is about common sense, that it is expensive to include in project plans, and that it is time consuming. None of these could be further from reality..." (page xxvi). If anyone needs an argument in favour of human factors, Stuart Arnold's eloquent Preface should supply it.

To its credit, this book is sensible, practical, and aimed squarely at engineers, as the title suggests. The editors have contributed the introductory (Harvey) and closing (Sandom) chapters, and have invited a capable team of authors to contribute the chapters in the body of the book.

Multi-author books are often difficult to review, and unless carefully edited, they can also be hard to read and impossible to obtain an integrated view from. The editors have however been scrupulous in assembling this book, and the chapters have been carefully edited – the coverage is uniform, and there are cross-references between chapters as necessary to tie the components into a coherent structure.

The body of the book begins with a chapter by Michael Tainsh on HF integration, within a defence context – that is where it began, and it is also the author's background, but the lessons are, as he argues, largely generic.

Jan Noyes, Kate Garland, and Daniel Bruneau contribute a chapter on Humans: skills, capabilities, and limitations. This covers the usual ground of sensations, perception, attention, memory and errors, with implications for design. Human information processing is briefly illustrated with examples of visual perception illusions.

Jan Noyes also authors a chapter on the HF toolkit, by which is meant a set of conceptual tools necessary to the HF venture. These include subjective methods such as checklists and focus groups; "objective" methods such as observation, task analysis, human reliability assessment; and empirical methods like experiment, modelling, and fitting trials. Which to use? That key issue is discussed in a couple of pages.

Leslie Ainsworth covers Task Analysis, with a brief opening section on data collection for that purpose: of course, that task overlaps with requirements gathering; HF is only an aspect of system behaviour.

Erik Hollnagel discusses Automation and Human Work, considering the principles involved. This is a relatively chatty chapter, and it is somewhat over-illustrated with clip-art in places.

Sidney Dekker writes on the theme 'To Engineer is to Err', i.e. where errors come from and whether these are the human's fault.

David Embrey looks at Qualitative and quantitative evaluation of human error in risk assessment, covering human reliability analysis (again), criticality analysis, failure models, quantification, and the choice of techniques.

John Wood writes on Control Room Design, covering classical ergonomic issues like room layout, seating, workstation design, and environmental design -- everything from the principles for wall finishes and lighting to the provision of plants and the relevance of corporate image. It's all mentioned quietly and briefly in a non-controversial tone, just like a well-designed control room.

Robert Macredie and Jane Coughlan discuss Human-Computer Interfaces. They consider different types of interface, and give a detailed example (a railway ticket machine).

Martin Maguire writes about Usability, covering its definition, benefits, role, scope, and types -- both formative (enhancing usability within design) and summative (in terms of user experience), as well as user audit of implemented systems. The chapter ends with several examples of usability projects.

Iain MacLeod describes HF verification and validation. Surprisingly, this chapter quotes Standish Group on the reasons for system failure, in the manner of the introduction to a requirements textbook. It then discusses the MoD CADMID life-cycle, relating HF V&V activities to these. The two V's are almost exclusively used together, only being distinguished in the introduction, and never subsequently referenced separately except briefly in the context of task analysis. One could argue that checking Fitness for Purpose (FFP) was a classical case of Validating that the requirements were right, but the author chooses to use the term Verification for this meaning (nothing wrong with that), while the definition of HF Validation overlaps with that by also mentioning FfP.

Ed Marshall contributes a fascinating chapter about the application of simulators in systems evaluation, again inevitably with a strong military history. Simulation has gradually spread to the nuclear and civil aviation industries: like HF in general, it has been adopted only when its lack has been painfully felt.

Carl Sandom's chapter is not a summary or rounding-up of loose ends, but a straight discussion of Safety

assessment and HF. The overlap of course arises because hazard events in a fault tree may be caused by human interactions, whether these are misunderstandings of displayed information or wrong commands (slips).

Safety is something of a theme of the book, being covered from various angles in different chapters. The V&V chapter, for instance, splendidly quotes D. Norman on the "Three Mile Island nuclear power incident":

"They wanted to see what was the matter with their operators. Could we perhaps train them better? I discovered, much to my surprise, that if they had set out to design a plant to cause errors, they could not have done a better job."

That might have introduced or summarised the book.

This book should prove helpful to practising HF engineers, especially those new to the field; it will also be useful to students, and may be of interest to systems engineers in other fields.

© Ian Alexander 2004

## **Comparative Book Reviews: Mathematics Handbooks**

### **Mathematics Handbook for Science and Engineering (5<sup>th</sup> Edition)**

*Lennart Råde and Bertil Westergren  
Springer 2004*

### **Mathematics from the birth of numbers**

*Jan Gullberg  
Norton, New York, 1997*

Even requirements engineers occasionally need to resort to a little maths to get their system specifications in order, so it seemed interesting to look at a couple of popular handbooks of mathematics to see what they had to offer.

It turns out that there is a wide range of styles of mathematical presentation, and the two books here represent opposite ends of the spectrum. Råde's book is a tight, compact volume; almost every page is packed with equations or tables of coefficients. There are some explanatory graphs, but essentially you get a heading like "Newton's Interpolation formula" followed immediately by the relevant equations. If you know what to do with them, fine; if not, you need to look somewhere else. The book is thus not so much a handbook (in the sense of "Witherby's Handbook of

British Birds") but a memory-jogger for "scientists and engineers" who remember they need Newton but can't recall the precise formula to use.

The book does sometimes give illustrations that can be understood fairly readily, like those on fault trees, but even here the nature of the logic gates is implied rather than stated, and being told that you need an "especially designed computer program" is not really especially helpful.

In complete contrast, Jan Gullberg has written a long, elegant, and seductive tour of mathematics, with sections on almost every imaginable branch of the discipline. Each section explains what the topic is, describes its scope and history, and illustrates the background by giving the names, dates, and achievements of the great men of the past. Then the basics of the topic – vector algebra, whatever – are introduced in an essay which combines english, diagrams, equations and even cartoons. Rather than just stating results, each technique is illustrated with a worked example, commented as it goes along.

Gullberg covers topics which everyone finds fascinating, like fractals and the golden ratio. He is always clear, accurate, and entertaining, and there is certainly a tremendous amount of information gently conveyed. Even mathematical humour surfaces from the depths.

The intended audiences for these two books are clearly very different. Each suits its purpose; whether either would suit your and my purposes is a moot point. Råde's dry reminders are insufficient for someone who knows they need some statistic but is unsure which one is appropriate: the figures and equations are there, but the necessary guidance is not. Conversely, Gullberg's witty introduction is splendid for getting an insight into what the maths might do, but stops short, even after 1000 pages, of supplying enough guidance to practical techniques like the Chi-squared test for statistical significance.

If you did a good bit of maths and just need a quick reminder what equation to use, Råde may suit you perfectly.

If you did hardly any maths and want to feel a bit more comfortable with it, Gullberg will be your ideal companion.

If you are in between and want some expert guidance on statistics, etc, keep on looking.

© Ian Alexander 2004

---

## **RE-Sponses**

RQ welcomes comments and reactions to articles and reports published in its pages.

---

## ***RE*-Sources**

---

### **Books, Papers**

*For a full listing of books, mailing lists, web pages and tools that have appeared in this section in previous newsletters, see the RQ archive at the RESG website:*  
<http://www.resg.org.uk>

*Al Davis' bibliography of requirements papers:*  
<http://www.uccs.edu/~adavis/reqbib.htm>

*Ian Alexander' s archive of requirements book reviews:*  
<http://easyweb.easynet.co.uk/~iany/reviews/reviews.htm>

*Scenario Plus – free tools and templates:*  
<http://www.scenarioplus.org.uk>

*CREWS web site:*  
<http://sunsite.informatik.rwth-aachen.de/CREWS/>

*Requirements Engineering, Student Newsletter:*  
[http://www.cc.gatech.edu/computing/SW\\_Eng/resnews.html](http://www.cc.gatech.edu/computing/SW_Eng/resnews.html)

*IFIP Working Group 2.9 (Software Requirements Engineering):*  
[http://www.cis.gsu.edu/~wrobinso/ifip2\\_9/](http://www.cis.gsu.edu/~wrobinso/ifip2_9/)

*Requirements Engineering Journal (REJ):*  
<http://rej.co.umist.ac.uk/>

*RE resource centre at UTS (Australia):*  
<http://research.it.uts.edu.au/re/>

*Volere:*  
<http://www.volere.co.uk>

*DACS Gold Practices "Manage Requirements":*  
<http://www.goldpractices.com/practices/mr/index.php>

### **Mailing lists**

*RE-online (formerly SRE):*  
<http://www-staff.it.uts.edu.au/~didar/RE-online.html>

The RE-online mailing list aims to act as a forum for exchange of ideas among requirements engineering researchers and practitioners. To subscribe to RE-online mailing list, send e-mail to [majordomo@it.uts.edu.au](mailto:majordomo@it.uts.edu.au) with the following as the first and only line in the body of the message:

subscribe RE-online <your email address>

*LINKAlert:*  
<http://link.springer.de/alert>

A free mailing service for the table of contents of the *International Journal on Software Tools for Technology Transfer*.

<b>Requirements Engineering Journal</b>
---

For 2004, <i>Springer-Verlag</i> are continuing to offer RESG members a substantial discount on subscriptions to the REJ. Members can subscribe for only £38 (print + online) or £27 (online only). See <a href="http://www.springeronline.com">www.springeronline.com</a> .
--

## ***RE*-Actors**

---

### **The committee of the RESG**

**Patron:**

Prof. Michael Jackson, Independent Consultant,  
[jackson@acm.org](mailto:jackson@acm.org).

**Chair:**

Dr Peter Sawyer, Lancaster University, Computing  
Department,  
[sawyer@comp.lancs.ac.uk](mailto:sawyer@comp.lancs.ac.uk).

**Vice-Chair:**

Dr Kathy Maitland, University of Central England,  
[Kathleen.Maitland@uce.ac.uk](mailto:Kathleen.Maitland@uce.ac.uk).

**Treasurer:**

Prof. Neil Maiden, Centre for HCI Design, City  
University,  
[N.A.M.Maiden@city.ac.uk](mailto:N.A.M.Maiden@city.ac.uk).

**Secretary:**

David Bush, National Air Traffic Services,  
[David.Bush@nats.co.uk](mailto:David.Bush@nats.co.uk).

**Membership secretary:**

Dr Juan Ramil, Computing Department, The Open  
University,  
[J.F.Ramil@open.ac.uk](mailto:J.F.Ramil@open.ac.uk).

**Newsletter editor:**

Ian Alexander, Scenario Plus Ltd.,  
[ian@scenarioplus.org.uk](mailto:ian@scenarioplus.org.uk)

**Newsletter reporter:**

*– role currently unfilled –*

**Publicity officer:**

William Heaven, Department of Computing, Imperial  
College,  
[su2@doc.ic.ac.uk](mailto:su2@doc.ic.ac.uk)

**Regional officer:**

Steve Armstrong, Computing Department, The Open  
University,  
[S.Armstrong@open.ac.uk](mailto:S.Armstrong@open.ac.uk).

**Student Liaison Officer:**

Carina Alves, University College London, Department  
of Computer Science,  
[c.alves@cs.ucl.ac.uk](mailto:c.alves@cs.ucl.ac.uk)

**Immediate Past Chair:**

Bashar Nuseibeh (The Open University)  
[B.A.Nuseibeh@open.ac.uk](mailto:B.A.Nuseibeh@open.ac.uk)

**Industrial liaison:**

Dr. Wolfgang Emmerich, University College London,  
[W.Emmerich@cs.ucl.ac.uk](mailto:W.Emmerich@cs.ucl.ac.uk).

Suzanne Robertson, Atlantic Systems Guild Ltd.,  
[suzanne@systemsguild.com](mailto:suzanne@systemsguild.com)

Gordon Woods, Independent Consultant,  
[Gordon@cigitech.demon.co.uk](mailto:Gordon@cigitech.demon.co.uk)

Alistair Mavin, Praxis Critical Systems,  
[alistair.mavin@praxis-cs.co.uk](mailto:alistair.mavin@praxis-cs.co.uk)