Requirenautics Quarterly

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

© 1997, BCS RESG

Issue 9 (January 1997)

RE-Locations

RE-Soundings

Editorial

If this issue arrives a little later than expected, it's because the editorial team had too much fun at RE'97, in Annapolis. We'll be featuring a full review of the conference in the next issue, but for now the short review: "Wow!" You can find the list of technical papers later in this issue. Along with all the usual departments. Enjoy!

Next issue will be in April. The copy deadline is 28th March 1997. Send me some contributions!

Steve Easterbrook,
NASA IV&V Facility, Fairmont WV

Chairman's Message

I hope your holiday met your requirements. What? They were never specified? Shame on you! How about rectifying this by setting some realistic goals for the New Year ("New Year Resolutions" they used to be called in the good old days).

Leading by example the RESG committee has specified an agenda of events for a large part of 1997 included in this issue of RQ. OK, OK, so it's incomplete and inconsistent with previous announcements. That's alright, requirements engineering is an incremental process that tolerates inconsistencies and handles change. It is up to you, the customers, to provide us, the developers, with feedback about this prototype. Otherwise, you know the consequences: meetings you don't attend...

Next, some stakeholder analysis. The New Year will see an RESG committee re-shuffle. Stewart Higgins has resigned as the group's Secretary to take up a post with Philips in Holland (thank you Stewart for a job well done and good luck for the future). Michael Bearne from Philips Research Labs (Redhill) has agreed to take on the role of RESG Secretary, and I am delighted to welcome him onto the committee. Olly Gotel will also be leaving City University to take up a post at the Defense Research Agency (DRA). The consequence of this is that she will now take on the role of Industrial Liaison Officer for the group, swapping jobs with Andy Vickers who is now the group's Publicity Officer.

All clear? You know who is doing what? We have spared no expense on using the latest traceability tools to update the contact details of the RESG committee on page 15 of this newsletter. On behalf of this new organisation, I would like to wish all RESG members, old and new, a very Happy New Year!

Bashar Nuseibeh,
Imperial College, London

RE-Treats

Forthcoming events organised by the group

Requirements Engineering - Clapping with one hand?

Wed, 19th February 1997

The second in our occasional series of talks on "topics that should be banished to room 101".

Speaker: Dr. Vic Stenning, Visiting Professor, Department of Computing, Imperial College

Venue: Rm 418, Department of Computing, Imperial College, London.

Time: 2:00pm to 4:00pm

Cost: free to members, £5.00 to others

Abstract: Requirements engineering is often regarded as a self-contained activity that can and should be separated from design. Indeed, one of the traditional principles in the requirements arena is that so-called "premature design" must studiously be avoided.

We argue that this stance is misguided, and that requirements and design are in fact mutually inter-dependent - solutions govern needs at least as much as needs govern solutions. Rather than attempting to compartmentalise and separate the activities of requirements definition and solution design, we should recognise that both form part of one holistic activity that takes the form of a continuous learning exercise for everybody involved.

About the Speaker: Vic Stenning spent eleven years with Systems Designers Ltd and seven as Technical Director of Imperial Software Technology. Since 1988 he has been with Anshar Ltd, providing consultancy on software engineering and related topics. Increasingly his focus is on the use of IT to enable business change.

Cost: free to members, £5.00 to others

Business Specification Patterns

Wednesday 26th March, 1997

A half-day tutorial

Speaker: Haim Kilov (IBM TJ Watson Research Center, New York, USA)

Location: Room 418, Huxley Building, Imperial College, London

Time: 2:00pm to 5:00pm

Cost: £30 to members of the RESG & students, £50 to BCS members, £100 to others

See the RE-Views section later in this newsletter for details of this workshop.

System Requirements Engineering

Sunday April 6 to Wednesday April 9, 1997

An IEE Residential Course, co-sponsored by the BCS Requirements Engineering Specialist Group

Location: Moller Centre, Cambridge

For more information: contact Sarah Evans (SEvans@iee.org.uk)

Requirements for Off-the-Shelf Systems and Software

Wednesday, June 11, 1997

Location: Room 418, Huxley Building, Imperial College, London

Time: 2:00pm to 5:00pm

Cost: free to members, £5 to others

Requirements Acquisition and Modelling: The Impact of Reuse

Wednesday September 10, 1997

An evening panel session, organsised in conjunction with the BCS Reuse Specialist Group.

Location: UMIST, Manchester (TBD)

Time: 2:00pm to 5:00pm

Cost: free to members, £5 to others

Quality Certification for the RE Process

Wednesday November 26, 1997

Location: Room 418, Huxley Building, Imperial College, London

Time: 2:00pm to 5:00pm

Cost: free to members, £5 to others

Industrial Experiences in RE

Wednesday February 4, 1998

Location: York University

Time: 2:00pm to 5:00pm

Cost: free to members, £5 to others

Please note: *RESG's policy is that distribution of speakers' slides will be free of charge to those who attend the meeting and specifically request a copy. Those people who do not attend the meeting and request a copy of the slides will be asked to pay £5 to cover photocopying and mailing costs.*

RE-News

Requirements Engineering Journal

The first issue of the Requirements Engineering Journal is now available both in hard copy and on the World-Wide-Web (first issue only, abstracts only thereafter) at http://www.mac.co.umist.ac.uk/RE/Journal.html

Viewpoints'96

Viewpoints 96: An International Workshop on Multiple Perspectives in Software Development was held at the ACM Symposium on Foundations of Software Engineering in San Francisco on October 14th & 15th 1996. You can find the proceedings of this workshop in:
http://www.cs.city.ac.uk/homes/gespan/vptoc.html

Safety Critical Systems Papers

The listing of reports and papers from the DTI/EPSRC-funded Safety Critical Systems R&D Programme is now available as a Word 6.0 file by ftp from ftp.comp.lancs.ac.uk (courtesy of Ian Sommerville). The file is called listing.doc and can be found in directory pub/scs/.

For information about technology transfer activities out of the Programme see:
http://web.ukonline.co.uk/Members/mike.falla/scsp.html

Calendar

February 1997

1997 IEEE Aerospace Conference, Snowmass, Colorado, February 1-8, 1997.
http://chirp.plk.af.mil:1050/ieee/index.html

International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97), Rio de Janeiro (Brasil), February 4-6. *http://www-laforia.ibp.fr/Context*

Third International Workshop on Object-oriented Real-time Dependable Systems (WORDS 97), Newport Beach, California, February 6-7, 1997.
http://www.eng.uci.edu/dream/words97.html

First Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD97), Singapore, February 23-24, 1997. *http://www.iscs.nus.sg/conferences/pakdd97.html*

March 1997

IEEE Virtual Reality Annual International Symposium (VRAIS '97), Hyatt Regency, Albuquerque, NM, March 1-5 1997. *http://www.eece.unm.edu/eece/conf/vrais/*

2nd European Research Seminar on Advances in Distributed Systems (ERSADS), Zinal, Valais, Switzerland, March 17-21, 1997. *http://lsewww.epfl.ch/conferences/ersads*

Workshop on Human Error and Systems Development, The Senate Room, University of Glasgow, 20-22 March 1997.
http://www.dcs.gla.ac.uk/~johnson/HF_Engineering.html

Empirical Assessment & Evaluation in Software Engineering (EASE), Keele University, Staffordshire, UK, 24-26 March 1997.
http://www.keele.ac.uk/depts/cs/Announcements/conferences.html

International Conference and Workshop on Engineering Of Computer Based Systems, Monterey, California, March 24-28, 1997.
http://www.ece.arizona.edu/department/edl/ecbs/callforpapers.html

April 1997

The Pacific Asia Conference On Information Systems (PACIS'97), Brisbane, Queensland, Australia, 2-5 April 1997. *http://www.fit.qut.edu.au/PACIS97/*

8th Conference on Software Engineering Environments (SEE'97), Cottbus, Germany, 8-9 April 1997.
http://www-sst.informatik.tu-cottbus.de/~wwwsst/SEE97/

2nd European Conference On Cognitive Science (ECCS 97), Manchester, UK, April 9-11, 1997.
http://www.cs.man.ac.uk/ai/AISB97/call

The Third International Symposium on Autonomous Decentralized Systems (ISADS 97), Berlin, Germany, April 9 - 11, 1997. *http://www.fokus.gmd.de/ws/isads97/*

10th Conference on Software Engineering Education and Training (CSEE&T) Virginia Beach, VA, April 13-16, 1997. *Info: kpierce@d.umn.edu*

Software Engineering Association Workshop on Integrating Industrial and Academic Viewpoints. Imperial College, London, 14 -16th April 1997

May 1997

Reasoning About Function: A Special Track to be held during the Tenth Florida Artificial Intelligence Research Symposium (FLAIRS '97) Daytona Beach, Florida, May 10-14, 1997. *http://huckleberry.sfsu.edu/*

1997 Symposium On Software Reusability (SSR'97), Boston, Massachussetts, USA, 18-20 May 1997, *http://www.owego.com/~ssr97/*

7th International Workshop on Software Configuration Management (SCM7), Boston, 19-20 May 1997. *Info: conradi@idt.unit.no*

International Conference on Software Engineering (ICSE 97), Boston, Massachusetts, USA, May 18-23, 1997. *http://www.ics.uci.edu/icse97/*

The 7th European - Japanese Conference on Information Modelling And Knowledge Bases, Toulouse, France, May 27- 30, 1997.
http://www.ladseb.pd.cnr.it/infor/Ontology/ontology.html

Tenth International Software Quality Week 1997 (QW`97), San Francisco, California USA, 27-30 May 1997. *http://www.soft.com/QualWeek/*

June 1997

Third International Symposium and Forum on Software Engineering Standards (ISESS 97), Walnut Creek, California, June 1-6, 1997. *Info: John.Harauz@hydro.on.ca*

Sixth International Conference On User Modeling, Chia Laguna, Sardinia, Italy, June 2-5 1997.
http://www.cs.uni-sb.de/UM97/

Fifth International Symposium on Assessment of Software Tools and Technologies (SAST'97), Pittsburgh, PA, USA, 3-5 June, 1997. *http://www.sei.cmu.edu/~case/sast97/*

4th Eurographics Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS'97), Granada, Spain, 4-6 June 1997.
http://www.info.fundp.ac.be/~jvd/dsvis

International conference on Computer Ethics, Linkoping University, Sweden, 9-10 June 1997.
Info: ninni@tema.liu.se

ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'97), Seattle, WA, June 15-18, 1997.
http://www.research.att.com/conf

Third International Workshop on Requirements Engineering: Foundation for Software Quality, (REFSQ'97), Barcelona, Catalonia, June 16-17 1997.
http://alabi.upc.es/~caise97/REFSQ97.html

The 9th Conference On Advanced Information Systems Engineering (CAiSE'97), Barcelona, Catalonia, 16-20 June 1997. *http://www-fib.upc.es/caise97*

12th Annual Conference on Computer Assurance (COMPASS'97), Gaithersburg, MD, June 16-20, 1997. *http://hissa.ncsl.nist.gov/compass/*

IEEE Sixth Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE'97), Cambridge, Massachusetts, USA, June 18-20, 1997.
http://www.cerc.wvu.edu/WETICE/WETICE97.html

Ninth International Conference on Software Engineering and Knowledge Engineering (SEKE '97), Madrid, Spain, June 18-20, 1997. *http://www.fi.upm.es/seke97*

European Symposium on Validation and Verification of Knowledge Based Systems (EUROVAV-97), Leuven, Belgium, June 26-28, 1997.
http://www.econ.kuleuven.ac.be/congres/EuroVaV/eurovav97.htm

July 1997

Software Technology and Engineering Practice (STEP'97) 8th International Workshop (incorporating CASE'97), Holiday Inn, King's Cross, London, 14-18 July 1997. *http://www.co.umist.ac.uk/STEP97*

Second IFIP International workshop on Formal Methods for Open Object-based Distributed Systems (FMOODS'97), Canterbury, UK, 21st-23rd July, 1997.
http://alethea.ukc.ac.uk/Dept/Computing/Research/NDS/FMOODS/

August 1997

Fifth International Conference On Conceptual Structures (ICCS'97), Seattle, USA, August 4 - 8, 1997.
http://www.cs.uah.edu/~iccs97/

The Twenty - First Annual International Computer Software and Application Conference (COMPSAC 97), Washington DC, August 13-15, 1997. *Info: paulra@acq.osd.mil*

Designing Interactive Systems: Processes, Practices, Methods, And Techniques, (DIS 97), Amsterdam, The Netherlands, 18-20 Aug 1997. *http://www.acm.org/dis97/*

Fourth ISPE International Conference On Concurrent Engineering: Research And Applications (ISPE/CE97), Troy, Michigan, USA, August 20-22, 1997. *http://www.secs.oakland.edu/SECS_prof_orgs/ISPE/CE97.htm*

Second International Conference on Cognitive Technology, (CT'97) "Humanising the Information Age", University of Aizu, Japan, 25 - 28 August 1997.
http://www.u-aizu.ac.jp/news/news.html.

September 1997

Second International Conference on Coordination Models and Languages (COORDINATION'97), Berlin, Germany, September 1-3, 1997.
http://www.wins.uva.nl/research/coordination/

Third IEEE International Conference on Engineering Of Complex Computer Systems (ICECCS'97) Villa Olmo, Como, Italy, September 8-12, 1997.
http://www.elet.polimi.it/iceccs97

Sixth European Software Engineering Conference (ESEC), together with the Fifth ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE), Zurich, Switzerland, September 22-25, 1997.
http://www.ifi.unizh.ch/congress/esec97.html

Eighth Specification and Description Language (SDL) Forum, Evry, France, 22-26 September 1997. *http://alix.int-evry.fr/lor/SDL97/*

International Conference on Software Maintenance (ICSM'97), Bari, Italy, Sept 29 - Oct 3, 1997.
http://www.cis.ohio-state.edu/~harrold/icsm97

October 1997

10th European Workshop on Knowledge Acquisition, Modeling, and Management (EKAW-97), Sant Feliu de Guixols, Catalonia (Spain), October 15 - 18, 1997.
http://www.acia.org/ekaw

The 1st International Enterprise Distributed Object Computing Workshop (EDOC'97), Marriott Resort, Gold Coast, Australia, October 24-26, 1997.
http://www.dstc.edu.au/events/edoc97/

November 1997

The 8th International Symposium on Software Reliability Engineering (ISSRE'97), Albuquerque, New Mexico USA, November 2 - 5, 1997.
http://www.cs.colostate.edu/~malaiya/issre.html

Fourth International Symposium on Software Metrics (Metrics '97), Albuquerque, New Mexico USA, November 5 - 7, 1997. *http://www.cs.pdx.edu/conferences/metrics97/*

4th International Conference On Object-Oriented Information Systems (OOIS'97), Brisbane, Australia, 10 - 12 November 1997. *http://www.cs.uq.edu.au/conferences/oois97*

Symposium On Current Trends In Software Engineering, Louvain-la-Neuve, France, November 15, 1996. *http://www.info.ucl.ac.be/event/CTSE.html*

December 1997

Joint 1997 Asia Pacific Software Engineering Conference (APSEC) and International Computer Science Conference (ICSC) Hong Kong, 2 - 5 December 1997.
http://www.comp.hkbu.edu.hk/~apsec

March 1998

European Joint Conferences on Theory and Practice of Software, Lisbon, Portugal, March 30 - April 3, 1998. *http://www.di.fc.ul.pt/~llf/etaps98/*

April 1998

Third IEEE International Conference on Requirements Engineering (ICRE'98), Colorado Springs, Colorado, USA, April 5-10, 1998. *http://www.cs.technion.ac.il/~icre98/*

RE-Issues

Business specification patterns

Wednesday 26th March, 1997

In March, we will be hosting another half-day tutorial.

Speaker: Haim Kilov (IBM TJ Watson Research Center, New York, USA)

Location: Room 418, Huxley Building, Imperial College,

The tutorial shows:

- how to understand businesses (including business rules) independently of any computing systems used for their possible automation;
- how to express this understanding in clear, precise, and explicit business specifications to be used both by business domain experts and software developers;
- how to reuse concepts and constructs ("patterns") common to all business specifications, in order to save intellectual effort, time and money;
- how to elicit business requirements during walkthroughs with the customers, and how to separate concerns using viewpoints;
- how to transform business specifications into various business designs, including the role of workflow; and how business specifications support business transformation;
- and how to transform business specifications and business designs into various information system (software) designs and implementations.

The tutorial includes numerous industrial examples and is based on engagement experience. The concepts and constructs presented here are methodology- neutral and have been incorporated into several international standards. References will be provided.

The material of this tutorial has been successfully presented, by invitation, at TOOLS, OOPSLA and Object World conferences, at OMG meetings, at the London School of Economics, and in various industrial settings, including customers. This material has been published in numerous papers and books, and publicly praised by practitioners - analysts and customers.

Outline of the tutorial

Understanding and explicitly specifying requirements is the most important aspect of solving business problems. (The goal is to solve business problems, not to produce code.)

- insurance specifications written in 1835 still work: they describe business rules in a precise, concise, and explicit manner; so why can't we do the same?
- create applications that do what the business wants them to do;
- clear distinction between business and software specifications;
- explicitly recover "buried" business knowledge;
- active participation of business domain experts is essential;
- business designs based on business specifications show how to provide for business transformations and how to organize workflow;
- software design and development based on business specifications and business designs can be outsourced to third parties (reusable implementation libraries);
- degree of abstraction (and precision) appropriate for each intended audience;
- computer systems (in context) determine some properties of business specifications ;

Simplicity and basic constructs

- how to tame complexity: separate concerns;
- business domain experts and managers understand precision and abstraction in business rules;
- antithesis to "too much stuff"; clear distinction between different business viewpoints and between business and technology frames of reference;
- simple and powerful reusable constructs - things, relationships, contracts for operations - represent a natural way to understand and specify businesses and

their rules;
- no artificial restrictions;
- constructs fit together providing a good, solid foundation;
- never start with a blank sheet of paper;
- business specifications and business designs can and should be understandable by business domain experts, analysts, developers, re-users, and their management: they should be abstract and precise, explicit, unambiguous, declarative, and reusable;
- incomplete (precise; detailed) and partially inconsistent (viewpoints) specifications.

Collections of things and generic business patterns
- things in business are never isolated - therefore collective behavior and relationships;
- business operations are not isolated - therefore triggering conditions and workflow;
- the rules defining a relationship are independent of the identity and nature of related things (relationships' invariants and pre- and postconditions are generic);
- details of invariants (e.g. for property determination);
- using relationships together with operation specifications; the fundamental role of invariants;
- generic relationships as abbreviations of common concepts from business rules;
- different viewpoints (application-specific) of the enterprise lead to multiple and dynamic supertypes, and multiple and dynamic compositions.

Requirements elicitation and business patterns
- dialogues between analysts and subject matter experts;
- guidelines for incremental construction of business specifications (the "do"s and "don't"s; when is a model finished);
- patterns of reasoning;
- specifications, names and their contexts;
- business patterns: generic (e.g. composition, containment), business-generic (e.g. information gathering, assessment, decision, satisfaction), more specific (e.g. underwriting);
- pattern-based elicitation and “mass customization”.

Industrial experience
- examples from customer engagements: reading and writing specifications;
- ‘complete’ specifications don’t exist: importance and role of viewpoints;
- reconciling specification fragments produced independently;
- top-level (roadmap) and more detailed business specifications; refinement (e.g. decomposition) of business rules about things and operations;
- horizontal (at the business frame of reference) and vertical (to create software systems) refinement of specifications using the same constructs and concepts;
- precise definition of reusable software components (in the same manner as reusable business specification components);
- international standards (e.g. ISOs RM-ODP, GRM).

About the speaker

Haim Kilov has been involved in all stages of information management system specification, design, and development. His approach to information modeling, widely used in telecommunications, financial, document management, and insurance areas, has contributed clarity and understandability to enterprise and application modeling, leading to business (and system) specifications that are demonstrably better than traditional ones. It has been described in Information modeling: an object- oriented approach (Prentice-Hall, 1994). Haim Kilov is using and extending his approach in customer engagements, and does research and consulting in the areas of business specifications and information modeling. He is a member of and active contributor to several international standardization technical committees, as well as an invited speaker at OMG task force meetings. He co-chaired five OOPSLA workshops on object-oriented behavioral specifications, and co-edited their Proceedings. He also co- edited a book (recently published by Kluwer) based on the first four of these workshops. He has been a speaker and a program committee member at numerous national and international conferences. He has a significant number of publications in journals and conference proceedings. His interests are in the areas of information modeling, business specifications (including business patterns), and formal methods.

RE-Readings

Reviews of recent Requirements Engineering events.

Workshop on Formal Methods & Requirements Engineering: Challenges & Synergies

The December meeting of the RESG was a two day workshop organised jointly with the Formal Aspects of Computing Science (FACS) Specialist Group

Report by Sara Jones

The aim of this workshop was to encourage communication and understanding between academics and practitioners in the fields of requirements engineering and formal methods, and to identify areas where work in each field might contribute to the other. The workshop was jointly organised by the RESG and FACS (Formal Aspects of Computing Science) special interest groups and was held over two days in the week before Christmas at City University in London.

Around 70 people attended the workshop to hear a distinguished panel of invited speakers presenting RE and FM perspectives on:
- Change in Software and System Requirements and Specifications
- Requirements Traceability

- Requirements Elicitation and Validation
- Non-functional Requirements
- Inconsistency in Software/System Requirements and Specifications
- Use of Multiple Notations
- Architecture, and
- Domain Knowledge

There were also group discussion sessions on each day in which delegates were divided into smaller groups to discuss issues arising from the presentations, as well as more specific questions about what FM could deliver for RE practitioners, and vice versa, in the next 1, 5 or 10 years. In the final plenary session, delegates were invited to put any further questions directly to the speakers.

The feeling at the end of the workshop was that the two communities of RE and FM have much in common. We hope we may be able to support further collaborations in future, but in the meantime, readers should watch this space for a more comprehensive discussion of issues raised during the course of the workshop which will appear in the next edition of this newsletter.

RE'97 - Annapolis

Our busy reporters are still finalising a review of the conference, so it'll have to wait for the next newsletter. But in the meantime, here's the list of the technical papers presented at the conference:

Ten Steps Towards Systematic Requirements Reuse. *W. Lam, J. A. McDermid, A. J. Vickers*

Reusing Operational Requirements: A Process-Oriented Approach. *Robert Darimont, Jeanine Souquieres*

Analogical Reuse of Requirements Frameworks. *Philippe Massonet, Axel van Lamsweerde*

Enhancing a Requirements Baseline with Scenarios. *Julio Cesar Leite , Gustavo Rossi, Federico Balaguer, Vanesa Maiorana , Gladys Kaplan, Graciela Hadad, Alejandro Oliveros*

Producing Object-Oriented Dynamic Specifications: An Approach Based on the Concept of 'Use Case'. *Benedicte Dano, Henri Briand, Franck Barbier*

A Technique Combination Approach to Requirements Engineering. *Alistair Sutcliffe*

Analysing Inconsistent Specifications. *Anthony Hunter, Bashar Nuseibeh*

Requirements for Telecommunications Services: An Attack on Complexity. *Pamela Zave, Michael Jackson*

Systematic Trade-off Analysis for Conflicting Imprecise Requirements. *John Yen, W. Amos Tiao*

Naturalistic Inquiry and Requirements Engineering: Reconciling Their Theoretical Foundations. *Colin Potts, Wendy C. Newstetter*

Integrated Safety Analysis of Requirements Specifications. *Francesmary Modugno, Nancy Leveson, Jon D. Reese, Kurt Partridge, Sean D. Sandys*

Formal Methods for V & V of Partial Specifications: An Experience Report. *Steve Easterbrook, John Callahan*

Extended Requirements Traceability: Lessons from an Industrial Case Study. *Orlena Gotel, Anthony Finkelstein*

On the Use of a Formal RE Language: The Generalized Railroad Crossing Problem. *Philippe Du Bois, Eric Dubois, Jean-Marc Zeippen*

Auditdraw: Generating Audits the FAST Way*. *Neeraj K. Gupta, Lalita J. Jagadeesan, Eleftherios E. Koutsofios, David M. Weiss*

The Integrated Specification and Analysis of Functional, Temporal, and Resource Requirements. *Hanene Ben-Abdallah, Insup Lee, Young Si Kim*

Generating Provably Consistent Code from Hierarchical State Machines. *David J. Keenan, Mats P. E. Heimdahl*

Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. *Eric S. K. Yu*

A Decision Making Methodology in Support of the Business Rules Lifecycle. *Daniela Rosca, Sol Greenspan, Mark Feblowitz, Chris Wild*

A Logical Framework for Modeling and Reasoning about the Evolution of Requirements. *Didar Zowghi, Ray Offen*

Short commentaries on each of these papers can be found at http://www.itd.nrl.navy.mil/conf/ISRE97/isre97-papers.html

CORE-Blimey!

A regular column by Geoff Mullery, of Systematic Methods Ltd.

Reductionism ad Absurdum

General Systems Theory says a system is more than the sum of its parts. Western science is based on Reductionism - breaking down an entity into its constituent parts in order to understand the whole. Considering these two approaches suggests that breaking something down into parts may give some greater understanding, but will frequently miss key features.

In other words, reductionism has some benefits, but if you ignore what it misses you are asking for trouble. A particularly damaging form of reductionism is one which not only decomposes, but then goes on to discard some parts as trivia or even as sources of "disease" (e.g. procedural specification is "wrong" and should not be attempted). Many technical people in computing claim to understand these lessons, but I have my doubts.

In the way we develop systems, invalid reductionism is rife. I have for years tried to get my peers to recognise the absurdity of our approach to system development. Most agree, but they almost invariably think the problem resides in others. Everyone seems determined to compartmentalise and to pour scorn on others because they compartmentalise in a different way.

If we wish to understand the system development process we must understand that we are talking about a system for developing systems. This means that, though we may usefully break the structures and processes down, we can not entirely rely on the sum of the isolated descriptions/applications of the parts in order to construct a successful system-building system.

If you think that project management, planning, resourcing and costing are separate activities from specification and development - that those who do them are irritants or sources of unnecessary distortion (the technician's view) or that they are the only sources of reality in a black art environment (the managerial/administrative view) - then you are guilty of invalid reductionism.

If you think that theory is always irrelevant in a real project environment (the traditionalist's view) or that traditional methods are too informal and unstructured to be worth considering (the theorist's view) - then you are guilty of invalid reductionism.

If you think that there is a single driving principle to the process of system development which should dominate all other principles of development - e.g. rule-based specification or fast prototyping or application of mathematical formalisms - then you are guilty of invalid reductionism.

If you think that a single unified notation (if one were possible) is all that should be used in specifying a system or that development tools with a single dominant user interface policy are all that should be used - e.g. Prolog or Z or tools with a GUI based around use of a diagrammatic notation - then you are guilty of invalid reductionism.

If you think that the only influence on a requirement specification should be the "user" view or that the only "user" views to consider are those directly working in the proposed system's environment - or if you think that accommodating "user" views destroys the benefits of applying the available technology to the proposed environment - then you are guilty of invalid reductionism.

If you think, you can find other examples of how the computer industry - and more specifically the requirements end of the industry - get caught up in notions of compartmentalised development. If you think not, re-read that sentence. The notion of a wholly separate requirements end of the computer industry is another bit of invalid reductionism - as is the phase model the way we normally apply it - including the spiral model of development.

When we build systems - particularly large real time systems or complex distributed on-line systems the scope of the problems address is mind-bogglingly large. It seems that reductionism is the only way we can get some order into the process. Avoiding reductionism by letting semi organised chaos reign (e.g. Soft Systems Theory) is just reductionism according to another partitioning and exclusion principle. Reductionism is, it appears, inevitable!

So, have I led you all this way to tell you "damned if you do, damned if you don't"? Well, partly yes. I believe the overall problem of system development is greater than we currently know how to solve in other than a few special cases. There are certain types of system we recurrently build - e.g. payroll, inventory, accounting, point-of-sale - the requirements of these systems do change (e.g. due to government legislation), but relatively slowly, so that we can come close to what I call Electronic Mars Bars when we choose to build new variants.

However there are cases where, even though an Electronic Mars Bar is available, we choose to build systems using a team for which the project is not a minor variant of one they previously built. There are some systems which are inherently difficult and inherently changeable. A military real time system is quite likely to fall into this category. As it is built the environment changes, the contract changes, the time scales change, the available software and hardware technology changes and the development team changes. If you want to kill your favourite methods or support tools try them in that environment.

Don't slash your wrists just yet (or mine, which I expect you would prefer). Admittedly I can not offer you a solution to the problems, but what I believe I can do is offer you a path to controlling the scope of the disaster. The first step is to stop ignoring the limitations of reductionism. You can still use it, but you must recognise that decomposition does not lead to immutable insulation of the parts from each other and that the direct linkages between the parts are not necessarily the only linkages.

At the risk of becoming too specific, let me try to give you an example which relates to theorems in formal methods and class definitions in Object Oriented approaches. What appears to be provided in these environments is a repository of proofs/definitions and a means of calling up a proof/definition when needed.

For example, I may be trying to validate a part of my system and I will be given some support in looking for a theorem applicable to the topic in my problem - or I may be trying to classify an object in my system's environment and I am likely to be able to call up some facility for short-listing possible existing classes to use or from which to derive a new class, and a browse facility to examine the class hierarchies which seem relevant.

So far, so good. I have a problem and I have some assistance in finding information I need in order to progress handling the problem. This is fine if I am reasonably expert at pattern definition and matching and if my problem and the potential solution aid domain is small enough and structured appropriately so that I can categorise my problem and match it to possible available solutions - and if nothing ever goes wrong with the available solution aids. But what if these criteria are not met and what problems remain even if they are met?

In the Object Oriented domain there is the known problem of deciding between class creation, derivation and composition. When I define a new class for my problem, do I inherit from a single class or several? Do I create a class which is composed of parts which are instances of other existing classes. Or do I combine all these as well as creating a new class which inherits from nothing (other than via its component parts)?

In the formal methods domain there is the problem of transforming from a vaguely defined something into a precisely defined, narrowly focused set of concepts. Is it a set or a bag or a list? Is it a matrix or a tree or a lattice? I can often make the decision, but can I make the people whose requirement it expresses understand what I mean when I try to get them to tighten up their definition to the point where I can make the decision? Often it is in reality their decision and if I make it for them there is a danger of leading them astray.

Suppose that all these problems can be overcome and I arrive at what seems the right definition. Now I will have probably called up several existing theorems or class definitions for use in this bit of the specification. I repeat the exercise for other parts of the specification and end up with a specification which uses numerous existing theorems/class definitions and possibly adds some new ones to become available for later projects.

Now I go to another project and go through the same process - using some of the same theorems/classes, some different ones and creating some more new ones. I go to the next project, and so on. Five years and several projects later someone finds a fault in one of the theorems/class definitions. Where has it been used? Not just which projects might I have used it on, but which places in which projects did I use it - and in which places on which projects did someone else use it?

Well configuration management deals with that doesn't it? In most projects, no it does not. Configuration management typically deals with whole files, not with their content . It deals with the update and release of files and their association with other files. It does not deal with all interactions between files and between parts of files. A theorem may be defined in a single file, but typically it calls up other theorems and typically the dependencies between the theorems is not stored or not easily available, so even when I find the first level of theorem which called up the invalid one, I am left with the problem of finding which other theorems depend on the first level dependent.

What's wrong with a text editor and a search engine (another form of data mining?)? While I am on a small project and have control of the precise terminology used, nothing. Put me on a large project, or put me in an environment where the proof of one theorem calls up another, but with a minor variant of the name and suddenly my text editor is anything from very time consuming to irrelevant and my search engine has to get a great deal cleverer than any I have seen so far. Unless of course you don't mind missing the odd case. Do you think you wouldn't mind if you were on the warship which was about to be destroyed because its system just went down due to the odd missed case in that system?

What I am getting at here is that by compartmentalising specification, support "libraries" and configuration management, something has dropped through the cracks. We should really be able to do much better than that. If we compartmentalise we should look at what the reduction process has dropped - then we should look at where we need to and how we can block the cracks.

With available technology we would have difficulty totally inter-relating every aspect of every specification (and implementation) of every release of every system we produce. That is a database problem of gigantic proportions (I am working on it and I might say something about it in a later contribution!). But what we can look at is the means by which we can provide access to less spectacular tools to support some of the crack-blocking.

That is my message for this contribution. If you build a method or a support tool, it is not the things you have done which limit its utility it is the things you have not done and have, by design, prevented others from doing with the information your method/tool produces which are the real limiters.

I attended the Viewpoints 96 Workshop in San Francisco at the end of 1996 and in the proceedings of that you can find a paper by me which illustrates some of the things you need to do to provide open-ended crack-blocking capabilities. You will also find papers by others which contribute detailed techniques relevant to the generalisations of my paper. I state in my paper that the technology is already available and some of the other papers demonstrate the truth of what I say - but of course reductionism applies here too - we are compartmentalised and widely distributed, so the reality of crack-blocking may depend on you, in your compartment becoming a little less isolationist.

Geoff Mullery
gmul@sml.win-uk.net

RE-Bites...

They're still laughing about this at IBM. Apparently the computer giant decided to have some parts manufactured in Japan as a trial project. In the specifications, they set out that the limit of defective parts would be acceptable at three units per 10,000.

When the delivery came in there was an accompanying letter:

'We Japanese have a hard time understanding North American business practices. But the three defective parts per 10,000 have been included and are wrapped separately. Hope this pleases.'

RE-Publications

Book Review:

R.J.Wieringa. Requirements engineering: Frameworks for understanding. Wiley, 1996, xvi, 453 pp., ISBN 0 471 95884 0.

Reviewed by Simon R Smith, S.R.Smith@durham.ac.uk

Roel Wieringa's book is primarily a teaching text, and has evolved through being used as such over the past five years. Its presentation of requirements engineering is based on a series of structuring mechanisms with which to corral the various strands of this expansive topic. The book "advocates an eclectic approach to methods" and, for the student, a flavour of controlled eclecticism is certainly provided. It is also intended that current practitioners be able to use the book to locate their favoured methods in relation to others.

The book is structured in three parts. The first part introduces a set of concepts with which to discuss models and modelling methods. For example, the useful distinction between prescriptive and normative models is stressed, as is the distinction between bespoke product development and development for a market. Some of the more tractable concepts, or dimensions, are then packaged into multi-dimensional matrices (the 'frameworks' of the subtitle) into which, it is suggested, various models or methods can be slotted in order to assess their suitability for particular kinds of requirements engineering tasks. For example, the framework for assessing behavioural specifications is based on the static/dynamic and universe-of-discourse/system-under-development dimensions.

In the second part, several modelling methods are presented, including analysis methods from the business strategy area. The methods have a heritage in information systems development, and are based principally on entity relationship, data flow and JSD structure diagrams. These central chapters are arguably the least successful when considered in relation to the rest of the book, being too detailed to support the theme of eclecticism (at around 200 pages of description and examples), yet too brief to act as a realistic user's guide.

The third part returns to the frameworks introduced in the first part, in order to demonstrate the complementarity between these models and methods, and gives useful additional material on development strategy and process issues. This includes several all-too-short sections in Chapters 14 and 15 on how the problem-solving process, used as a pedagogical device throughout the book, might be implemented in practice.

This three-part presentation is useful from the point of view of the student: it gives a clear example of how a collection of methods can be viewed as being suited to certain complementary aspects of expressing the desired behavioural characteristics of a system to be built. However, the presentation also starts with a broad remit (requirements engineering for information, communication and control systems), and attempts to provide enough structure to this area in order to render it manageable. Inevitably this leads to the use of simplified examples, and the consequent scant treatment of issues such as validation and traceability. The broad remit also leads to a rather complex narrative progression - this is not bedtime reading!

Wieringa is perhaps not unaware of the pitfalls of this presentation, providing comprehensive end-of-chapter sections on related work and alternative ways of presenting the material. These no doubt act as useful resources for the student, but also help to defuse the criticisms of those reading the book independently of a delivered course.

It seems unlikely that one book can articulate the field of requirements engineering, given the current broad definitions of that term. This book takes a particular slant on requirements engineering (focused primarily on behavioural modelling for information systems), and attempts to present a coherent view within this scope. If read with this caveat in mind (treated, for example, as a much more thoughtful account of the requirements tasks than that found in standard texts such as [You89], or in the more superficial of the object-oriented analysis books, such as [CoY90]) then the book amply achieves its aims.

In summary, the book provides a thorough, well-researched introductory course on requirements engineering, and is well worth reading simply from this perspective. As for the 'frameworks' of the subtitle, the treatment given to these is hampered by the need to develop a textbook which remains coherent while covering a broad area. However, Wieringa has perhaps prepared the way with this book for one which, freed from this need, and perhaps with a narrower scope, can give a more thorough presentation of 'frameworks for understanding'.

References.

[You89] E. Yourdon: Modern Structured Analysis. Prentice Hall (1989)

[CoY90] P. Coad & E. Yourdon: Object-Oriented Analysis. Prentice Hall (1990)

Reviewers Needed!

If you would like to review any recent books for RQ, please get in touch. We get a steady trickle of new books in the editorial office, which we send out free to the first person to volunteer to write a review. I plan to announce the arrival of new books to my mailing list of volunteer reviewers: if you want to be added to this list, drop me a line (steve@atlantis.ivv.nasa.gov).

I also welcome unsolicited book and conference reviews, so don't feel you have to wait for a freebie before you put pen to paper, especially if you've gone and bought one already!

RE-Calls

Third IEEE International Conference on Requirements Engineering (ICRE'98)

Colorado Springs, Colorado, USA, April 5-10, 1998

Sponsored by IEEE Computer Society Technical Council on Software Engineering with Fujitsu and MCI

ICRE '98 is the third in a biennial series of conferences aimed at bringing together practitioners and researchers to discuss software requirements-engineering-related problems and results. Further, as a technology transfer conference, it is designed to provide (1) practitioners with an evaluation of promising requirements research and practice, and (2) researchers with an exposure to real-world requirements problems.

Papers

Authors are invited to submit papers addressing theory and/or practice.

An experience paper evaluates the effectiveness of a requirements engineering technology, which may have begun as a research effort, in a real-world, industrial setting on non-trivial requirements engineering problems

A research paper describes an innovative requirement engineering technology aimed at cracking previously unsolved problems in requirements engineering and demonstrates, perhaps by application to an exemplar, the usefulness of the technology in solving these problems.

Authors are encouraged to focus on one of these two aspects as the primary theme of their contributions. In all cases, practical and near-term applicability of the proposed ideas must be emphasized. Each paper is evaluated according to the standards of its category.

Topics include, but are not limited to:

- Requirements problems, techniques & tools
- Defining the system-human interface requirements
- Business process relationship to system requirements
- Supporting requirements elicitation & evolution
- Requirements engineering in legacy system migration
- Advances in formal methods & their practical use
- Requirements-based system testing
- Requirements definition, analysis & validation
- Prototyping, animation & visualization of requirements
- Non-linear & heterogeneous requirements descriptions
- Requirements engineering as a group activity
- Role of software architecture in requirements
- Lessons from domain-specific requirements practices
- Role of requirements standards

Early Feedback

All authors are strongly encouraged to submit extended abstracts of from one to three pages by May 19, 1997. In the interest of rapid response, these must be submitted either as a plain text file in ASCII form by e-mail to icre98@cs.technion.ac.il or by fax to Prof. D.M. Berry at +972-4-822-1128, with a preference to e-mail. The feedback will be in the same form in which the abstract was submitted. The abstract should summarize the problems addressed, the plan of the paper, the conclusions, and the contributions. Promising papers will receive early feedback by June 16 that can be applied to improving the full paper. Full papers received by 1 August without a previous extended abstract or for abstracts for which no feedback was given will nevertheless be given full consideration.

Panels

Proposals that focus on requirements engineering controversies are encouraged, especially those that highlight the gulf between requirements research and practice. Preference will be given to panels that present a diversity of views on the topic chosen. Proposals should include the panel's title, a brief description of issues to be debated, the names of prospective panel members, and a description of their roles.

Submission

Full length papers are limited to 6000 words, typeset with enough room for comments by reviewers. They should include a short (150-word) abstract, a list of descriptive keywords, specification of the paper's category (experience / research), and complete contact information for the lead author. Authors should send six hard copies of their papers and panel proposals, in English, to one of the program chairs:

From Western Hemisphere:
Brian Lawrence
335 Keeler Court
San Jose, CA 95139
U.S.A.

From Eastern Hemisphere:
Prof. Daniel M. Berry
Faculty of Computer Science
Technion
Haifa 32000, ISRAEL

The best papers in the conference will be considered for publication in IEEE Software.

Internet Hotline

Further information on ICRE '98 may be obtained by sending Internet electronic mail to icre98@cs.technion.ac.il, or through the WWW using the URL http://www.cs.technion.ac.il/~icre98/.

Steering Committee

D. Berry	B. Lawrence
C. Chang	N. Mead

A. Davis
M. Dorfman
P. Hsia
C. Shekaran
J. Siddiqi

Conference Chair: Nancy Mead (nrm@sei.cmu.edu)

Program Chairs: Daniel M. Berry (dberry@cs.technion.ac.il) Brian Lawrence (brianl@netcom.com)

Program Committee**:** People from Industry as well as Academia!

Registration Chair**:** Charlene Rauber

Tutorials Chair**:** J. D. Berdon (jdberdon@mozart.uccs.edu)

Local Arrangements Chair: Rick Hubbard (74035.601@compuserve.com)

Financial Chair: Steve Easterbrook (steve@atlantis.ivv.nasa.gov)

Publicity Chair: Yahya Al-Salqan (alsalqan@Eng.Sun.COM)

Panels Coordinator: Greg Zelesnik (gz@cs.cmu.edu)

Exhibits Coordinators: David Hottman (70254.1560@CompuServe.com) and INCOSE

Key Dates

Extended abstracts due:	May 19, 1997
Feedback to authors:	June 16, 1997
Full papers due:	August 1, 1997
Panel proposals due:	August 1, 1997
Author notification:	October 20, 1997
Camera-ready copy due:	December 29, 1997

Third International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'97)

Barcelona, Catalonia, June 16-17 1997 (preceding the CAiSE 1997 conference)

Purpose

The ultimate measurement for software quality is the degree to which user requirements are fulfilled by a system. Early elicitation and correct definition of requirements prevents costly rework during later development stages and provides the foundation for building high quality systems. Therefore, requirements engineering is considered as a more and more crucial part of the system life cycle.

During requirements engineering the user and engineers have to find a way from an initially opaque and diverse system understanding to exact, reconciled and at least partially formalized system specifications. A multitude of methods from software engineering, ethnology, social sciences, and psychology have been adapted to support this process and to achieve a growing quality of the requirements specification as a foundation of higher system quality. Most of these methods are relying on adequate specification languages which are expressive and formal enough so that the represented quality requirements can be verified or validated.

At the REFSQ'94 and REFSQ'95 workshops researchers and practitioners from various disciplines presented approaches that focused on the improvement of the definition and implementation of (quality) requirements. The success of REFSQ'94 and REFSQ'95 encouraged us to provide a follow-up workshop REFSQ'97 as a stage for the discussion of quality-related problems in requirements engineering as they have developed over the last year. In particular, we like to encourage people from the software and information systems engineering field to present their approaches to higher software quality and to discuss how requirements engineering can contribute to it.

Goal

The main goal of the REFSQ'97 is to bring together people working in the fields of requirements engineering and software (information systems) engineering focussing on the

- specification of quality requirements;
- their traceability back to user needs and forward to the design;
- their realization by SE methods;
- the measurement of their achievement, as well as
- consolidating the achievements of the two previous REFSQ workshops.

Themes

REFSQ'97 invites contributions from research and industry within the following four main themes:

1. Embedding RE in the organisational context.

Relevant topics include: change management, procurement, organisational learning, business processes, etc.

2. Managing the quality of RE processes.

Relevant topics include: traceability, process modelling and monitoring, RE project organisation, quality models of RE and the RE process, environments for supporting RE processes, CAME environments, etc.

3. Quality assurance and RE.

Relevant topics include: models for quality assurance, considering quality assurance in RE, software quality and RE, specification of software quality requirements, measuring the quality of requirements, etc.

4. Mapping requirements specifications to software architecture and design.

Relevant topics include: transformation and mapping methods, the interplay between requirements and software quality features, formal representation methods, etc.

Note that the list topics mentioned for each theme is not intended to be exhaustive. High-quality papers on other topics within each of the four themes are welcomed.

Papers should emphasize what is new and significant about the chosen approach and adequately compare it with similar work. Integration of the contributions with mainstream RE methods and products (like SA, OMT, ER, and the like) are especially encouraged.

Participation

The workshop will be an interactive forum. Each presentation will be summarized by two discussants and followed by a panel discussion between the authors and the audience. Attendance will be limited to around 25 people and all participants must contribute accepted full or position papers. The workshop language is English.

The workshop is being organized in conjunction with the CAiSE*97 conference, and all workshop participants are expected to attend the main conference.

Instructions for Authors

Send your full paper (max. 6000 words) or position paper (max. 2000 words) by e-mail or via normal post before March 17th (arrival date) to:

REFSQ'97
Department of Information Science
University of Bergen
N-5020 Bergen
Norway

email: Andreas.Opdahl@ifi.uib.no

Papers will be published in the REFSQ'97 workshop proceedings, and preprints of the papers will be made available for accepted and registered participants at the beginning of the workshop.

Important Dates

Submission deadline:	March 17th 1997
Acceptance notification:	April 17th 1997
Camera ready paper due:	May 17th 1997

Organisation

Eric Dubois
Dep.of Computer Science
FUND Namur
Namur
Belgium
Tel. +32-81-72 41 11
Fax. +32-81-72 49 67
edu@info.fundp.ac.be

Andreas Opdahl
Dep. of Inf. Science
University of Bergen
N-5020 Bergen
Norway
Tel.+47-5558-4115
Fax. +47-5558-4107
Andreas.uib.no

Klaus Pohl
Informatik V
RWTH Aachen
D-52074 Aachen
Germany
Tel.+49-241-80-21512
Fax. +49-241-8888-321
pohl@informatik.rwth-aachen.de

Managing Inconsistency in Software Development

A Special Issue of the IEEE Transactions on Software Engineering.

Guest Editors: Carlo Ghezzi, Politecnico di Milano, Italy, and Bashar Nuseibeh, Imperial College, London, UK

Aims & Scope

A wide range of inconsistencies can arise during software development - for example, during requirements engineering if contradictory requirements are elicited, during design as developers explore alternative design solutions, and during programming if exceptions are not anticipated or handled. Inconsistencies can also arise between different stages of the development process, for example, between specification and implementation. In fact, a significant proportion of software development activity is spent detecting, analysing and handling inconsistency in development processes and products.

This special issue of IEEE Transactions on Software Engineering will address the management of inconsistency in software development. The aim is to cover the middle ground between detection and resolution of inconsistency, with a focus on living with and making use of inconsistencies that arise in software development. Topics covered by the special issue include but are not limited to:

- identifying/classifying inconsistency (e.g., logical contradictions, conflicts, exceptions, deviations/non-normative behaviour, interference)
- analysing inconsistent descriptions (e.g., identifying sources/causes, prioritising, tracking, monitoring)
- strategies for handling inconsistency (e.g., avoidance, tolerance, resolution)
- acting in the presence of inconsistency (e.g., reasoning, circumvention)
- notations/formalisms for representing inconsistency (e.g., paraconsistent and fuzzy logics)
- organising inconsistent descriptions (e.g., viewpoints, subsets, types)
- tool support (e.g., automation, guidance)
- applications of inconsistency management (e.g., requirements specification, process modelling/enactment, CSCW, distributed systems)

Submissions

Papers are solicited that describe basic or applied research on any of the above topics. An electronic copy of submission title, abstract and keywords should be emailed to Bashar Nuseibeh by 8th September 1997. Six hard copies of submission should be sent to Bashar Nuseibeh by 15th September 1997.

To discuss potential submissions please contact either of the two Guest Editors.

Submitted papers must not have been published previously or be concurrently under consideration for publication in any other journals or conferences.

Important Dates

Email submission of abstracts:	8th September 1997
Full paper submissions (6 copies):	15th September 1997
Notification of acceptance:	15th December 1997
Final revised versions of papers:	2nd February 1998
Expected publication date:	Spring 1998

Guest Editors

Carlo Ghezzi
Dipartimento di Elettronica e Informazione
Politecnico di Milano
Piazza Leonardo da Vinci 32
I-20133 Milano, Italy
Email: ghezzi@elet.polimi.it

Bashar Nuseibeh
Department of Computing, Imperial College
180 Queen's Gate, London, SW7 2BZ, UK
Tel: +44-171-594 8286
Fax: +44-171-581 8024
Email: ban@doc.ic.ac.uk

For the latest information please consult:

http://www-dse.doc.ic.ac.uk/~ban/misc/tse.html

RE-Sources

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 on the web: http://research.ivv.nasa.gov/~steve/resg/

Books

Macaulay, L. A. “Requirements Engineering”. Springer, 1996. ISBN 3-540-76006-7

Pohl, K., “Process-Centered Requirements Engineering”. John Wiley and Sons Ltd, 1996. ISBN 0 86380 193 5

Prasad, B., “Concurrent Engineering Fundamentals. Volume II: Integrated Product Development”, Prentice Hall, 1996. ISBN 0-13-396496-0

Veryard, R., “Information Coordination” Prentice Hall, 1994. ISBN: 0-13-099243-7

Journals

Personal Technologies, a new journal from Springer-Verlag, first issue due to be published in March 1997. http://gate.uwe.ac.uk:8000/csm/research/PIM/PeTe.html

RE-Creations

How to contribute to RQ

Please send contributions to Steve Easterbrook (steve@atlantis.ivv.nasa.gov) before the publication deadline. Submissions must be electronic copy, preferably plain ASCII text. A list of the kinds of contributions we welcome can be found in the January 1996 newsletter, or on the web at *http://research.ivv.nasa.gov/~steve/resg/rq5/ReCreations5.html*

	Copy deadline
Issue 10 (April)	28th March 1997
Issue 11 (July)	27th June 1997
Issue 12 (October)	26th September 1997

Web Pages

The BCS RESG home page can be found at:

http://www.OiT.co.uk/resg/

Back issues of Requirenautics Quarterly:

http://research.ivv.nasa.gov/~steve/resg/

SR/Institute's Software Quality HotList:

http://www.soft.com/Institute/HotList

Mailing lists

Requirements Engineering Newsletter

A Requirements Engineering Newsletter is published as an educational service by Prof Anthony Finkelstein at City University. If you wish to contribute send your material to the moderator at: requirements@cs.city.ac.uk

Subscription (or removal) requests should be sent to: requirements-request@cs.city.ac.uk Send an email containing
subscribe <address>
or
unsubscribe <address>

The Requirements Engineering Newsletter is archived at: http://web.cs.city.ac.uk/homes/acwf/rehome.html
or ftp://ftp.cs.city.ac.uk/pub/ (Files are called renl1, renl2, etc)

Software Requirements Engineering Mailing List

To subscribe to the Software Requirements Engineering (SRE) mailing list, e-mail listproc@jrcase.mq.edu.au, with the only line in the body of the message:
subscribe SRE your-first-name your-second-name

Articles to the SRE mailing list should be sent to SRE@jrcase.mq.edu.au.

RE-Actors

The committee of RESG

Chair: Dr. Bashar Nuseibeh, Department of Computing, Imperial College, 180 Queen's Gate, London SW7 2BZ. ban@doc.ic.ac.uk, Tel: 0171-594-8286, Fax: 0171-581-8024

Treasurer: Dr. Neil Maiden, Centre for HCI Design, City University, Northampton Square, London EC1V OHB. N.A.M.Maiden@city.ac.uk, Tel: 0171-477-8412, Fax: 0171-477-8859

Secretary: Michael Bearne, Philips Research Labs, Cross Oak Lane, Redhill, Surrey RH1 5HA. bearne@prl.research.philips.com, Tel: +44 (0)1293 815428, Fax: +44 (0)1293 815500

Membership Secretary: Dr. Sara Jones, School of Information Sciences, University of Hertfordshire, Hatfield, AL10 9AB. S.Jones@herts.ac.uk, Tel: 01707 284370, Fax: 01707 284303

Industrial Liaison Officer: Dr. Orlena Gotel, Dept Computer Science, City University, Northampton Square, London EC1V OHB. olly@soi.city.ac.uk, Tel: 0171-477-8000 Ext. 3700, Fax: 0171-477-8587

Publicity Officer: Dr. Andrew Vickers, Department of Computer Science, University of York, Heslington, York Y01 5DD, andyv@minster.york.ac.uk, Tel: 01904 434727, Fax: 01904 432708.

Newsletter Editor: Dr. Steve Easterbrook, NASA/WVU Software IV&V Facility, 100 University Drive, Fairmont, WV 26554, USA. steve@atlantis.ivv.nasa.gov, Tel: +1 (304) 367-8352, Fax: +1 (304) 367-8211