BCS™

Requirements Quarterly

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

© 2006 RESG http://www.resg.org.uk RQ42 (December 2006)

Contents

RE-soundings

From the Editor

In this issue, we're taking some of our own advice about **involving stakeholders** – you! You are invited to participate in a survey of how you actually define your requirements. Do you write a mission statement? Do you model your goals and assumptions? Do you simulate the behaviour of your future systems? Do you do everything in Microsoft Office? We'd like to know. Help us to put some proper numbers on our intuition about how Requirements are really done. We suspect the situation is richer and more varied than some people give it credit for – but more cautious and pragmatic than some might hope. See RE-calls for how to join in. We'll send you the results.

This issue also reports on the **joint IET/RESG requirements event** held at the start of October. Several members of your Committee spoke on different topics in RE. One thing that came up was the word "**Engineering**" itself. An extremely experienced consultant, formerly a manager of £1Bn projects in the oil industry, said that he'd never heard of the field of Requirements until this event, and tellingly he said he'd always done a stakeholder analysis on his projects – but he'd never ask an Engineer to do such work!

Perhaps it is about time that we dropped the E from our title – it may have been trendy in 1992 but it seems only to be confusing people, or worse, putting them off from attending our events when in fact what we offer is highly relevant.

"Requirements Specialist Group" sounds to me solid, serious, sensible, practical. In fact, I rather fancy being a "Requirements Specialist". How do you "engineer" a participative inquiry workshop with a mixed bunch of stakeholders, anyway? Sounds like the jokey phrase "Domestic Engineer" (ie housewife). What do you think – is it time we dropped the E? Let RQ know – if your views are in favour, RQ will propose the change to the Committee. But it's rather more than just a name change, isn't it?

Ian Alexander,
Scenario Plus

Chairman´s Message

Much of my teaching is on requirements. This often meets with bemusement by students on our Computer Science programmes which tend to have a strong systems architecture flavour. The point of why it makes sense to model a business problem, explicitly banishing, if only temporarily, all thoughts of how to implement the software takes some getting across.

To help do this for our MSc students, I usually invite external speakers to give an industry perspective on the problem. This year, I managed to persuade both our very own RQ editor and Simon Hutton of Headstart Analysis to give talks. They both kept the students rapt with their mix of stories from the trenches and lucid descriptions of how to tackle problems for which sheer scale poses one of the greatest barriers to success.

Simon made a point that I hadn't fully realized; that the UK faces a serious shortage of skilled systems engineers with requirements expertise. Needless to say, this was just what I wanted our MSc students to hear.

Could the cancellation of this month's RESG PhD workshop be another, albeit tangential, indicator of the skill shortage? The paucity of interest could be due to a number of reasons, of course. We might not have advertised it enough or the timing or location might have been bad. And of course, the health of the community of PhD students doing requirements work is unlikely to be strongly aligned to practice in industry.

I do worry, though, that it might just be an indicator of a decline in the perception that requirements work has sufficient to interest a developing engineer or computer scientist. I can scarcely imagine anything more interesting and challenging (and intimidating) than being a requirements engineer (sorry Ian), business analyst, product manager or whatever you want to call it on a project with lots or stakeholders, business processes and legacy technology. Maybe I'm out of touch (as with my taste in popular music – see later in this issue) and others don't share my enthusiasm. Or maybe I and others in the community are failing to communicate the great things about working on requirements. If you think we're missing a trick on how to stimulate people to choose careers in requirements, do let us know.

In the meantime, let me wish all our readers a wonderful, requirements-evangelizing, 2007.

Pete Sawyer,
Computing Department, Lancaster University

RE-treats

For further details of all events, see www.resg.org.uk Forthcoming events organised by the RESG:

Service-Centric RE

7 March 2007, City University, London

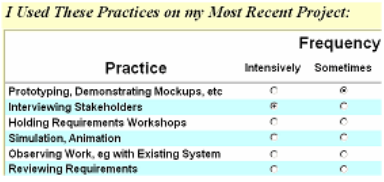
AGM and Lecture

July 2007 - Date TBC, London

RE-calls

Recent Calls for Papers and Participation

Survey: How Do You Define Your Requirements? An Invitation…



I Used These Practices on my Most Recent Project:

Practice	Frequency: Intensively	Frequency: Sometimes
Prototyping, Demonstrating Mockups, etc	○	◉
Interviewing Stakeholders	◉	○
Holding Requirements Workshops	○	○
Simulation, Animation	○	○
Observing Work, eg with Existing System	○	○
Reviewing Requirements	○	○

Three members of the RESG committee – Ian Alexander, Suzanne Robertson and Neil Maiden – are collaborating to investigate the question of how people actually do their requirements. Do you write "shall" statements? Do everything in UML? Document your assumptions? Build prototypes? There's a huge range of possibilities, and we suspect people mix'n'match.

Leaving aside all the big talk about goals and SysML and scenarios and suchlike, we'd like to know what you really do on your projects.

You are invited to visit

http://easyweb.easynet.co.uk/~iany/survey.htm

and complete the survey form. There are just 5 structured questions, all answered by clicking on radio buttons or check boxes.

Please take a few minutes to help us in this work. If you leave your email address, we'll send you a copy of our results.

By all means let your friends know about the survey, but to keep spam down to acceptable levels, please don't post the address on the web!

Working with the Experts!

In the first quarter of 2007, Mithun Training & Consulting will organise three Master classes "Working with the Experts!". These classes will run in Utrecht, centrally located in the Netherlands.

During these master classes, requirements engineers and requirements managers will be able to exchange ideas and view directly with their heroes.

By providing you direct access to some of the greatest minds in our discipline, we will provide you with new insights. They will help you improve your grasp on the requirements engineering process.

Our three experts have earned their track record in their domains and will discuss requirements management & engineering from three different perspectives.

- Dorothy Graham, from her broad experience in software testing;
- Ian Alexander, from working with stakeholders;
- Jeremy Dick, from traceability of information.

15th of February 2007	Masterclass by Dorothy Graham	Software Testing Essentials for Requirements Engineers
15th of March 2007	Masterclass by Ian Alexander	Requirements Engineering Workshop
19th of April 2007	Masterclass by Jeremy Dick	10 Principles for Writing Better Requirements

http://www.mithun.nl/html/master_classes1.html for full details of the masterclasses.

Mastering the Requirements Process

26-28 February 2007, London, presented by Suzanne Robertson, Atlantic Systems Guild

This 3 day seminar & workshop presents a complete process for eliciting the users' requirements, testing for correctness and recording them clearly, comprehensibly and unambiguously. Delegates will learn to:

- Determine their client's needs, and know they are correct
- Write requirements that are complete, traceable and testable
- Precisely define the scope of the project
- Identify the appropriate stakeholders and keep them involved
- Get the requirements quickly and incrementally
- Use up-to-date techniques such as storyboarding and e-collaboration

www.irmuk.co.uk/1/ for full seminar details, contact IRM UK on +44 (0)20 8866 8366, or e-mail customerservice@irmuk.co.uk

Mastering the Requirements Process Part 2

19-20 April 2007, London, presented by Suzanne Robertson, Atlantic Systems Guild

Good requirements are crucial for good systems. This seminar and workshop is about better requirements. This is an advanced course: it improves the skills of experienced business analysts, and teaches how to use the requirements deliverables for project management.

- Choose the best set of requirements
- Identify techniques for quantifying the business value of your requirements investment
- Learn how to anticipate market opportunities
- Understand how to deal with requirements for existing systems
- Learn how to discover the correct stakeholders for your project

www.irmuk.co.uk/58/ or see contact details above.

Introduction to Requirements

24-25 April 2007, The IET, Savoy Place, London, presented by Ian Alexander, Scenario Plus

This 2 day course introduces the requirements process, in the context of engineering a system.

The course covers the whole process from launching the project, through discovering the requirements, prioritising, formalising, and validating them. The use of tools to manage requirements is explored, along with requirements reuse.

The course is always a lively mix of explanation and practical exercises to get you familiar with applying effective requirements techniques.

www.theiet.org/courses for details and bookings.

RefsQ'07

The 13th International Working Conference on Requirements Engineering: Foundation for Software Quality, Trondheim, Norway. 11-12 June 2007 http://www.refsq.org

For its thirteenth birthday, REFSQ evolves into a working conference!

However, REFSQ will perpetuate its tradition of being a highly structured and interactive forum for researchers and practitioners to address the problem of ensuring software quality through requirements engineering (RE).

RE is as integral to the assurance of software quality now as it was when the first REFSQ took place in 1994. Compared to 1994, our understanding of RE has improved, while newer and better methods and tools are available to practitioners. At the same time, however, new challenges have emerged.

REFSQ'07 seeks reports of innovative work in RE that enhance the quality of software and systems, particularly where challenged by new development paradigms or technologies. We encourage researchers and practitioners from the RE, software engineering, information systems, and embedded systems fields to

present original work. Contributions from cognate areas such as formal methods, systems engineering, economics and management and social sciences are very welcome for the insights they provide in RE.

RE'07

15th IEEE International Requirements Engineering Conference, 15-19 October, 2007, Delhi, India

Understanding Requirements in the Global Economy

As software development is now part of the global economy, requirements engineering is the key bridge between the customer and supplier. Understanding and translating users' needs into effective solutions has always been vital: however, as development is outsourced requirements have to reflect cultures and languages and local needs. Furthermore, understanding requirements becomes a collaborative activity across time and space.

The IEEE International Requirements Engineering conference provides the premier international forum for researchers, educators and industrial practitioners to present and discuss the most recent innovations, trends, experiences and concerns in the field of requirements engineering.

RE07 focuses on the international context for requirements engineering; as off-shoring and outsourcing become increasingly common, issues of culture and localisation become critical. Requirements engineering itself will change as it becomes a 24/7 collaborative activity across national boundaries. Globalisation highlights the problems and will stress-test the solutions that already exist in RE, so particular emphasis will be placed on:

- RE in the global economy
- Collaborative Requirements Engineering
- Requirements, culture and localisation

Submission Deadlines

Paper Abstracts	5 February 2007
Paper Submissions	12 February 2007
Tutorial, workshop and panel submissions	9 March 2007
Notification to authors	27 April 2007
Doctoral symposium, poster and other submissions	11 May 2007

http://www.re07.org/

RE-readings

Reviews of recent Requirements Engineering events.

IET/RESG Requirements Event

Answering the Six Questions about Requirements

3rd October 2006, the IET, Savoy Place, London

I keep six honest serving-men
(They taught me all I knew);
Their names are **What** and **Why** and **When**
And **How** and **Where** and **Who**.

Kipling

Ian Alexander introduced the seminar and then spoke about WHO: Stakeholder Analysis. People talk about User and System Requirements, as if there were only two kinds of people – users and system people, presumably. But people have many valid interests in a project: financial, political, as actual operators, as beneficiaries, managing and purchasing products, as regulators, even as enemies. A rather richer view of stakeholders seems to be necessary. Some terms that people imagine are basic, like users, seem to be composites; a user is someone who is both the normal operator of a product and the functional beneficiary of that product.

Alexander therefore proposed a template for stakeholder roles, and showed using 'onion' diagrams how these related to the product and to each other. It was usual, he suggested, for stakeholders to have multiple roles. Equally, they often want systems to behave in incompatible ways. It is best to find this out early, and to negotiate the conflicts out of the system.

Discovering the stakeholders enables you to ensure you have not missed whole groups of requirements. Identifying negative stakeholders enables you to discover and allay their concerns, or at least to trade them off. In the case of hostile stakeholders, identifying the threats they pose at least allows you to devise mitigations. Finding an empty stakeholder 'slot' suggests you may have missed essential requirements. Finding a multiply-filled slot suggests the project may experience tensions as stakeholders try to pull the project in different directions. You can also think about "contractual fault-lines" between groups of stakeholders – again, these can be drawn on the onion diagram. At least that enables you to consider how to deal with the drop-offs that are bound to occur across such boundaries.

Nadia Amin (University of Westminster) spoke about WHAT: Goal Modelling. This is all about organisational change, she argued: what are the goals of the organisation – for how it wants to become. Objectives are vague high-level goals; true goals are more specific and should be measurable. Goals are "the objects of a person's ambition or effort", or "the state of affairs that a plan is intended to achieve". Goal modelling aims to express the purpose of a business' processes, and also to express how the objectives can be realised. Eventually all goals will be realised by

business processes, she claimed. These contain roles, objects (presumably hardware as well as software), activities, and business rules.

Goals can be defined informally in text, by brainstorming or by SWOT (strengths, weaknesses, opportunities, threats) analysis. More formally, goal graphs (diagrams) can be used. But simple goal diagrams can't express the fact that goals relate to issues, which arise because stakeholders want different things. A goal graph is an AND/OR tree of goals, met by alternative or complementary subgoals.

Goals can usefully be classified in a matrix (ie each goal can belong in several categories). For example, "Reduce demands on human operators" is both a usability and a safety goal. To meet such a goal, you need to identify new goals which can be implemented through business change; you have to choose between the alternatives, the candidate subgoals. Thus you only ever deploy part of the goal tree.

Goals of the enterprise can only be discovered through workshops which bring together the stakeholders, resolve any conflicts, and hence arrive at agreed enterprise goals to bring about business change.

In the discussion that followed, we agreed that requirements come from goals – they are particular ways of realising goals, specific solutions. In banking, for instance, goals are targets which are explicitly measurable in financial terms. But one person's goal is another person's requirement: generally, from the perspective of an individual project, a business goal appears as a high-level, direction-setting target, such as to run 40 trains an hour or to provide a journey time capability.

Neil Maiden (City University) spoke about WHERE: Scenario Modelling. The RE process consists in embedding systems in their environment, rather than on the prescription of the system's functionality or structure, he quoted (Jarke et al 1993). Similarly, Michael Jackson's definition of a requirement as describing the phenomena shared between machine and environment emphasises the vital connection between thinking about results in the world and what a system has to do. Hence, a technique like scenario modelling that can explore the environment is valuable for doing a decent (good enough) requirements engineering job.

Scenarios allow us to describe the situations and events that new systems must handle; situations that new systems will change; a structure for walkthrough meetings to discover missed requirements; and a common ground for communicating with stakeholders. Scenarios are thus very powerful.

There are various kinds of scenarios, ranging from stories to richly-structured analyses (Alexander and Maiden 2004). UML may be much-derided, said Maiden, but Jacobson's original 1994 definition is still useful: "a description of a set of actions, including variants, that a system performs that yields an observable result of value to a particular actor" (ie meets a goal). Use Cases can be applied to structure the requirements: under each scenario step you can group the requirements of whatever type that relate to it – usually one or more functions, together with training, user interface, safety and other requirements. This is far more readable, and hence more powerful, than having them scattered all over the project documentation.

Scenario representations can equally vary widely. For example for a warship, threat scenarios can be drawn directly on a radar-like plot, showing how a missile attack on the ship could proceed and what could be done about it.

How much detail should we go into when describing the environment, the situations and context where the system must operate? Do we talk about a passenger, or a mother with young children, or a persona ("Jane", who has kids Anna aged 7 and Mick aged 3)? These offer differing benefits to a project.

After a wonderful lunch in the sunny Riverside suite, with the balcony doors thrown open giving lovely views over Westminster and the City, **Simon Buckingham Shum** (The Open University) spoke on WHY: Rationale Analysis.

A design rationale is a representation meant to help you reconstruct the context of a design activity, specifically to illuminate why it is the way it is, to help some subset of system stakeholders with design.

This can take the form of text, a persuasive argument, a formal notation, a methodology, documentation, or an explanation in response to a query. (Buckingham Shum said he had no idea what knowledge management really was, and doubted it was feasible.) Useful design rationale (DR) does not come for free: it comes at the price of something else, possibly making things worse on your project. DR can be risky, eg if you make yourself traceably the source of a design approach that causes an accident (say, the hard rubber for the Shuttle's O-rings).

Design problems are often "wicked" (Rittel's term). These are problems with no definite number of solutions, no stopping rule, no proof that you have solved them. Every attempted fix changes the problem, so you only have one shot at a solution – no scientific approach is possible. It's the real, messy world of strategic planning, where requirements and assumptions constantly shift.

Rittel came up with IBIS, an issue-based information system, to try to deal with such wicked problems. This is a colourful graphical tool for diagramming explanations (big purple question marks for issues, big yellow light-bulbs for candidate solutions etc). The Open University's Compendium tool is freely downloadable and open-source, by the way (www.compendiumInstitute.com). Buckingham Shum impressively demonstrated (in real time) that he could make a helpful, intelligible graphical model of Alexander's Stakeholder classification, and document a

discussion or decisions about it, in a few moments. DR can thus capture anything from meeting minutes to the full justification for a project. IBIS templates provide a quick way of recording a DR, structured in a predefined way.

Another approach is QOC design space analysis – Questions, Options, and Criteria. Again, this leads to a visual analysis of an argument – the options available to answer a question, and the criteria for preferring different options.

Rolls-Royce have adopted yet another version of IBIS, DRed (the design rationale editor). This analyses text into arguments for and against, and makes each one an object overlaid with the relevant fragment of the original text, on a diagram. See www.edc-eng.cam.ac.uk/kim/people/rob-bracewell.html.

Experience at NCR showed that the DR spanned the experience of many different stakeholders, and hence revealed 7 critical, interdependent design problems. This had a payoff 3 to 6 times the effort invested.

Other applications include modelling emergency responses, political options in a given situation, NASA mission control, and many more. The

Two good books on DR are Visualizing Argumentation (Kirschner et al, 2003) and Rationale Management in Software Engineering (Dutoit et al, 2006). Jeff Conklin's Dialogue Mapping (2005) describes the skill of IBIS mapping, live in meetings.

Suzanne Robertson (Atlantic Systems Guild) spoke on HOW: Making Requirements Verifiable. You don't have to wait until you have a complete set of atomic requirements: you can start verifying as soon as you have some high-level goals (setting the large targets for more detailed requirements); you can even start verifying that your list of stakeholders is correct. High-level doesn't mean "vague and waffly": it means large, and by implication possibly long-term; but there's no reason not to put measurable targets on large goals.

The quality gateway is a single point where you verify your requirements knowledge at any level. That means finding a consensus on objective measures for every requirement. For instance, you can check that a requirement is relevant if it maps back to a business process; that it's viable, if the designers can show a believable way of building it; that it uses consistent terminology, if it fits with the project dictionary (rather than using "slidy" terms).

Andrew Farncombe (JBA) spoke on WHEN: Triage and Prioritisation. Projects often need to classify their requirements by when they can realistically implement them: as the children's rhyme for counting cherry stones has it, this year, next year, sometime, never. Prioritisation is something of a Cinderella of RE: everybody finds her useful but nobody talks about her much. There are however quite a few practical techniques, often very simple, for prioritising requirements.

At the simplest, a change control board can decide which requirements to include, or more democratically you can take a vote. More elaborately, you can facilitate a meeting to vote on several points along a scale (such as Essential – Important – Useful – Luxury). That gives you a curve for each requirement rather than a single point. A sharp peak then means good consensus; a blunt peak means some doubt in people's minds; a double peak means definite ambiguity.

Alternatively you can give all your criteria a weight, and derive weighted scores which you simply add up. This is quick and easy, but it risks melting all the factors down to a "brown soup" in which the apples, oranges and everything else are reduced to a single number.

If you think that a bad idea, you might prefer, said Farncombe, to analyse the variations between criteria so as to find out what you should be paying attention to – and then make your decision humanly (not by spreadsheet) on the basis of the evidence. You could use Kendall's coefficient of concordance to look for agreement between rankings. You can then use Kendall's tau or Spearman's rank correlation coefficient to look for patterns – do some criteria correlate positively or negatively with each other?

Finally, you can use Principally Component Analysis (PCA) to reduce the number of dimensions (from the original number of criteria) by calculating eigenvectors. PCA also shows you how good the correlations are between each new dimension and the original criteria. This elegant technique may help you see what really matters, and hence to choose rationally: any components that have eigenvalues greater than 1 will account for a large amount of the variation. In fact, the power (eigenvalue) of each new dimension (eigenvector) can be viewed directly as a percentage of the total amount of variation. PCA also shows whether correlations are positive or negative, so you can see if some of your criteria are pulling in opposite directions. During optioneering (trade-offs) this is potentially a very valuable technique for identifying the best options based on multiple criteria for comparing them.

Kathy Maitland (University of Central England) spoke on WITH WHAT: requirements tools. She suggested that many RE processes are essentially human: tools can only do so much. Drawing diagrams such as use case diagrams, data models and flowcharts (activity diagrams) can be helpful, but they don't help you decide whether they are what you want – humans are needed for that. Model-drawing (CASE) tools can draw diagrams and check their syntactic correctness, but that's about all.

RM tools have several key functions. These include version control, traceability and the ability to retrieve documents, models and pictures. Of these, traceability is central. A trace from a requirement back to a goal can act as the requirement's justification or rationale – it's there to help satisfy the goal. Traceability is also

vital for identifying dependencies, and hence for assessing the impact of change.

Simple tools like Word and Excel are fine for listing requirements but quite inadequate for all the other RM tasks like version control, change management and traceability.

The more heavyweight RM tools like Telelogic DOORS and IBM Rational Requisite Pro thus differ from basic office tools in permitting point-to-point (eg requirement to test case) traceability, and in providing varying degrees of control over changes and versions.

Discussion from the floor ensued. One participant suggested that DOORS and suchlike tools were too hard to set up. Other people said that no matter the tool, their users just created a list of requirements. Different tools would be needed for special tasks like risk analysis; while simple projects needed few and simple tools that needed little setting up.

The speakers then formed a panel for the closing **Question Time** discussion.

Kathleen Maitland, Simon Buckingham Shum, Andrew Farncombe, Suzanne Robertson, and Neil Maiden in the Question Time panel session

What is the difference between a Wish and a Requirement? It depends on the cost of getting it wrong: if it's large, you should be ready to do a lot of work to clarify the requirements.

What if the users are just asking for the same requirements as last time, right down to the data fields? Perhaps they are not being allowed to think out of the box: creativity workshops are one answer. Or perhaps there are some new stakeholders who have further requirements.

How to do requirements where there is rapid change and high risk? It's a pointer to choosing life-cycles to control risk – certainly not a big-bang waterfall life-cycle approach, especially.

What is a requirements engineer? Do universities teach it? Industry is learning that there are test engineers, but requirements, well. City University does teach it; other places teach software engineering, analysis, and so on. There is a large overlap between systems engineering, RE, and so on. In Germany there is a big move towards certification of requirements engineers (at a lower level than Chartered Engineer). In the UK we're not so sure.

A manager from the Oil industry said he'd never heard of requirements until this event, but he'd heard many useful (conceptual) tools in this seminar that he'd take back with him. What word did the Oil industry use instead? It is "Basis for Design". But Oil projects generally run well and are tightly managed. He always carries out a Stakeholder Analysis but he certainly wouldn't use an Engineer for that. The panel agreed: Requirements work is a socio-technical thing, not "Engineering" as generally understood. "Engineer" is a stereotype with both positive and negative connotations. Similarly, people in the rail and defence industries often say systems integration rather than systems engineering.

We reached 5:30 still talking warmly. Several of us adjourned to the Tup tavern round the corner to continue the discussion. We felt that it had been a very successful event, with a good buzz, lively participation throughout, and many interesting ideas discussed by people from a wide range of backgrounds – including experienced people new to requirements work. The strong element of collaboration between the IET and sister institutions – the BCS RESG and UK INCOSE – was especially welcome. Long may it continue.

Early Aspects

Lancaster University, 2nd November 2006

On a beautiful autumn day, the Lancaster University's InfoLab21 showed its best aspect to the RESG event participants. Our regional events do not attract the biggest crowd (the audience was 20+ strong) but they definitely attract sun and enthusiasm to go with it which kept us all captive during a fully packed afternoon schedule.

The first presenter, **Awais Rashid,** was the right choice to gently introduce the topic of "Aspect-Oriented Requirements Engineering" (AORE) as he leads research in aspect-oriented software engineering in the Computing Department at Lancaster University. Focusing one's attention on a certain aspect while not completely ignoring the other ones – so called "separation of concerns" – was first mentioned by Dijkstra [E. Dijkstra, *Discipline of Programming*, Prentice Hall, 1976, pp.210]. Rashid illustrated the need for separation of concerns on the development of large complex distributed systems. He also introduced the problem of crosscutting concern, a broadly scoped property such as security, mobility and distribution, which has an affect on multiple requirements. How do we identify crosscutting concerns? We need to consider the dominant decomposition (goals, use case descriptions, viewpoints) of the requirements and look for scattered and tangled requirements. Crosscutting concerns affect modularisation hence need for compositional information which is currently not readily available in goal-based, viewpoint and use case models. Rashid pointed out that AORE **is** RE and that

we do not need to throw away our RE practices but should complement them with modular and compositional reasoning about stakeholder needs and benefit from improved understanding of the problem domain.

Anthony Finkelstein (Computer Science, UCL) talked engagingly about "Aspects and Consistency Management". In his view, conflict resolution is the essence of RE. One of the key motivations of early aspects is multiple perspectives which bring potential for the inconsistency. Mechanisms for composing perspectives entail resolving inconsistencies. Finkelstein argues that inconsistency is not a problem (and it might even be desirable) unless we want to action in the presence of it. Thus we need to think about managing consistency and not removing it. This means preserving inconsistency where it is desirable to do so, identifying inconsistency at the point where decisions are required, and removing inconsistency prior to taking action. However, managing consistency is not an easy task as the assertions can be vague, so that it is difficult to determine their precise meaning and how they relate, and inconsistencies can be hidden in a mass of other assertions or distributed across many assertions. Finkelstein's answer is: tools (to help establish, express and reason about relationships between formal languages and check consistency with respect to these relationships), tools (to visualise and track inconsistencies) and tools (to support resolution). In the rest of the talk he gave a brief overview of how repair and conflict resolution can be achieved using CLiX (declarative rule language for checking distributed documents) and Validator [xlinkit]. His vision for the future: end-to-end consistency management framework for industrial scale software development. This talk was followed by lively discussion which seamlessly continued into a short coffee break.

Instead of advertised talk on A Viewpoint-Based Approach for AORE by Ana Moreira and Joao Araujo, **Ruzanna Chitchyan** from Lancaster University gave a talk on "Using Natural Language Semantics for Identification & Composition of Aspectual Requirements". She outlined several aspects of Lancaster University's ongoing research on the AOSD-Europe project: aspect identification (EA Miner), crosscutting representation (CoCA), and temporal conflict detection (MRAT). Not surprisingly, Chitchyan continued from the point where Rashid has left us off. She and her colleagues are addressing a problem of insufficient support for concern composition in non-AO approaches by proposing AORE model. In this model, concerns and their composition are identified and represented using Early-AIM approach and EA-Miner tool. Composition-Centric Approach (CoCA) to requirements has led the team to proposing Requirements Description Language which is used to describe the semantics of the requirements relationships. Chitchyan's animated talk was cut short due to the very strict time limit as half of the audience wanted to catch an earlier train back South. However, I believe that participants got enough information to follow it up through the supplied list of recent publications by the research team (slides of all presentations are available on the RESG website).

Aspects play an important role in analysis and design. In his presentation **"**Model Composition in Aspect Oriented Modeling", **Jon Whittle** (George Mason University, Fairfax, VA, USA) told us why Aspect Oriented Modeling (AOM) is special and what are existing approaches to model composition. In AOM, model composition is crucial for complete analysis and overall understanding of the problem domain as well as an enabler for MDA/MDD. Models are at different level of abstraction (an aspect should crosscut other models at the same level of abstraction) and represent different perspectives, both structural and behavioural (an aspect should crosscut other models at the same perspective). On the example of Car Parking System, Whittle explained two AOM approaches: AspectJ-like and HyperJ-like. He also presented the findings of a small case study he used to evaluate expressiveness of these approaches. He argues that existing approaches for model composition in AOM are not practical. Whittle's approach is to use model transformation to do aspect transformation. He favours graph transformations for composition as there are no restrictions on a composition strategy used and they are applicable to any modelling language.

Unfortunately, there was no time for the discussion at the end, but I believe that the participants (especially AORE novices like myself) were skilfully immersed in AORE to make them wanting to learn more.

The BCS Roger Needham Lecture

Computer Vision & the Geometry of Nature
Dr Andrew Fitzgibbon, Microsoft Research UK

RQ does not often go to 'central' BCS rather than RESG events, but this one sounded too good to miss, and so it proved.

Dr Andrew Fitzgibbon

In the splendid surroundings of the Royal Society, Andrew Fitzgibbon communicated his enthusiasm and expertise in computer vision.

Back in the 1960s, the Artificial Intelligence (AI) pioneer Minsky thought that computer vision would make "a good summer project" for a research student.

40 years later, the problem looks rather harder than that. But, said Fitzgibbon, it has long been traditional for speakers on vision to spend the first 20 minutes explaining how hard a problem vision really is. Instead, he went on, he would start by showing how easy it was. If vision needed to solve all of the AI problem, then it's a hopeless case; we just don't know how the brain works, or how to make computers intelligent.

But maybe vision is easier than AI. For instance, to see 3 dimensions (3-D) with two eyes, a baby or anyone else does not need prior knowledge of the world. If you believe that to see a tree in a landscape, you have first to identify the tree in the image from each eye, and then to merge the tree-objects to form a stereo image, then clearly the task is immensely difficult. But the random-dot stereograms from the 1960s – patterns of red and green dots, arranged by computer to yield a stereo image of squares floating in space, for instance – prove conclusively that people (well, 90% of the population) can see in 3-D given no prior knowledge of shapes or lines or objects at all. The output is purely data-driven: it is a simple, algorithmic, bottom-up process. Or in Fitzgibbon's words, it demands only "simple, low-level geometry and a little appearance".

The task of recognising people, bicycles or cars (possibly partially obscured) in an image is obviously useful – for surveillance, in warfare, for congestion charging and so on.

In practice, given a reasonably good image with nice sharp corners and edges, the problem can be solved quite reliably. But in a disorientatingly plain image of woolly cloud or dull sky with few features, the computer may well fail. Actually, human vision gets disoriented in similar circumstances – large faraway objects can seem like small things, close up.

Fitzgibbon worked on a different problem: recovering the motion of the camera from a video film. The cameraman walks through an office, filming the usual dull cubicles, computers, shelves and files, turning corners as he goes. The camera records a mass of pixels. Each image is 2-D. Can the computer work out the shape of the room from this, the 3-D objects present, and the path of the camera? Again, the answer is usually yes.

Essentially, each 3-D point in the office (say, the top front right-hand corner of a monitor) moves in a wiggly track across the screen as the camera jiggles and advances. That track forms a list of x,y co-ordinates – a column in a matrix, in fact. The matrix of all the tracks of all the 3-D points the computer chooses to process is the input; the task is to recover a matrix of the unchanging 3-D co-ordinates of those points, and another matrix of the track of the camera.

This task is rather like cryptography, being the challenge of retrieving two matrices which when multiplied together give the message actually received by the eyes or the eavesdropper. The difficulty, of course, is the needle-in-a-haystack problem that there is a huge number of possible solutions, and there may or may not be good clues as to the answer.

On a nice clear scene, the computer "sees" very well, and builds a reliable model of the 3-D space. It can make mistakes, and does so 1% - 50% of the time depending on the image – fuzzy and fluffy is worst.

Retrieving the path of the camera is useful for dead reckoning – you (or a robot) can work out where you are if you know how you've moved through space, and where you were when you started. But the more immediate application is in film-making. You can make small models of the 500-foot high statues that framed the approach to the Falls of Rauros on the great river Anduin in Tolkien's *Lord of the Rings*.

Rock-solid integration of CGI with BOUJOU

The camera can then move down the real river in a real boat, filming the real cliffs. Back in the studio, a studio camera can move along the same track, reduced in scale, to film the small model statues. Then you can add the statues to the filmed river and cliffs. They will not "move" at all, as they are fixed in 3-D space, though while they (and the cliffs) were filmed from the moving boat/studio camera, their image jiggled all over the place from one frame to the next. The program, called BOUJOU, is already a commercial success, a must-have for inserting virtual (CGI) or model objects into a real scene convincingly.

Before BOUJOU, artists had to toil over every frame, trying to insert an image of the object (statue or whatever) from the right angle, at the right position, in the right size. It was practically impossible. Now it can be done routinely.

It seems that computer vision can be good but not perfect: perhaps our vision too works in much the same way. There is no proof of this but it's extraordinarily persuasive, and far more satisfying than all the old philosophical talk of Qualia (qualities of things that we perceive, eg the redness of pillar-boxes) and the problems of perception (did anything really exist out there?). The mathematics reaches corners that philosophy didn't. Showing that you can make a computer recognise objects, navigate, and build convincing images does rather show that the computer can see. Is it then intelligent? Well, perhaps seeing

doesn't need intelligence at all. Nor does chess. Nor … well, what does? Perhaps the brain is a collection of simple, stupid, algorithmic gadgets. Does that add up to intelligence?

The evening ended with a profusion of wine and canapés, and much relaxed and enjoyable conversation. Suddenly London seemed a more civilised place, the world a little more comprehensible, technology a little more impressive. Fitzgibbon is clearly a first-class researcher, and an accomplished speaker, to boot.

RE-writings

Teaching Writing for Engineers

Nicola Longden
n.longden@lancs.ac.uk

Writing has become a key part of the life of an engineer, whether working in industry, academia or as a student in the field. Gone are the days where Computer Scientists can expect to spend their working lives in blissful isolation, writing software and communicating only with their computer and the coffee machine, replaced with the need to generate endless progress reports and the dreaded D word: Documentation.

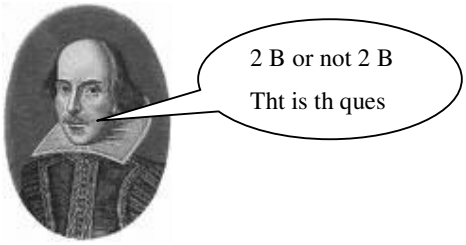
Documentation can feel like the Engineer's Bane

Communicating with the general public requires the ability to convert often complex, highly technical ideas into plain English that is clear, understandable and interesting to read. Employers expect graduates in any field to have a set of fundamental skills, with the ability to communicate through writing high on the list.

So where do people immersed in the engineering world learn the skills necessary to write in so many different styles for so many different purposes? Universities have to ensure that newly qualified graduates are furnished with these essential life skills. This seems to mean that people who enter engineering directly, without passing through a university education, have to somehow pick it along the way.

While writing skills are accepted as being important, students are entering higher education with ever growing deficits in this area. This is particularly pronounced when students have taken A levels that do not involve writing practice, most notably mathematically based subjects – the same subjects required for entry onto an engineering degree.

"Text speak" and the Internet have conspired to reduce the capability of students to construct sentences in the English language. If public opinion is to be believed, the quality of the education received at schools is diminishing year by year, as GCSEs and A levels get easier. It is no longer uncommon to find phrases such as "as u can c" cropping up in coursework submissions (they really do) and with the works of Shakespeare seemingly about to undergo a translation into the language of text messaging, things, in this respect, r only likely 2 get worse.



Student writing is not exactly Shakespeare

This forces universities to act as black boxes, taking in students whose ability in writing may be limited to GCSE English, and outputting graduates who can communicate effectively in the written word to their employers, their peers and the man down the pub. But how can this be achieved? In the past, the process of teaching writing in engineering subjects at universities has largely been left to "learning development tutors", practitioners in the art of writing or linguistics who could be trotted out at various times throughout a degree course to magically transform the student body into writers.

The main problem with this approach is that the development tutors feel as if they had the presence, influence and authority of a school supply teacher (and, unfortunately, many students seem to share this standpoint). Coming from a background in normally the arts or linguistics, tutors can sometimes miss the point of engineering writing due to assumptions, and maybe fear, about what it is that engineers actually do.

There is a misconception that the only document an engineer is ever likely to write is a technical report. Leaving the argument as to what constitutes a technical report aside, this ignores the work that engineers do in producing articles, instruction manuals, web pages, executive summaries and even, possibly, a story (poetry is perhaps a step too far). For good engineering writing, function is more important than form. While well constructed and witty prose may be a joy to read, brevity and clarity of explanation are more likely to ensure that bridges do not fall down.

From the perspective of the student, learning to write seems a black art, something that they are not taught directly on their degree course but which they are expected to somehow be able to do. While subject areas may have accepted a particular writing style and

vocabulary, individual teachers within the subject will have their own idea about what constitutes a good piece of writing. This is paralleled in the working world where one manager may want a short, concise summary of work undertaken, while another expects a weighty tome. This means that students are hardly better, or more confident, at writing than when they left school.

To try to remedy the situation and bring students up to the level expected by the real world, learning development tutors are starting to rebel, insisting that they do not work in isolation from discipline teachers and assessed courses. By embedding the teaching of writing into a degree course, taught by writing and subject specialists, students can be taught how to write like an engineer rather than simply how to write.

Don't Just Tell Them What To Do – Practise!

Improving writing skills is an iterative process, practice followed by feedback. Simply telling students what elements a piece of writing should contain is in no way enough to ensure that every report submitted by students will contain these sections.

Moving away from the bolt-on approach to teaching writing and embedding writing into courses should help all students to attain a minimum standard of writing ability. Many should exceed what they would have been able to achieve on their own or with remedial classes. But engineers who have already graduated, or who have never entered the university system, still have little help available to them – apart from adult education classes in creative writing.

As someone who has recently become involved in the process of integrating the teaching of writing into a degree programme, I am faced with three overriding questions:

- Do I actually know how to write myself?
- Is the style of writing that I am teaching to students correct?
- Is the proposed teaching approach effective?

While some examples of practice used in other universities do exist, and an embedded approach to teaching writing is widely believed to be better, there are no hard and fast rules for achieving this. There is no universal consensus on what constitutes good writing, nor on how to teach it.

The sort of tips that I give to students can be broadly grouped into the following areas:

- **Read.** Read as much of the work produced by other engineers as possible to build up a feel for the style of writing used.
- **Structure.** Have a definite beginning (introduction), a middle (the content) and an end (conclusion).
- **Use precise technical language**, the language that makes you sound like an engineer. No wouldn'ts, couldn'ts or sort of likes please!
- **Be concise.** Avoid waffling to fill up space. If something is not important enough for you to bother talking about, do not put it in.
- **Write in your own words.** As a reader, I hear the voice of the writer in my head. It is painfully obvious (the voice changes) when passages have been copied from other people. Differences in grammar, sentence structure and font are also flashing lights that indicate a hasty cut and paste job.
- **Tell the story** of what you have done from start to finish, whether it is work undertaken or research carried out.
- **Keep it short.** Start, say what you have to say and then stop.
- **Plan.** Aim for a coherent body of writing that flows from section to section rather than simply putting disconnected ideas on a piece of paper.

The way to improve writing is through practice and feedback. The same is true for teaching of writing. Any feedback on these tips would be greatly appreciated, particularly from industrial practitioners. Please feel free to send me an email.

Visualizing Requirements

by Sameer Wadhwa, *Infosys Technologies Limited*
6607 Kaiser Drive, Fremont, CA 94555, USA

Sameer_Wadhwa01@infosys.com

Visualization – a form of Prototyping – has long been recommended by researchers for specifying and validating requirements. Recently, interest in the technique has increased, because

- Several tools that support rapid prototyping by visualization, such as IRISE, STPsoft and Sofea's Profesy, have come to market.
- Many legacy applications are being converted into client-facing portal-based applications with new user interfaces. Hence an increased focus on screens and visualization.
- As application complexity is increasing, visualization becomes an important medium to communicate between business and IT teams to ensure correct and comprehensive requirements.

The development of commercially available rapid prototyping and visualization tools has pushed usage of prototyping earlier in the development life-cycle. Projects now use visualization prototypes in the elicitation phase, where before they were applied during analysis and validation.

However, experience shows that if visualization is not used appropriately, it can still result in costly errors. In this article, we attempt to highlight key issues that can

arise from inappropriate usage of visualization technique and recommendations to address the potential pitfalls.

What is Visualization?

Visualization means using graphic techniques to assist in a requirements discussion. Graphical techniques may include drawing prototype screens, charts and models. These can help to facilitate a requirements session. Visualization can depict

- User Interface Requirements as wire frames (simplified models of screens without color schema/artwork) or more detailed prototypes
- Non Functional Requirements such as availability and performance graphically

Here, we are focusing on using visualization to elicit requirements.

How is Visualization used in Elicitation?

Your mileage may vary, but a typical visualization scenario is:

1. Requirement Analysts meet with the users and capture initial requirements in textual format.
2. Analysts create wire frame screen design prototypes, and go back to users for validation.
3. Users give feedback and suggest requirements on the basis of what they have seen.
4. Analysts create detailed formal requirements.

Visualization thus helps to elicit requirements from users.

Case Study: Developing A Bank's Portal

A leading midwestern US bank wanted to merge several applications into one client-facing portal. This would allow clients to log in to one website to access all the financial products they had purchased from the bank. The project team used a leading visualization tool to capture the requirements.

We used the visualization tool to prototype the proposed website. For the given functionality, a total of around 16 web pages were created capturing various scenarios. Besides the prototype, a high level natural language document was also created within the tool to capture rules and constraints. Text and prototype were handed over to the design team for development.

Visualization evoked a very participative and enthusiastic response from the user team. The elicitation sessions were lively and led to discovery of detailed requirements. At this stage we considered the approach a success.

During the course of interaction with the business team, the key scenarios were modeled on the prototyping tool. However, there were several "other scenarios" that were not identified or were intentionally left out.

The development team considered prototyping and associated documents to be the complete set of inputs for development. They did not get any standard artefacts such as detailed use cases or process models for creating the application.

When the development team came back with the application as per the requirements, the business team rejected it, as several scenarios were left out. This resulted in costly rework across the whole application.

Conclusion

Good elicitation achieved by visualization needs to be complemented with a rigorous, formal analysis of requirements to cover all significant scenarios (Doing everything is impossible. Ed.) and identify hidden requirements.

Visualization can contribute significantly where there is heavy user-system interaction. However, manual steps and behind-the-scenes (system-to-system) interface requirements are not susceptible to visual exploration. They need to be covered using traditional requirements techniques.

Hence, an ideal requirements process captures requirements using textual, visual and formal methods within the same environment. Analysts can then iterate and cross-check the requirements to achieve reasonable completeness.

RE-verberations

In the Nov/Dec 2006 issue of *IEEE Software*, Lisa Crispin argues the case for using test-driven development to improve software quality. Both studies and her personal experience say that it works.

Code (according to Crispin) was less buggy and passed more of the tests thrown at it. Less time was spent on debugging, and not surprisingly productivity improved.

Crispin is bold enough to make the claim that

> "TDD produces code that has orders-of-magnitude fewer unit-level bugs, far fewer functional bugs, and an exponentially [sic] higher probability of meeting stakeholder expectations".

In other words, planning for testing helps not only with testing but with coding and functional requirements too. Get those acceptance criteria in there!

RE-flections

Who's Requirements?

by Pete Sawyer

Hope I die before I get old
Pete Townsend of *The Who*

In the week following RE'06 I had the pleasure of seeing *The Who* play Madison Square Garden. Now in their sixties, the two surviving members of the original line-up showed no embarrassment including in their set the classic '*My Generation*', which famously features the words quoted by this piece's tagline. As hostages to fortune go, this takes some beating. Even if he didn't plan on the early demise of the two non-surviving *Who* members, Keith Moon and John Entwhistle, it's clear that Pete Townsend didn't anticipate still being bound to this particular piece of intellectual property when he wrote it back in the sixties.

A similar lack of foresight continues to infect software and systems development.

Despite the well-known benefits of requirements tracing, particularly for change management and impact assessment, practice is still very patchy. This became clear during the panel on traceability at RE'06, chaired by Jane Cleland-Huang. The panel included former RESG exec member Olly Gotel and RESG event regular Jeremy Dick; two people who probably know as much about tracing and wider requirements management issues as anyone. It was no surprise, therefore, that the panel provided a wonderful insight into the whys and wherefores of tracing.

So why is tracing not universally practiced? In complete contrast to the funny and provocative position statements of the panel members, tracing is boring, unrewarding work that no-one wins any plaudits for. It's the 21st century's analogue of the job in the drawing office where all the engineering drawings were versioned and filed away in a big drawer in the basement. No self-respecting draughtsman wanted to do it so it was farmed out to the apprentice or the temp – jobs that would have been familiar to many of *The Who*'s generation.

These days, there are good requirements management tools but they don't answer all the needs of effective tracing. Jeremy Dick made the point that it isn't enough to trace the requirements, we need to trace requirements *information* – all the contextual and source material as well as the requirements themselves.

Some people work to change practice for the better and to encourage the industry to improve its requirements management, with better tools and better staff motivation.

Other people are working to mitigate the effects of poor requirements management by, for example, the automatic reverse engineering of traces.

These are all encouraging developments with significant potential benefits for practitioners. Depressingly, however, it turns out that the risks of not doing tracing are not universally acknowledged after all. To illustrate this, Olly used a term that was new to me but comes from the agile development world:

YAGNI, or 'Ya ain't gonna need it'.

Olly wasn't arguing against tracing. She was illustrating how tracing is often seen as a pointless overhead. In an ideal world where development schedules were instantaneous, where elicitation methods were fool-proof and where nothing else changed during the useful life of the product, YAGNI might apply.

One area of the industry where YAGNI definitely does not apply to tracing was explored in a panel on the previous day at RE: product management. One of the things about product management is the investment that development organisations have in their products and their families and variants of products. In all of this, requirements tracing is absolutely fundamental to being able to manage a software product and enforce effective configuration management.

The need for tracing goes beyond product-lines, though, and even custom software is subject to changing priorities.

For example, RE has seen recent recognition of the need to help stakeholders reach consensus and resolve their competing priorities. The elicitation workshops run by this very newsletter's editor, the Robertsons and others at the leading edge of RE practice are testament to how important this is. Yet, reaching consensus is a complex and difficult process. People sometimes forget what's been agreed and why. Unless, these rationales can be recovered, transparently-agreed decisions become opaque and the consensus starts to crumble. This is just one of the things tracing can avoid.

So let's not be seduced by the prospect of saving marginal effort on an activity that we hope we'll never have to make use of and even then, some time in the

distant future. Like Moon and Entwhistle, we might die before we get old, but hopefully not if we avoid the excesses of their lifestyles and almost certainly not before we are faced with proposed requirements changes; probably next week, next month and onwards, deep into the development process.

So let's

> 'get down on our knees and pray,
> [that] we don't get fooled again'.

© Pete Sawyer 2006

RE-partee

Plain Wording

> "If you pay a cheque into your account we will pay interest on it from the day after the second bank working day after we get it (including the day we get it)."

Got that? Good!

Genuine item from a bank, by the way.

The letter went on to explain

> "For example, if you pay a cheque in on a Tuesday, we'll pay interest from the Thursday."

But the damage had been done…

Plainer Wording

> "Partricipants have to take part in an RE project as well ass sit a written exam." (sic)

Let's hope the cushions were soft, and no kicking ensued.

The Computer Shall Be Reusable

From www.armenianteens.com/blog/. But please remove all the plastic trim and covers before grilling.

Proverbs

Since we are on requirements and food already, let us continue. Probably the very oldest and truest requirements proverb is:

> The Customer is **Always** Right

I overheard a wonderful example in a Chinese Restaurant the other day. Two ladies of a certain age sat down, keeping their overcoats on (always a telltale sign, don't you think?). The waitress took their orders.

> "No mushrooms or seafood, please"

they said. Our table was piled with dishes full of the two ingredients.

> "Oh, and no pork."

(In a *Chinese* restaurant? What was the kitchen going to cook, then? Surely there was pork in every sauce?)

The waitress glanced at the two ladies to check she'd heard correctly, noted down the requirements, and vanished into the kitchen.

RE-Definitions

- Prototype – the version of the system that you're going to deliver
- Requirements Creep – a person who knows what SysML is
- Tacit Knowledge – the truth about your project that can't be mentioned to the boss
- Ambiguity – the fine print in the contract
- Company Glossary – list of obsolete acronyms
- Standard Procedure – what nobody does
- Going out to Consultation – making a decision and then ignoring what the public say about it

RE-creations

To contribute to RQ please send contributions to Ian Alexander (iany @ scenarioplus.org .uk).
Submissions must be in electronic form, preferably as plain ASCII text or rtf.
Deadline for next issue: 7th March 2007

RE-sources

Books, Papers

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:
www.cc.gatech.edu/computing/SW_Eng/resnews.html

IFIP Working Group 2.9 (Software RE):
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 template:
http://www.volere.co.uk

DACS Gold Practices:
http://www.goldpractices.com/practices/mr/index.php

Software Requirements Engineering Articles (India):
http://www.requirements.in

Mailing lists

RESG Mailing List

Subscribe: admin-mail-list@resg.org.uk with "subscribe" in the subject line.

RE-online

http://www-staff.it.uts.edu.au/~didar/RE-online.html

Subscribe: majordomo@it.uts.edu.au with message body set to "subscribe RE-online <your email address>"

Requirements Networking Group (RQNG)

www.requirementsnetwork.com

RE Yahoo Group

http://groups.yahoo.com/group/Requirements-Engineering/

RE-actors: the committee of the RESG

Patron:

Prof. Michael Jackson, Independent Consultant,
jacksonma @ acm.org

Chair:

Dr Pete Sawyer, Computing Department,
Lancaster University,
sawyer @ comp.lancs.ac.uk

Vice-Chair:

Dr Kathy Maitland, University of Central England,
Kathleen.Maitland @ uce.ac.uk

Treasurer:

Prof. Neil Maiden, Centre for HCI Design, City University,
N.A.M.Maiden @ city.ac.uk

Secretary:

David Bush, NATS,
David.Bush @ nats.co.uk

Membership secretary:

Dr Lucia Rapanotti, Computing Department, The Open University,
l.rapanotti @ open.ac.uk

Publicity officer:

William Heaven, Department of Computing, Imperial College,
wjh00 @ doc.ic.ac.uk

Newsletter editor:

Ian Alexander, Scenario Plus Ltd.,
iany @ scenarioplus.org .uk

Newsletter reporter:
Ljerka Beus-Dukic, University of Westminster,
L.Beus-Dukic @ westminster.ac.uk

Regional officer:

Steve Armstrong, Computing Department, The Open University,
S.Armstrong @ open.ac.uk

Student Liaison Officers:

Zachos Konstantinos, City University,
kzachos @ soi.city.ac.uk

Andrew Stone, Lancaster University,
a.stone1 @ lancaster.ac.uk

Immediate Past Chair:

Prof. Bashar Nuseibeh, The Open University,
B.Nuseibeh @ open.ac.uk

Industrial liaison:

Prof Wolfgang Emmerich, University College London,
W.Emmerich @ cs.ucl.ac.uk

Suzanne Robertson, Atlantic Systems Guild Ltd.,
suzanne @ systemsguild.com

Gordon Woods, Independent Consultant,
Gordon @ cigitech.demon.co.uk

Alistair Mavin, Rolls-Royce,
alistair.mavin @ rolls-royce.com

The Requirements Engineering Specialist Group of the British Computer Society

Individual Membership Form for 2007

1. Membership Type

Please indicate type of membership:

BCS/IEE members, please

BCS / IEE member (£10) [ ] *indicate membership number*_____

Non-BCS / IEE member (£20) [ ]

Full-time student (free) [ ] *Studying at:*_____

If you need a receipt, please tick here [ ]

Corporate Membership also available – details at www.resg.org.uk or ask the Membership Secretary

Payment by cheque only. Please make it payable to "The BCS Requirements Engineering Specialist Group"

---

2. Your Details Title: Mr/Mrs/Ms/Dr/Professor/Other:_____(delete as appropriate)

First Name:_____Surname:_____

Address for correspondence:_____

_____Postcode:_____

Phone:_____Fax:_____

E-mail address: *Please write your e-mail address clearly using BLOCK CAPITALS*

As an RESG member your e-mail address will be added to the RESG mailing list

Optionally, indicate:

Your organisation's name:_____

Its type of business/domain:_____

3. Your Specific Interests

Areas of interest for the RESG given at *http://www.resg.org.uk/about_us.html*. Is there another area would you like us to add?

---

4. Your Participation Preferences

Please indicate what timings/duration for RESG events would suit you:

Whole day [ ] Half day [ ] Evening [ ] Other (please specify) [ ]_____

Please indicate whether you would be willing to help the RESG with:

Publicity [ ] Newsletter contributions [ ] Organisation of meetings [ ]

5. Your Mail Preferences

The RQ quarterly newsletter is sent to all RESG member as a PDF e-mail attachment unless you prefer a hard-copy of RQ delivered by post. For hardcopy delivery, please tick here [ ]

I understand that the information supplied on this form will be held in a database and used for the purpose of distributing information relevant to the activities of the RESG.

6. Your signature:_____Date:_____

Please send this form with payment (if applicable) to: Lucia Rapanotti
RESG Membership Secretary membership-RESG@open.ac.uk Fax +44 (0)1908-652140
Computing Dept., The Open University, Walton Hall, Milton Keynes, MK7 6AA, U.K.