



Requirenautics Quarterly

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

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

©2003, BCS RESG

Issue 30 (December 2003)

RE-Locations

RE-Locations	1	RE-Readings	4
RE-Soundings	1	RE'03	4
Editorial	1	Tom Gilb presents 'Planguage' a planning language	
Chairman's message	2	for requirements and more	5
RE-Treats	2	Creativity, the Path to Innovative Requirements	6
R-Day'04	2	RE-Papers	8
Engineering Organisational Solutions from Human		Innovating Requirements for Process Control Systems:	
Information Requirements	2	Ideas Based on Recent Events of RESG	8
RE-Calls	3	The Orthodox Barber or, What Tools Shouldn't Be	
Third International Conference on Aspect-Oriented		Used For	11
Software Development	3	RE-Publications	12
Twenty-Sixth International Conference on Software		Classic Book: Designing for People	12
Engineering (ICSE 2004)	3	Classic Book: Philosophical Investigations	13
Tenth Anniversary International Workshop on		RE-Sources	14
Requirements Engineering: Foundation For Software		Mailing lists	15
Quality (REFSQ'04)	3	RE-Actors	15
Twelfth IEEE International Requirements Engineering		The committee of RESG	15
Conference (RE'04)	3	Requirements Engineering Journal	16

RE-Soundings

Editorial

Welcome to RQ30. This last issue of 2003 contains Ian Alexander's review of the Glastonbury Festival of the RE research world: the IEEE international conference on RE (RE'03). Ian was there in Monterey, suffering for his art, and assiduously attending everything that it was possible for one person to attend. Weren't there? Then let Ian's review help you decide what you missed and whether you should buy a copy of the proceedings.

This issue also contains an article on creativity by Islam A. EL-Maddah, a PhD student at King's College London. Islam's thoughts have been stimulated by recent RESG events, and in particular, by the 'Creativity' tutorial run by Neil Maiden and Suzanne Robertson (reviewed in this issue). Creativity in this sense, of course, is about unearthing requirements that best serve the stakeholders' goals. Yet the word creativity is an emotive one. It's frequently used in a pejorative sense, as in 'creative accounting'. By analogy, RE is sometime creative in the sense that

requirements are invented to compensate for poor elicitation.

Even when creativity doesn't have negative connotations, it sometimes fails to serve our purpose. In some peoples' minds, and most damagingly in those of school-leavers, creativity is increasingly seen as the preserve of the arts (or at best, of game programmers). Yet the engineering of systems is all about being creative, but in ways that are constrained by (e.g.) cost and the systems' environment.

Also in this issue are some easy to remember homilies from Gordon Woods. I'm collecting these for a crib sheet for my RE students. All contributions welcome, and considered for inclusion in RQ. Finally, don't forget that contributions published in RQ win a book for their author. Contributions from practitioners are especially welcome.

Pete Sawyer
Computing Department, Lancaster University.

Chairman's message

Longstanding RESG members may remember the burst of energy that the RESG committee had in 1997 and 1998, organising "RE-Day" in 1997 and a two-day version, CEIRE (Conference on European Industrial Requirements Engineering) in 1998. The two very successful events left the committee satisfied but exhausted, but enough time seems to have passed for us to remember the pluses only! So, stay tuned for Episode 3: Requirements Day (R-Day), scheduled for Wednesday 31st March 2004 in London. It promises to be a cracker, including keynotes, tutorials, workshops, posters, and a tools fair. Please do etch the date in your diaries - we need you to be there. Details of the programme will be incrementally added to the RESG website, but an overview is included in this newsletter as a taster.

R-Day will follow two days of meetings in London by the Programme Committee of RE'04: the 12th International Requirements Engineering Conference to be held in Kyoto, Japan in September 2004. The advantage of this is that some of the top RE

researchers in the world will be in London that week and will attend or participate in R-Day. Don't miss it!

The RESG is a sponsor of RE'04, and I would encourage you to submit a paper or two by the 26th January 2004 submission deadline. The UK RE community is one of the strongest on the world, and it would be nice to see this reflected in the conference programme, as it often is.

OK, all done for 2003. I hope you enjoyed the RESG programme of activities in 2003, including this fine newsletter! With the newsletter, existing RESG members will be receiving membership renewal requests that I warmly invite you to deal with expeditiously so that you continue to receive RQ in the New Year.

Best wishes for the festive season, which seems to have started early this year - or is it just me wandering into implementation too early?

*Bashar Nuseibeh
The Open University*

RE-Treats

*For further details of all events, see www.resg.org.uk
Next event organised by the group.*

R-Day'04

Date: 31st March 2004, 9.00 am
Location: Thistle City Barbican, Clerkenwell, London
Contact: Alessandra Russo (ar3@doc.ic.ac.uk)

R-Day'04 is a special full-day RESG event aimed at RE practitioners and researchers. There will be a full programme of parallel tracks presenting work dealing with many of the problems associated with the generation and management of requirements. There will be presentations by internationally known experts, mini-tutorials, poster sessions, a tools exhibition and a book display. Best of all, R-Day'04 will provide an opportunity for practitioners and researchers to share ideas and experiences while enjoying the exciting cultural diversity of the city of London.

The range of topics for R-Day'04 include:

- Ontology Development
- Requirements Evolution
- Non-functional Requirements
- Integrating Requirements, Design, and Testing
- Research Methodology
- Scenario-based RE
- Metrics and Measurement
- Agile Methodologies
- Building Bridges between Business and Development

- Problem Frames

Keep watching the RESG web pages (www.resg.org.uk) because further updates on registration, and submission details will soon be available online.

Engineering Organisational Solutions from Human Information Requirements

Date: 12th May 2004
Location: To be confirmed, London
Contact: Pete Sawyer

This one-day interactive workshop presents the MEASUR methods. Practical applications have yielded major benefits in system quality, cost reduction, and adaptability to changing business needs. They are based on analysing knowledge, responsibilities and meanings to produce rigorous specifications of human information requirements. From MEASUR specifications, supporting prototype computer systems can be generated directly.

MEASUR advances the theory by replacing the usual "information flow" model by one of "information fields" defined by the communities that share sets of norms (social, organisational or legal) that determine their members' information needs.

The workshop will be presented by Ronald Stamper and Yasser Ades.

RE-Calls

Recent Calls for Papers and Participation

Third International Conference on Aspect-Oriented Software Development

22nd - 26th March, 2004, Lancaster University, UK.
<http://aosd.net/conference>

The construction and evolution of a software system involves many different aspects including features, qualities, perspectives, use cases, collaborations, requirements, processes, and implementation structures. Aspect-oriented approaches provide new means for modularizing such crosscutting aspects of a system.

AOSD 2004 will bring together researchers and practicing software developers to discuss new ideas in aspect-oriented technology.

The conference program will feature technical papers, practitioner reports, workshops, tutorials, demonstrations, a code fest, and a student event. Keynote speakers include Dr. Daniel Sabbah (IBM Vice President, Development and SWG Technology, Application Integration & Middleware Division, Software Group) and Prof. Bashar Nuseibeh (The Open University). A panel, moderated by Dr. Brian Barry (Bedarra Corporation), will address AOP and enterprise development.

Twenty-Sixth International Conference on Software Engineering (ICSE 2004)

23rd - 28th May, Edinburgh International Conference Centre, Edinburgh, Scotland

<http://conferences.iee.org.uk/icse2004>

ICSE is the premier software engineering conference, providing a forum for researchers, practitioners and educators to present and discuss the most recent innovations, trends, experiences and concerns in the field of software engineering.

ICSE 2004 will offer an exciting program of events, including keynote talks by leaders in the field, tutorials, workshops and presentations of technical papers on innovative research and the cutting edge of practice. The main program will be complemented by an array of social events, providing further opportunities for informal networking.

Tenth Anniversary International Workshop on Requirements Engineering: Foundation For Software Quality (REFSQ'04)

7th - 8th June 2004, Riga, Latvia (In conjunction with CaiSE'04) www.refsq.org

At nine previous REFSQ workshops, researchers and practitioners from various disciplines have contributed to improve the definition and implementation of quality requirements. REFSQ'04 will continue this tradition as a highly interactive stage for discussion of quality related problems in requirements engineering as they have developed over the last years. In particular, we encourage people from the requirements engineering, software engineering and information systems fields to present their approaches to higher software quality and to discuss how requirements engineering can contribute to it.

Key dates:

- Submission deadline: 1st March 2004.
- Notifications to authors: 15th April 2004

Twelfth IEEE International Requirements Engineering Conference (RE'04)

6th - 10th September 2004, Kyoto, Japan
<http://www.re04.org>

The importance of requirements engineering has been recognised for many years. In the 1990s this recognition led to an IEEE Conference and Symposium series. Ten years on, the RE Conference has become the international platform for presenting new research, transferring research results to industrial practice, and presenting industrial experiences and best-practice to the widest possible audience. In 2004 RE will take place in Kyoto, Japan for the first time. To reflect this RE'04 will continue to be interested in all aspects of RE, but is particularly interested in requirements for embedded systems in automotive and consumer products, and requirements engineering for innovative product design. Key dates:

- Research paper submissions: 26th January 2004
- Industry experience report submissions: 26th February 2004
- Notifications to authors: 2nd April 2004
- Camera-ready papers received: 11th June 2004
- Workshop and tutorial proposal submissions: 14th February 2004
- Doctoral symposium submissions: 16th April 2004
- Poster and research demonstration submissions: 23rd April 2004

RE-Readings

Reviews of recent Requirements Engineering events.

RE'03

8th - 12th September 2003, Monterey bay, California

After a somewhat dizzying journey over the Arctic ice and the tundras of northern Canada, we were welcomed into the dazzling sunshine of the California coast. Outside, there were pelicans, whales and sealions in the bay; inside, many of the world's most distinguished requirements people.

On Monday there were tutorials by Ralph Young on developing RE skills, Don Gause on NFRs, and Michael Mannion and Hermann Kaindl on product lines. There were workshops on comparative evaluation (how do you know if one RE technique is better than another?) and open systems.

On Tuesday, Suzanne Robertson and I gave our tutorial on Stakeholders without tears (a reference to the onion-model of stakeholder responsibilities – see RQ 29 for a report on the RESG session of the same tutorial); Chris Rupp and Rolf Goetz gave one on UML called 'From the sentence to the perfect requirement', and Barry Boehm and Bob Briggs gave one on Theory W Requirements Engineering. There were workshops on high-assurance systems and adaptable architectures.

The conference proper began on Wednesday with a keynote from Vint Cerf, father of the Arpanet/Internet; he did try to mention requirements but essentially it was a history of the design choices and pitfalls, most of which he fell into. Who could have imagined that one would need more than the 4 billion Internet hosts provided by a 32-bit IP address space? A host for every person on the planet, each capable of accommodating hundreds of users, must have seemed more than enough in 1977, when there were just 4 hosts with an IP address! Clearly the decision was a trade-off between the requirement for growth and the constraints of current implementation, but beyond a rueful smile, Cerf did not discuss the point. Perhaps there is nothing useful that can be said about it, beyond the trite 'the future is unpredictable' or worse 'if it is successful that won't be enough'.

The technical papers began with two exceptionally interesting pieces of work.

Mark Bergman and Gloria Mark (University of California at Irvine) looked at how people actually do requirements analysis *in situ*. They looked at NASA's new millennium project, and showed with their own 'authority-activity' process charts that decisions start high up at organizational level, descend via project level to the technical level (yes, that's us) and then climb up to the top again for the bosses to decide if

projects should go ahead. If they say yes, then there's a second looping down-and-up cycle as requirements are analysed, evaluated and finally given the go-ahead at the three levels of power. In other words there is always a power struggle between the techies who say 'this is absolutely vital for it to work', the project managers who say 'only time for this and this' and the organization which says 'ah but we have to do this and that'. Requirements stabilise through a process of authority, design, sense-making, and negotiation (with unequal power). Like all good things, this sounds obvious, but did we ever hear it before? At the extreme right of the diagram (stage 8!) the implementation actually begins...

David Bush (UK NATS) presented a pathbreaking paper called Requirements Stability Assessment Using Scenarios. (Perhaps stability / adaptability was the theme of the conference.) His scenarios are like those pioneered by Shell (oil company), looking at imagined future worlds and asking if each requirement would exist in each of them. If so, it would probably be stable over the next 10-20 years. Since I had already written (commercial plug coming up, oops), in the introduction to the scenarios book that Neil Maiden and I are editing, that this form of scenario was rarely used in systems engineering, it was delightful to see a practical instance of it. The process of assessing stability involved developing a baseline goal model and alternative scenarios, conducting an initial stability assessment, propagating the instabilities to related goals (an optional, time-consuming step), analysing and re-plotting the instabilities (in colour on the goal tree/network), revising the goals and finally revising the architecture.

Mikio Aoyama (Nanzan University) and colleagues from Fujitsu spoke on multiple product line frameworks. They assemble each product line by composing frameworks at different levels. All product lines incorporate a base framework, a database, and a platform. Then any river system (for instance) incorporates the Waters framework and either the River Information Management, Water Supply Monitoring, or Dam Monitoring framework, etc. The claimed effect is to cut 60% from the functional design through integration test stages, or 40% from the whole life-cycle. This has been achieved by analyzing the requirements across all product lines and redesigning the layered application frameworks to maximise commonality.

Meanwhile, Neil Maiden was chairing the 'embedded' workshop on RE and COTS – it was impossible to be everywhere I wanted to be at once. There was a wonderfully creative discussion on the issues in COTS integration. Some of the problems are obvious – why bother to write the requirements for your integration when you already know that you'll be using Excel for

the calculations and graphing: there's no point in writing 'Excel shall be able to multiply real numbers' in your spec. On the other hand, if you don't say what you want, how do you hope to get it or test it? Clearly you need to give an overall picture and add detail for the interfaces between COTS packages. There was much more. Let's hope the notes get written up and shared, and that the effort continues.

On Thursday I sat in on the Posters and Demos session, and saw and heard several interesting ideas. A team from the University of Minnesota presented a tool for model-checking and proof, enabling "shall" requirements to be revised and improved. A Belgian team spun off from Axel van Lamsweerde's KAOS work showed off FAUST, a formal analysis tool forming an add-on to GRAIL. It checks scenarios to see if you can violate any postconditions, e.g. can you open the doors while the train is running. William Stufflebeam and Thomas Alspaugh demonstrated a lightweight scenarios tool. Martin Feather (CalTech) showed how his risk-mapping tool could be applied (with a bit of a mental twist) to making decisions about requirements.

After lunch I joined the panel session on Marketing meets RE. The panelists were Georg Herzwurm (from a business school), Neil Maiden (the required requirements engineer), Sanjit Sengupta (a professor of marketing), and Tuure Tuunanen (an economist). There was much good stuff about the differences between classical RE for custom products, and the cheap-and-cheerful, fashion-based, requirement-free technology-led (etc) product market. It was all very well to say you needed an Integrated Project Team to develop mass-market products; but how did one integrate engineers and marketing people if they despised each other?

On Friday Steve Lipner, the Director of Security Assurance at Microsoft, spoke in a very secure way (not giving anything away) on secure systems. He illustrated from his experience since the 1960's how physical locks on doors had given way to multi-user timesharing systems in the 1970's and 1980's with access controls and internal controls on data. Then PCs came along and stepped back a generation to the 1960's with simple operating systems and only physical locks for security. He didn't blame any particular hardware or software companies for this retrograde step (or name any products like IBM PC-DOS, MS-DOS for that matter). The 1988 Worm gave a brief foretaste of what was to come. Then in the 1990's the Internet exploded on to the scene, and the world of firewalls, hackers, Internet time, "good enough" code, security features, and spam came to everybody's attention. Only now in the 2000's does it finally seem likely that people are really ready for 1980's-style multi-level security (it flopped first time round).

Alistair Mavin and Neil Maiden gave a fascinating paper on their experiences with generating and

walking through scenarios. The CREWS-SAVRE tool took some use cases (and, yes, writing good ones suitable for input was hard) and applied both generic and domain-specific exception classes to multiply up a list of questions: could a human make a mistake here? could a machine fail there? could events happen in the wrong order over here? A specific hypothesis was that being more specific to the domain would help by generating more requirements, at least of certain types like functional, performance, usability. Strangely, in the Air Traffic Control domain it didn't seem to be so; generic exceptions worked well, though perhaps the specific ones were being grouped by the controllers under the generic headings in their replies. 71% of the discovered requirements were functional; the rest were on usability and reliability. It occurred to me (and others) that you could do even better by adding some generic but NFR-specific questions such as 'Could there be a security risk here?', 'Could doing this make the system harder to maintain?' etc. Perhaps quite a short list of generic exception classes could be universally applicable, making elicitation a lot more systematic: an exciting prospect.

Peri Loucopoulos (UMIST) showed how simulation models could be used to demonstrate the effects of different requirements for venue operations at the Olympic Games, effectively playing out different scenarios. For instance, if you have 20,000 people in a stadium and you only have 2 hot-dog stands or 4 turnstile entrances, how long will the queues be? You'll quickly discover the trade-offs between the space required for the installations and their operating cost, and the acceptable performance targets. In Mark Bergman and Gloria Mark's terms, you are shifting the balance of power from political to technical considerations, as everybody can see that these things matter.

The banquet was held (on the Wednesday!) in the dreamy atmosphere of the Monterey Aquarium. A sumptuous dinner with sushi, salads and roast meats was provided in dim bluish light amidst the illuminated fish tanks. Giant tuna and turtles swished silently past our tables; a kelp forest three storeys high (just how thick is that glass?!) swayed gently; silver sardines rushed round and round in a tall cylinder. It was a relaxed and happy RE'03 conference.

© Ian Alexander 2003

Tom Gilb presents 'Planguage' a planning language for requirements and more

RESG Event, 17th September 2003, Imperial College, London.

On the afternoon of September 17th at Imperial College, Tom Gilb took centre stage in a room packed with enthusiastic attendees expectantly awaiting the "pearls of wisdom" that were bound to drop from the lips of one of the world's expert in the area of software

engineering. It is gratifying to see the increasing interest there is in the requirements area, if the level of attendance to the various events organised by the RESG is anything to judge by.

There is no doubt that whenever one has the opportunity to sit in a talk given by a person as experienced as Tom Gilb, it is easy to pick up useful information. Nevertheless, I feel slightly disappointed by the fact that, regardless of the useful bits I took away with me at the end of the afternoon, the content of the workshop itself fell far shorter than I expected in view of the claims made both in the workshop's title and abstract available at the RESG website.

The first immediate point that springs to mind is that there was no set of slides for the workshop as such. It seems that both Tom and Kai Gilb had a busy programme in London during the following days, therefore we as an audience had to make do with snippets of the slides that had been prepared for the other events they were carrying out later in the week.

After almost fifteen minutes listening to Tom Gilb's references to the many publications he has generated throughout his working life, we were finally introduced to the standard number of reasons why an organisation should invest more time and effort in the requirements phase of its development process.

Once he and the audience had apparently warmed to one another, he proceeded to stress the importance of having a clear understanding of the stakeholders because they are the main source of a product's requirements. He also showed the importance of getting the stakeholders to think in terms of measurement, because in this way it is possible to quantify the quality requirements making the subsequent development phases easier to carry out. Furthermore, if it is possible to measure, it becomes straightforward to validate, and to determine fairly early in the development process the chances of success throughout. Another element of the requirements specification process discussed was requirements templates. Good templates provide an excellent means to achieve good requirements because they provide the means to achieve the first step in the route to achieving the necessary level of clarity, testability, etc that are held desirable in each and everyone of a system's requirements. Tom Gilb completed his discussion on scales of measurement and quality requirements by describing a procedure on how to compose a scale of 'measure' that is part of his Planguage.

As we were running out of time for the first part of the session, the discussion on requirements, qualities, measures, and templates was brought to an end by the speaker emphasizing the importance of adopting an evolutionary approach for the requirements specification phase.

One of the highlights of the second half of the workshop, was a brief description of the requirement

structure chart, a hierarchical diagram that shows the various elements that are usually associated to a requirement, performance constraints, resource constraints, design constraints, etc.

A second interesting development were the ideas put forward by Kai Gilb who took centre stage for approximately ten minutes of the second half. I had the feeling that he was trying to convince us that we were spending too much time and effort specifying the functional requirements of a system when that effort could be better spent in taking more care on the other aspects of a system. He argued that there was no such a thing as a new "function", e.g. the "edit" function of a text editor. Civilisation has moved forward on the manner and speed that the functions can be implemented and perform but, if we look closely at the "basics" the set of functions that many systems carry out these days has always existed. It is an astonishing claim when you consider the many functions, e.g. remote login, that are only feasible thank to the advances of technology. Of course, it is always possible that Kai Gilb's philosophical view of the term "function" allows him to make this kind of claim. Whether this is the case or not, I fear it does not make it a strong enough argument to undervalue the importance of specifying a system's functions as he seemed to be doing.

In the last part of the session Tom Gilb delved more into process issues. He described how the processes support the generation of the resources and the type of performance that is expected of a system's function. He summarised the relation between processes, functions, performance and resources in a very neat table.

Overall, an afternoon of some "useful" guidelines on how to proceed and what things to look for when you are involved with the requirements phase of a product development process, but not much on the 'Planguage', the planning language for requirements.

© Elena Pérez-Miñana

Creativity, the Path to Innovative Requirements

RESG Event, 9th October 2003, Imperial College, London.

Neil Maiden and Suzanne Robertson described the techniques they had perfected to encourage creative thinking in the context of different organisations. Things they've done have included getting in a fusion chef to show how he uses creativity in his work, and then putting some kind of pressure on the workshop participants to think outside their usual boxes.

Creativity can be exploratory (delivering knowledge of analogical domains), combinatorial (delivering guidelines for and constraints on combining ideas in novel ways), or transformational (delivering

knowledge about possible solution spaces, and to explore new ideas by working backwards from possible solutions).

The original CORA-2 workshops had ‘gone like a rocket’ — the facilitators just ‘lit the fuse’ and a huge pile of candidate requirements emerged, but not much structure. So some lessons were learnt.

- One was that you needed divergence — but then convergence.
- Another was that it was on the second day that people’s creative juices really started to flow — so Neil and Suzanne have vowed never to run another one-day creativity workshop again!
- Another was that analogies stimulate some people but not others — analogical reasoning isn’t for everyone. For example, if you have a scheduling problem at work, maybe getting in someone to talk about railway scheduling will help.
- Yet another was that creativity means generating useful ideas, not necessarily radical ones — engineering is all about creating new products that work, rather than wacky ones. Workshops help to ‘surface’ ideas that start useful trains of thought.

Their repeated creativity workshops in air traffic control, using the RESCUE process, has become more effective as the techniques have been refined. A key technique is to identify constraints, and then to think out how they can be removed. In one air traffic control workshop, 80% of what people initially thought were constraints turned out not to be!

In a class exercise, everybody in the RESG meeting (sitting at group tables) was asked to identify constraints on having a private plane. Some people set to writing post-it notes on their own; others immediately started chatting and thinking up ever more outrageous ideas. Then they immediately had to work out how to remove their constraints. Someone said they’d just become a flying instructor. Others wanted to create self-cleaning or self-maintaining planes. Had people genuinely come up with insights from the brief exercise? Mostly not (this would correspond to the early discovery phase of a real workshop), but the flying instructor idea is certainly a practical one. Even wacky ideas should not be discarded out of hand — they may help to trigger other more practical things.

To set up a workshop, you need to assemble a suitable team of experts — people who know the domain, people to facilitate and scribe, people from other areas to stimulate through analogy. Usually after 2 days of workshops, the room looks as if a bomb has gone off in it — encourage that. Mess is good. An environment that doesn’t look like work is part of the method. An environment of playfulness is essential. Plan what food to bring; what music to play; what materials to use.

For each ‘expert’ session, you also need a time when the expert is available to be chatted to and answer questions of any kind. It seems to be almost impossible to write decent use cases until you have had such a workshop. It has to be totally respectable to change use cases! Expert roles for exploratory creativity can include cartoonists, fashion designers, musicians. For combinatorial: fusion chefs, DJs, cocktail bartenders — who all mix and match ideas. For transformational: science writers, graphic artists, who make difficult things understandable and visualisable. These people have explicit and tacit rules that guide their work. Such rules — if suitably transformed — can be valuable in engineering.

For the next exercise, David Bush explained what it was like to be a bicycle courier. This of course was an analogy to having a personal plane to beat the traffic jams. You had to have a waterproof bag; a strong bike with sturdy tyres; a map; several spare inner tubes; a deodorant spray; a phone, pager and/or radio; a strong lock. Safety was critical — keep away from car doors; watch out on the pavements for pedestrians who stop suddenly. Insurance too.

Neil explained that analogies were relational, not about attributes: A is to B as C is to D: literal similarities as well as analogies. You could have mappings between actions, agents, objects, requirements, design features, constraints, and so on. People called out a stream of biker things that had more or less plausible aircraft analogies, and then worked in their table-groups to explore analogies in more detail. One group suggested a wearable flying-suit (in place of a plane). Another had a powered parachute. A third could travel on water or road as well, as a swiss army knife can take on many roles. The avionics could have a removable display to defeat theft. The cycle bell could be equivalent to a radar transponder or a flashing lamp to warn off passing aircraft. There was quite a bit of laughter.

In a real RESCUE-process workshop, attention would now move to storyboarding, collages, sketches, anything to integrate the ideas. RESCUE works around use cases; the rule is to link all workshop outcomes to use cases so that all the ideas can be embedded in the storyboards. Storyboards are drawn on pre-formatted A0-size sheets, allowing for up to 16 story-frames per sheet. (The formatting really helps: plain paper isn’t nearly as good.) Stories don’t have to be complete and consistent: just something for the team to hang their ideas on, and afterwards to fill in the details to make well-formed use cases.

People in the workshop aren’t aware of it, but an enormous amount of planning goes into each one. Suzanne and her husband (James) developed one plan on a café’s paper tablecloth; brainstorming can use mind-maps or other such devices to help create robust and thorough workshop plans. Timing has to be quite precise: again, schedules don’t have to be made explicit to participants. It helps to list the deliverables

of each time-slot down the right hand side of the schedule. The breaks including lunch are parts of the workshop: people get cards with little prompts on them — and they get prizes for the best suggestions. Otherwise people can switch off in the breaks and lose momentum. Similarly for the overnight break, people get some simple homework so the workshop isn't limited by the four walls of the room.

Ideas then need to be solidified. One way is to make cards to document ideas, their value to the current product release, to future releases, to your grandchildren. There's a slot for 'Concrete Idea', and for Questions and Concerns. To get people to be creative, you have to be ultra-planned yourself.

© Ian Alexander 2003

RE-Papers

Innovating Requirements for Process Control Systems: Ideas Based on Recent Events of RESG

*Islam A. EL-Maddah
King's College London*

Recently, the RESG organised a workshop by Suzanne Robertson and Neil Maiden on 8th October 2003 "Creativity, the Path to Innovative Requirements", and a seminar by Tom Gilb on 17th September 2003 on "Planguage: a planning language for requirements and more" (see *Re-Readings* in this issue - Ed.). The two events addressed the requirements analysis stage from different views, highlighted strong points, agreed about the main points, and chose different approaches to tackling other aspects. After I managed to bring together their ideas, the feedback from the participants, and my recent work on requirements analysis of process control systems, I decide to put down these ideas and share it with the RESG members.

After listening to the creativity workshop, I asked myself why systems engineers and system analysts are reluctant to write down new requirements. Reducing this reluctance might lead to better requirements and leave space for considering new design. This is a particularly interesting question in applications like process control systems, where the existing physical components constitute the major part of the system.

But how can we encourage and prepare the systems engineer to ask for better requirements? The problem usually has some Psychological reasons, which have to be targeted in order to free the thinking style of the systems engineers from the way that has enslaved them. In the following paragraphs, I will introduce the problem and suggest the attribute set for each requirement to make the enhancement stage easier. I will then provide a brief introduction to the related psychological issues. Finally, the proposed directions will be illustrated.

The requirements stage is considered the most crucial stage to define the success of software applications [Nuseibeh and Easterbrook 2000]. This success depends on how well the software applications satisfy the user needs [Zave and Jackson 97]. Although in many cases, the requirements derived from the

stakeholders are satisfactory, they can be enhanced because they are not as demanding as they could be. Apart from application feasibility, the concept of acquiring the agreement and acceptance of the client should be extended to requiring better systems [Robertson 02], where new requirements can be added or the quality of initial requirements can be increased. Claiming that the enhanced requirements can lead to infeasible applications regarding the cost or effort can be defended by carrying out a feasibility study, but with better illumination, after exploring the new possible requirements. In the field of process control systems, the innovation may be obstructed by the physical constraints of the system: its local environment and plug-in components. In addition, the systems engineers may be reluctant to accept innovative requirements that demand new components rather than using the existing ones. The systematic style of thinking developed by engineers can also stop them from being creative and enhancing the requirements. Therefore we should pay more attention to inspire the systems engineer to invent better requirements regarding the interactions between the different components of the systems.

Before we start our argument about the suggested directions to create new requirements, it is important to consider the knowledge representation for each single requirement. We agree with Gilb [Gilb 03] that we should pay more respect to each requirement and never dumb them down within requirements documents as bullet points. Requirements should have a definition of their situation/local scope, such as during one phase of the production process, or a sub-system of the application. Each requirement addresses at least one aspect, such as safety, security, understandability, feasibility, running cost, or operation. Each requirement should have an attribute indicating the stakeholder who requests it. This can be easily noted in large systems where there are many stakeholders Gilb in [Gilb 03] has suggested that the requirements should be expressed using who, what, and where modifiers. Another important attribute to consider is the degree to which the requirement is important to fulfil. Requirements vary from the client's perspective from being essential to optional features. When new requirements are invented, it would be helpful to annotate each of them according to their importance for (for example) energy-consumption reduction,

shortening service time or ensuring extra safe conditions. This can provide a higher chance of deriving a flexible implementation that fulfils the most important requirements and offers some of the optional ones. Robertson and Maiden suggested that some of the optional requirements could be marked as recommended in "future versions".

This suggests that we modify each requirement to add "what aspect(s)", "where and when", "details", "names", "identifier" and "annotation" [Maiden and Robertson 03]. Considering these attributes as dimensions in the requirements space, the requirements can be considered as points. Thus, the attributes' values can derive structuring or grouping of the requirements. In addition, these modifiers can work as a base to deduce new ones around their neighbourhood of the existing requirements. Having prepared the requirements' search space, the problem now turns to how to explore the search space and suggest places to dig in for new requirements or how to invent new ones based on requirements of similar applications.

The activity of creating new requirements combines features of problem solving and creativity. In problem solving, humans use diverging and converging styles of thinking [Hayes 93]. Artists usually think divergently about related thoughts, significance, and abstraction of similar problems. Scientists attempt to converge the ideas to solve the problem, and use analogy and mapping to solve the problem in different domains. Thus, the two approaches have to be brought together to guide the systems engineer to invent new requirements and then to refine them and ensure their feasibility cost- and effort-wise. Problem solving sometimes faces obstacles resulting from the way humans are used to think, such as Functional Fixedness (difficulty of using the same object differently) [Glucksberg 62] and Einstellung (being limited by the problem boundaries) [Hayes 93]. These problems need to be targeted in order to increase the chances of creating new requirements. Creativity needs to be analysed and supported to understand how humans are enabled to create ideas, artwork and inventions. In [Ghiselin 52], the life cycle of developing creative ideas or artwork consists of three main phases. These are: familiarization (where the human needs some time to know about the problem domain and understand the basic information), incubation (when the human seems to be inactive while their unconscious minds prepare the creativity), and activity (this is the final phase where the human creates the idea or the artwork). Thus, by encouraging systems engineers who are familiar with the application and availing a suitable time and environment for them, they can create new effective requirements. We could support the incubation phase of creativity by Lateral thinking [de Bono 77] to enable the engineers to diverge their thoughts around the problem and the application, and not to restrict them. Also, with effective plans for Brain Storming,

we could set an environment to produce pool of thoughts in one stage without any limit or constraint. Afterwards, we can arrange and evaluate the discovered thoughts.

We propose three directions to enhance and modify the requirements based on inspiring the systems engineer to enable the lateral thinking to diverge in order to create better requirements. The outcomes of the different directions should be brought together and evaluated to produce enhanced and realistic requirements [Maiden and Robertson 03]. Through the following proposed directions, we explain examples from a simple production cell case study [Lewerentz and Lindner 95]. The production cell receives metal blanks, stamps them and then delivers the processed ones to the collection area for the next stage. To achieve these tasks, the production cell consists of a feed belt, rotary table, robot, press and deposit belt. In [Lewerentz and Lindner 95], the stated requirements were provided to operate the cell normally and ensure that every metal blank arriving to the cell via the feed belt will be stamped and delivered to the deposit belt (Liveness aspect). Safety conditions must be maintained at all times such as avoiding collisions between the different machines and avoiding having two blank metals on the table tray or inside the press at the same time. It was required to achieve this effectively in a way to increase the system throughput. In addition, the requirements specification should be easily adapted for similar production cells. We arranged the requirements of this case study in a goal-model using the GOPCSD tool [El-Maddah and Maibaum 03]. Then, we checked the requirements for completeness and consistency. Thus, we ensured requirements correctness and user-agreement. Although, we produced new requirements/refined requirements, which were not stated in [Lewerentz and Lindner 95], we did not consider the enhancement, creation and innovation of requirements as a separate preliminary stage. However, we can pull the different strings together as follows:

Direction 1. This direction attempts to free the way of the systems engineer's thinking by partially removing the physical restrictions and constraints of the system components. Provided that the components have to be used as they are, we aim at freeing the interaction between them from the conjunctive restrictions from both sides. We free one component at a time and consider the other related components' restrictions and constraints to inspire new interaction requirements. Each component in turn will be replaced by a more flexible component, or by an imaginary human. For example, the robot can be replaced by a human performing the same tasks assigned to the robot. We start by writing down what the human can do in these tasks already assigned to the robot; then, we can extend these situations by imaging that the human can think of better ways to achieve the tasks. We will pay particular attention to tasks concern interacting with other components, as well as recording why the human

does these tasks (to fulfil which aspects). This can free our thinking from the restrictions we imposed on the robot and think of it as a human. Thus, we “deceive” ourselves to get rid of the functional fixedness and Einstellung. Similarly, we can consider the other components of the production cell.

Direction 2. Unlike the first direction, we may look for an entire system that has similar themes of requirements to the intended system, but with higher flexibility. This direction has been considered in light of the idea of “Experts from Different Fields” as in [Maiden and Robertson 03]. For instance, one can draw some similarities between the production cell and a basic bakery, where the loaves of bread are prepared in batches and handled to the oven then picked up to the display area. This direction projects the bakery operations/situations/entities onto the production cell problem. It helps the search for similar characteristics in the production cell. Because the bakery system is simple and flexible, many thoughts, concepts and aspects can be considered to enrich the production cell with them. It is helpful to realise that we do not look for the similarities blindly. Instead, after we have decided that the bakery is a similar system to the production cell, we list many situations within the bakery system and attempt to find or “invent” similar situations in the production cell. For example, the bakery requires that we do not burn the loaves, and that we check whether the loaves are ready from time to time. This can suggest modifying the press component and encouraging the control program to implement counters to estimate when the press will finish stamping the blank metal.

Direction 3. Arranging the ideas in the human mind makes them easier to be access as well as easier to create new ideas. Similarly, considering each requirement as an idea enables us to build a requirement net. Arranging requirements in trees as goal-models [Van Lamsweerde et al. 91] or in nets or graphs seems to provide better understanding and better elicitation chances. For example, we can relate the different requirements using: refines, inherits, depends_on, and conflicts_with relationships. These relationships should be selected to provide guidance for the converging as well as the diverging (eliciting new requirements) stages [Maiden and Robertson 03]. We should not think of the different alternatives but different views, which can be combined together; i.e. we can have a rich net that offers refinement and dependency relationships. The requirements net, graph and/or tree can provide guidance for extending the requirements through creating new ones. It aids reasoning about why and how in goal-models [van Lamsweerde et al. 91], helps consider how else a requirement might be satisfied [Mylopoulos and Castro 00], and may help in voting stages [Maiden and Robertson 03].

As illustrated earlier, the converging stage has the responsibility of bringing together the new

requirements and filtering out the infeasible ones. Its importance is particularly apparent after creating and inventing a large number of new requirements. The different modifiers that each requirement possesses can guide the selection and evaluation of the requirements. The requirements net, graph and/or tree may help in voting stages [Maiden and Robertson 03].

To draw the main conclusions, some directions have been suggested for creating new requirements within process control systems, where there is a strong preference to reuse the hardware and physical components. This left us with a little chance for enhancing the requirements of the interaction between the components or of the high-level requirements of the entire application. We based our proposed directions on the way the systems engineer usually solves problems. Innovating new requirements can create a need to demand better systems as well as to leave the door open for dreaming of better future systems.

References

- [de Bono 77] de Bono, E. (1977) *Lateral Thinking*, Harmondsworth: Penguin.
- [Davis 03] Alan M. Davis (1993) *Software Requirements: Objects, Functions and States*, Prentice Hall PTR
- [El-Maddah and Maibaum 03] El-Maddah, I. A. and Maibaum, T. S. (2003) *Goal-Oriented Requirements Analysis of Process Control Systems*, Proc. First ACM & IEEE International Conference on Formal Methods and Models for Codesign (MemoCode), France
- [Gilb 03] Gilb, T. (2003) *Competitive Engineering, a Handbook for Systems and Software Engineering Management using Planguage*, Addison-Wesley
- [Ghiselin 52] Ghiselin, B. (1952) *The Creative Process*, Berkeley: University of California Press.
- [Glucksberg 62] Glucksberg, S. (1962) The influence of strength of drive on functional fixedness and perceptual recognition, *Journal of Experimental Psychology*, 63, 36-41
- [Hayes 93] Hayes, N. (1993) *A First Course in Psychology*, third edition ISBN 0174481810, Thomas Nelson Sons Limited
- [van Lamsweerde et al. 91] van Lamsweerde, A., A. Dardenne, B. Delcourt, and F. Dubisy (1991) “The KAOS Project: Knowledge acquisition in automated specifications of software”, proceeding AAAI Spring Symposium series, Track: “Design of composite systems”, Stanford University, March 1991, pp 59-62.
- [Lewerentz and Lindner 95] Lewerentz, C. and Lindner, T. (1995) *Case Study: Production Cell: a Comparative Study in Formal Software Development*, Lecture Notes in Computer Science No. 891, Springer-verlag, 1995
- [Maiden and Robertson 03] Maiden, N. and Robertson S. (2003) “Creativity, the Path to Innovative Requirements”, RESG One-day Workshop, Imperial College

[Mylopoulos and Castro 2000] Mylopoulos, J. and Castro, J. (2000) Tropos: A Framework for Requirements-Driven Software Development In S. Brinkkemper and A. Solvberg (eds.), Information Systems Engineering: State of the Art and Research Themes, Lecture Notes in Computer Science, Springer-Verlag, p. 261-273, June 2000

[Nuseibeh and Easterbrook 2000] Nuseibeh, B. and Easterbrook, S. (2000) Requirements Engineering: a Roadmap, Future of Software Engineering, Limerick, Ireland, 2000

[Robertson 02] Robertson, J. (2002) Eureka! why analysts should invent requirements, IEEE Software, 19(4) 2002

[Zave and Jackson 97] Zave, P. and Jackson M. (1997) Four Dark Corners of Requirements Engineering, ACM Trans. Software Engineering and Methodology, 6(1).

The Orthodox Barber or, What Tools Shouldn't Be Used For

Ian Alexander
Independent Consultant

A little while ago, I dreamed I was back in the Edwardian-age London of 1909, discussing progress in our respective fields with a barber. I mused over an essay by G.K. Chesterton about the odd new fashion for gadgets at that time.

I had been invited to some At Home to meet the Colonial Premiers, and lest I should be mistaken for some partly reformed bush-ranger out of the interior of Australia I went into a shop to get shaved. While I was undergoing the torture the man said to me –

"There seems to be a lot in the papers about this new shaving, sir. It seems you can shave yourself with anything – with a stick or a stone or a pole or a poker" (here I began for the first time to detect a sarcastic intonation) "or a shovel or a–"

Here he hesitated for a word, and I, although I knew nothing about the matter, helped him out with suggestions in the same rhetorical vein.

"Or a button-hook," I said, "or a blunderbuss or a battering-ram or a piston-rod–"

He resumed, refreshed with this assistance, "Or a curtain-rod or a candle-stick or a–"

"Cow-catcher," I suggested eagerly, and we continued in this ecstatic duet for some time. Then I asked him what it was all about, and he told me. He explained the thing eloquently and at length.

"The funny part of it is," he said, "that the thing isn't new at all. It's been talked about ever since

I was a boy, and long before. There was always a notion that the razor might be done without somehow. But none of those schemes ever came to anything; and I don't believe myself that this will."

"Why, as to that," I said, rising slowly from the chair and trying to put on my coat inside out, "I don't know how it may be in the case of you and your new shaving."

G. K. Chesterton, *The Orthodox Barber*, in *Tremendous Trifles*

But did you know, I said to the barber, there are people today who believe you can describe a complicated business like banking or telephony just by drawing some stick-men and some ovals on a chart, and joining them together with little coloured lines without so much as a word of explanation?

Or that they imagine that they can prepare the specifications of some complex new device entirely in the form of logical symbols or engineering diagrams, hundreds and hundreds of pages at a time, and ask the poor customer to form a contract on the basis of his meagre understanding of what the analysts and engineers supposed was a logical representation of what the customer himself had said that he wanted?

Can you imagine, I inquired of the barber, there are salesmen who live entirely by selling tools on the pretext that these will instantly solve all the problems of engineering a new product?

And that they find a ready market, because development projects are so busy with complying with quality standards and guidelines, with going to conferences and meetings to find out about all the latest labour-saving gadgets, with keeping up with fashions and trends, with submitting to audits and preparing progress reports, with publicity and marketing and public relations and brochures, that the overworked engineers cannot spare the time to evaluate the products before they buy, and that although the tools often cost many thousands of pounds and are not meant for fools, the engineers can hardly take a moment to learn how to use them, so that often the newly-purchased goods languish and gather dust on a shelf, or else the engineers make such a hash of the work with the unfamiliar tools, that they would have been better off without them?

Have you heard, I whispered in the barber's ear, that well-known engineers speak to packed conference halls about doing away with tiresome product plans and specifications, preferring to save time and money by rushing into development unaware of either the risks involved, or of the real needs of their clients?

The barber wiped the last traces of foam from his dreadfully sharp old-fashioned cut-throat razor, and replied that he had not heard of these things, but that he expected it was the way nowadays.

As I awoke, I thought I glimpsed the plump cheerful figure of G. K. Chesterton, winking at me through his monocle from the door.

RE-Publications

All reviews by Ian Alexander

Classic Book: Designing for People

Henry Dreyfuss (1904-1972)
Simon & Schuster, 1955
Reprinted by Viking, 1974
Reprinted by Allworth Press, 2003
ISBN 1-58115-312-0

To me, Leonardo da Vinci is a symbol; artist, scientist, engineer, rebel -- and the greatest of all industrial designers -- he boldly put his dreams on paper. I find it fascinating to speculate about what Leonardo's thoughts would be if he were alive today. (page 19)

Dreyfuss' industrial autobiography (could anyone write such a thing today?) is not just a classic but a beautiful book in every sense. The echoes between Dreyfuss, with his Joe and Josephine figures (see illustration above) and Leonardo da Vinci's Man inscribed heroically in a circle are perhaps a little too self-conscious, but Dreyfuss really was a bit of a Renaissance Man, at once artist, designer, founder of the discipline of Human Factors, author, publicist, and businessman.

Time is visible in many ways in this book, not least in its physical format: a good-sized cloth-bound hardback, with an elegant typography on slightly yellowing paper, printed in two colours -- black and a warm reddish-brown for the illustrations and headings. The language and cultural preconceptions, too, bespeak an earlier and more gracious age. For Dreyfuss may have been an American, but he was well-travelled and loved France (among other things). When he writes

'Frequently I hear people say, with or without intended snobbery, that Americans are so engrossed in materialistic values that culture is lost on them.' (page 82)

he is plainly speaking in defence not just of his fellow-citizens but of himself. And of course he goes on to point out the merits of 'well-designed, mass-produced goods', even claiming them to constitute a new art form. His Hoover vacuum cleaner was the first to be designed to do what housewives needed - to go under furniture, to be light to carry, to stow the cable neatly, and it still looks serviceable today.

As Sven Birkerts writes in *The Gutenberg Elegies*, just the look of an old book gives us a reminder of the length of history, and the time people have thought about design:

Say what you like about books, they not only mark the backward trail, but they also encode this sense of obstacle, of otherness. The look of the printed page changes as we regress in time; under the orthographic changes are the changes in the language itself. Old-style textual research may feel like an unnecessarily slow burrowing, but it is itself an instruction (Birkerts, page 137)

There are plenty of reminders of time in Dreyfuss' book. Not least are the reminders of just how recent are the gadgets we take for granted: "Today the convenience of telephones is established". Requirements engineering was in those days called design, or at least in no way differentiated from it, but design was seen mainly as what happened on the surface, with 'engineering' going on underneath:

I suggested that a telephone's appearance should be developed from the inside out, not merely created as a mold into which the engineers would eventually squeeze the mechanism, and this would require collaboration with Bell technicians. My visitor disagreed, saying such collaboration would only limit a designer's artistic scope.

Several months later he returned ... (Dreyfuss, pages 102-3)

Dreyfuss exudes the confidence of a Benvenuto Cellini (an Italian renaissance silversmith who wrote a not very modest autobiography) -- he was evidently delighted with his own work, in a way which is engagingly simple and direct, but which would scarcely be acceptable today.

Yet he pioneered all manner of practices that we take for granted.

The cost of a model is more than compensated for by future savings. It not only presents an accurate picture of the product for the executives, but it also gives the toolmakers and production men an opportunity to criticize and to present manufacturing problems. Models of some products can be made for a few hundred dollars. ... A mock-up of a modern passenger airplane cabin may cost \$150,000 but it will be worth it, for it permits engineers and designers

to develop techniques of installation that would not otherwise be possible. Furthermore, sales executives can bring potential customers into a faithful, full-scale fuselage to see what it offers, long before production begins. It is far more effective to sit in a chair than judge its comfort by a picture of it. (page 62)

How many kinds of critical stakeholder did you notice being integrally involved in the multi-functional team's participative development life-cycle approach there? Dreyfuss writes without the grating management-speak jargon we see every day, but he understood exactly why it helped to consider what people needed, and how to ensure they understood what they would get. His emphasis on manufacturability -- as we now call it -- is also startlingly modern. There's even a chapter called 'The Importance of Testing'.

Dreyfuss is widely acknowledged to be a founder of modern industrial design and of human factors. On the evidence of this book, he might also be seen as a pioneer in participative design, in attention to stakeholder viewpoints from usability to production, and in the importance of continuity between specification and testing. He's also such fun to read that the search for an old hardcover edition is well worth while.

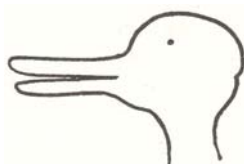
© Ian Alexander 2003

See also the Exhibit on Dreyfuss at the Cooper-Hewitt National Design Museum Smithsonian Institution (<http://ndm.si.edu/EXHIBITIONS/hd/ex6.html>)

Classic Book: Philosophical Investigations

Ludwig Wittgenstein (1889- 1951)
Blackwell 1953, Reprinted 2001
ISBN: 0631231277

This little book is a surprise in many ways. Not least, it is short and easy to read – at least, if you dip into it and read a page, you will have no difficulty seeing what Wittgenstein is talking about. The visual metaphor is intentional – Wittgenstein spends a lot of time discussing what we mean by seeing and understanding; what we can talk about publicly, and what is private and forever inside our heads. In fact, another surprise is that the book is illustrated, in one place even in colour! If you don't know what Wittgenstein's duck-rabbit is, perhaps you now will – showing is more powerful than telling (and there's a lesson in there for everyone involved in eliciting requirements).



And perhaps the book's title is less forbidding than the famous *Tractatus Logico-Philosophicus* – though he writes in the Preface: 'Four years ago I had occasion to re-read my first book (the *Tractatus*) and to explain its ideas to someone. It suddenly seemed to me that I should publish those old thoughts and the new ones together ...' In other words, this book is a commentary and reflection on all of his thinking. It was written in the final years of his life, and published posthumously. The *Tractatus* is a numbered list of short statements or aphorisms, which makes it quite hard to read in the ordinary sense – I don't know about you, but if I read a saying or a proverb, I have to stop and think, do I agree with this or not? What would it mean for me? Is it true? What implications does it have? and without this discussion written into the text, after a few aphorisms I have started so many hares I don't know where I'm going. But the *Investigations* presents ideas, gives examples, and discusses them, which is a lot more approachable; and since many of the ideas are from the *Tractatus*, this makes it a nice way in to that book as well.

It's a surprise too, that someone with a long german name should be so, well, English; so simple and direct; so clear; so interesting -- and so relevant to requirements. Normally, of course, one has only to mention something like *Weltanschauung* and everyone knows to switch off at once, their internal spam-filter cutting in to preserve an acceptable signal/noise ratio. So I will try to say in a few words why Wittgenstein should not be one of the long W-words among your 'trash' filters.

Writing requirements is about translating what people need but can't necessarily express, into a set of statements that developers can understand well enough to build something – a system – whose operation satisfies those people's needs. This task is hard not because we aren't clever enough at inventing gadgets and software (if anything we're too clever), but because the translation of needs into written meanings is far more than the clerical task (the word is Dijkstra's) of writing specifications.

Polanyi's maxim is 'We know more than we can tell', and this forms a convenient stepping-stone into Wittgenstein's thought, which ranges over questions of meaning and how we understand; of knowledge and consciousness; of language and logic; of sensory observation and communication; and more.

For example, how do we ensure that people on a project mean the same thing when we use a term like 'system reliability'? Obviously it matters, for instance if the contract that contains the requirements talks about measuring this quantity and rewarding the supplier accordingly. Wittgenstein says that

43. 'For a large class of cases .. in which we employ the word "meaning" it can be defined thus: the meaning of a word is its use in the language.'

And the meaning of a name is sometimes explained by pointing to its bearer.'

Therefore, we had better make sure that the language in use in our project is shared by all involved in it – for instance, we should make a dictionary, and we had better ensure that it corresponds closely to the language people already use; what is more, we must root out any other uses to avoid confusion. That might take a bit of doing? Well, yes; but the alternative is ambiguity and arguments over contracts. What was all that about the bearer of a name, then? We need to associate dictionary terms with things in the world – by what Jackson calls 'designations' – to form bridges that jump the gap between our formalised language and models, and the world that our stakeholders understand. Then if someone does not understand our requirements, we can point to the things we are talking about, and explain our meaning in terms of them.

Pointing perhaps doesn't sound very much like the subject of a *Philosophical Investigation*? But it is.

45. ... a name is not used with, but only explained by means of, the gesture of pointing.

Pointing, or ostension to give it its more general name, is a vital bridge between the world and what we and other people have inside their heads. How do we share meanings with other people? How do we know what other people see when there's a red letter-box in front of them? What if they're colour-blind, anyway? (The problem of 'qualia', or what we'd call attributes – like redness – of things in the world, is a deep one, and it doesn't go away.)

II xi. I look at an animal and am asked "What do you see?" I answer: "A rabbit". – I see a landscape; suddenly a rabbit runs past. I exclaim "A rabbit!"

Both things, both the report and the exclamation, are expressions of perception and of visual experience.

Well, what if I wasn't familiar with rabbits: what would I have said then? What would I have seen? (Wittgenstein may also be hinting here at the writings of W.V.O. Quine, who also discussed how people could share the name of an animal by looking and pointing at and naming it, and what sorts of misunderstandings and ambiguities could arise – but usually don't.)

Someone suddenly sees an appearance which he does not recognize. ... Is it correct to say he has a different visual experience from someone who knew the object at once?

As a birdwatcher, I can glance at a pond and see Pochard, Coot, Moorhen, Mallard, ... while a companion sees nothing special – if asked, he might say 'Ducks' (partly right, and partly wrong: swimming waterbirds would be more accurate). So, does someone with more knowledge and more specialised language see things differently? What happens if we try to gather requirements when we are talking a) to people less expert than ourselves; b) to people more expert in the domain (banking, aerospace, whatever) than ourselves? Either way, there is going to be a difference in meaning, if not indeed also of perception.

Perhaps these few hints may have helped to give some idea of what Wittgenstein was thinking about, and how fresh and immediate he is, and how important to our discipline. I think I'll give him the last word. He begins his *Philosophical Investigations* with a quotation (in Latin, but I'll give the English) which beautifully summarizes the subject:

1. When my elders named some object, and accordingly moved towards something, I saw this and I grasped that the thing was called by the sound they uttered when they meant to point it out [ostendere]. ... Thus, as I heard words repeatedly used in their proper places in various sentences, I gradually learnt to understand what objects they signified...

{Saint} Augustine {354-430 A.D.}, *Confessions*, I. 8 {397 A.D.}

These words, it seems to me, give us a particular picture of the essence of human language. It is this: the individual words in language name objects -- sentences are combinations of such names. In this picture of language we find the roots of the following idea: Every word has a meaning. This meaning is correlated with the word. It is the object for which the word stands.

© Ian Alexander 2003

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 at the RESG website:

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

Ian Alexander's archive of book reviews is available from:

<http://easyweb.easynet.co.uk/~iany/reviews/reviews.htm>

The requirement management place
<http://www.rmplace.org>

A good general resource for RE issues. Includes Alan Davis' Requirements Bibliography.

CREWS web site:

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

An interesting collection of 72 papers (!) and a description of an ESPRIT project on co-operative requirements engineering with scenarios.

Requirements Engineering, Student Newsletter:

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

IFIP Working Group 2.9 (Software Requirements Engineering):

http://www.cis.gsu.edu/~wrobinso/ifip2_9/

Requirements Engineering Journal (REJ):

<http://rej.co.umist.ac.uk/>

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

RE resource centre at UTS (Australia):

<http://research.it.uts.edu.au/re/>

Volere:

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

Mailing lists

RE-online (formerly SRE):

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

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

subscribe RE-online <your email address>

LINKAlert:

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

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

RE Acrostic

<i>Verifiable</i>	can the requirement be quantified and tested?
<i>Attainable</i>	can it be implemented? are there functional or cost constraints that make it unrealistic?
<i>Concise</i>	only one shall and no and's or or's;
<i>Unambiguous</i>	the words should be clear and precise – no fuzzy words with undefined meanings;
<i>Unique</i>	always identifiable with no duplications;
<i>Managed</i>	keep it updated throughout the project lifecycle, record and control the changes.

Remember that it is said that a universe was born out of nothingness.

Gordon Woods

RE-Creations

To contribute to RQ please send contributions to Pete Sawyer (sawyer@comp.lancs.ac.uk). Submissions must be in electronic form, preferably as plain ASCII text or rtf.

Deadline for next issue: 31st January 2004.

RE-Actors

The committee of RESG

Patron: Prof. Michael Jackson, Independent Consultant.
E-Mail: jackson@acm.org.

Chair: Prof. Bashar Nuseibeh, Computing Department, Faculty of Maths and Computing, The Open University, Walton Hall, Milton Keynes, MK7 6AA, UK. E-Mail: B.A.Nuseibeh@open.ac.uk.

Vice-Chair: Dr Alessandra Russo, Department of Computing, Imperial College, 180 Queen's Gate, London SW7 2BZ, UK. E-Mail: ar3@doc.ic.ac.uk

Treasurer: Prof. Neil Maiden, Centre for HCI Design, City University, Northampton Square, London EC1V OHB, UK. E-Mail: N.A.M.Maiden@city.ac.uk.

Secretary: David Bush, National Air Traffic Services, UK. E-Mail: David.Bush@nats.co.uk.

Membership secretaries:

Steve Armstrong, Computing Department, Faculty of Maths and Computing, The Open University, Walton Hall, Milton Keynes, MK7 6AA, UK. E-Mail: S.Armstrong@open.ac.uk.

Dr Juan Ramil, Computing Department, Faculty of Maths and Computing, The Open University, Walton Hall, Milton Keynes, MK7 6AA, UK. E-Mail: J.F.Ramil@open.ac.uk.

Newsletter editor: Dr Peter Sawyer, Lancaster University, Computing Department, Lancaster, LA1 4YR, UK. E-Mail: sawyer@comp.lancs.ac.uk.

Newsletter reporter: Ian Alexander, 17A Rothschild Road, Chiswick, London W4 5HS. E-Mail: iany@easynet.co.uk.

Publicity officer: Dr Sebastian Uchitel, Department of Computing, Imperial College, 180 Queen's Gate, London SW7 2BZ, UK. E-Mail: su2@doc.ic.ac.uk

Regional officer & Chair for the North of England: Kathy Maitland, University of Central England, Perry Bar Campus, Birmingham, B42 2SU. E-Mail: Kathleen.Maitland@uce.ac.uk.

Student Liaison Officer: Carina Alves, University College London, Department of Computer Science, Gower Street, London WC1E 6BT, UK. E-Mail: c.alves@cs.ucl.ac.uk

Industrial liaison:

Dr. Wolfgang Emmerich, University College London, Department of Computer Science, Gower Street, London WC1E 6BT, UK. E-Mail: W.Emmerich@cs.ucl.ac.uk.

Elena Pérez Miñana, Philips Digital Systems Lab. E-Mail: elena.perez-minana@philips.com.

Suzanne Robertson, Atlantic Systems Guild Ltd. 11 St. Mary's Terrace, London W2 1SU, E-Mail: suzanne@systemsguild.com

Gordon Woods, Independent Consultant, E-Mail: Gordon@cigitech.demon.co.uk

Requirements Engineering Journal

For 2004, *Springer-Verlag* are continuing to offer RESG members a substantial discount on subscriptions to the REJ. Members can subscribe for only £38 (print + online) or £27 (online only). See

www.springeronline.com.