Requirenautics Quarterly

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

http://www.resg.org.uk

RE-Locations

RE-Soundings

Editorial

The main RESG event since the last issue was undoubtedly R-Day. As anticipated, this attracted a large audience to hear and debate with many of the biggest names in RE. As you might expect, much of this issue is devoted to reports on R-Day from Ian Alexander, Carina Alves, Elena Pérez-Miñana and Gordon Woods. It's not often we deploy our full arsenal of roving reporters at the same event but, with parallel sessions, one person can't cover more than a small part of the event. Even with multiple reporters, we still can't report everything that happened. Nevertheless, I think you'll get a good flavour of what went on if you missed R-Day, and maybe discover things you missed even if you were there. What you won't get a flavour of is the frenetic activity behind the scenes by members of the RESG committee (RQ editor excepted!) to make sure it all went smoothly and didn't bankrupt us.

We also have our usual complement of articles and book reviews and a letter section (RE-Sponses). Letters are flooding in at the rate of almost one per year now. We have lots of spare capacity here in the bustling RQ editorial offices, so please don't be too shy to send us more letters, articles, news and reviews.

Finally, as you'll see from the chairman's message, there's an end-of-an-era atmosphere here at present. Bashar has chosen the RESG's 10th anniversary to step down, having led the group as chairman since the RESG's inception. Characteristically, he bows out by praising the rest of the committee while modestly claiming no glory for himself. Do not be fooled. We will really miss Bashar at the helm. However, we won't let him escape entirely since he plans to retain his links with the group and has promised a valedictory article for RQ in due course.

Pete Sawyer
Computing Department, Lancaster University.

Chairman's message

In the summer of 1994, Neil Maiden and I were invited to attend a BCS Technical Board meeting in London, to justify and answer questions about our proposal to form a new Specialist Group of the BCS on Requirements Engineering. The Board was very supportive of our proposal, recognising the importance of requirements engineering in the development of software intensive systems, and the growing community of researchers and practitioners interested in developing the state-of-the-art and the state-of the-practice. The new RESG held its inaugural meeting on "Requirements Traceability" on 19th October 2004, and has been holding regular meetings every two or three months ever since.

It has been very exciting to watch the RESG grow over the last ten years. The group has become the focal point of the UK RE community, and a valuable resource to many international researchers and practitioners. The group has always attracted good audiences to the wide range of events that it organises, and the list of distinguished speakers who have presented at RESG meetings reads like a Who's Who of requirements engineering! The group's finances are one of the healthiest of any specialist group of the BCS - allowing the executive committee to sponsor international conferences, award student prizes, and underwrite major events organised by the group. I am often told by RESG members that the group's newsletter, Requirenautics Quarterly (RQ), is keenly anticipated every quarter (or thereabouts!).

The achievements of the RESG, as it celebrates its tenth anniversary, are all the more remarkable in light of the fact that it is run entirely by the volunteers who serve on the committee. I have had the privilege of working with many such dedicated individuals over the last ten years, and it is these enjoyable collaborations with colleagues that I will miss most as I step down as Chairman at the next Annual General Meeting (AGM) of the group on 7th July 2004. I hope that many of you will attend the AGM, and perhaps consider standing for a committee position yourselves, to support the group as it moves into its second decade of existence. I am delighted to say that current RQ Editor, Pete Sawyer, has agreed to stand as Chair of the RESG, with Ian Alexander as the new RQ Editor and Kathy Maitland as the new Vice Chair. They all have my full support and best wishes. Pete is a former Head of Department of Computing at Lancaster University, and co-author of the tremendously successful book "Requirements Engineering: a good practice guide". He undoubtedly has the skills, the track record, and the expertise to guide the RESG into an even more successful future.

As for me, time to sign off. It's been fun. Thank you all.

Bashar Nuseibeh
The Open University

RE-Treats

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

Annual General Meeting to be followed by an audience with Anthony Finkelstein

Date: 7th July at 1.30pm
Location: Imperial College, London
Contact: Bashar Nuseibeh
(B.A.Nuseibeh@open.ac.uk)

The AGM is an opportunity for the membership to scrutinize and influence the running of the RESG. As usual, it will be followed by a technically-oriented event. This year, we have an audience with Professor Anthony Finkelstein. Anthony hardly needs any introduction but he is Professor of Software Systems Engineering at University College London, and one of the major international figures in software and systems engineering research. Unmissable.

RE for Defence Applications (provisional title)

Date: 27th October (time to be confirmed)
Location:Bristol (to be confirmed)
Contact: Gordon Woods
(Gordon@cigitech.demon.co.uk)

Birds of a Feather Meeting and RE Book Event

Date: 12th November (time to be confirmed)
Location: City University, London
Contact: Ian Alexander (iany@easynet.co.uk)

This evening event will be both an informal opportunity to meet other members of the RE community and to meet the authors of new books on RE. In particular, Ian Alexander and Neil Maiden will be on hand to launch their new book *Scenarios, Stories, Use Cases* (John Wiley) and James Robertson will be available to talk about his new book on *Requirements-Led Project Management* (Addison-Wesley).

RE-Calls

Recent Calls for Papers and Participation

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.

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

- 2nd International Workshop on Comparative Evaluation in Requirements Engineering (http://www.di.unipi.it/CERE04/)
- 2nd International Workshop on Requirements Engineering for COTS Components (http://www.lsi.upc.es/events/recots/)
- International Workshop on Requirements Engineering Patterns (http://rep04.desy.de/)
- RHAS `04 - International Workshop on Requirements for High Assurance Systems (http://www.sei.cmu.edu/community/rhas-workshop/)
- International Workshop in Service-oriented Requirements Engineering (http://www.elet.polimi.it/conferences/sore04/)
- International Workshop on Automotive Requirements Engineering (http://www.seto.nanzan-u.ac.jp/~amikio/NISE/AuRE/)
- International Workshop on Principles of Software Evolution (http://iwpse04.wakayama-u.ac.jp/)

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

Tom Gilb on Evo - Agile Evolutionary Project Management (BCS North London Branch event)

30th June 2004 at 5.30 pm, 2 – 3 Bloomsbury Square, London http://www.nlondon.bcs.org

25 Years of Communicating Sequential Processes (BCS-FACS event)

7th – 8th July 2004, London, UK, http://www.lsbu.ac.uk/menass/csp25/

Workshop on Intelligent Technologies for Software Engineering (WITSE '04)

20th - 25th September 2004, Linz, austria http://www-dse.doc.ic.ac.uk/Events/witse-04

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

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

International Journal on Software Engineering and Knowledge Engineering

Special issue on software traceability

Contacts: Dr. George Spanoudakis gespan@soi.city.ac.uk, Dr. Andrea Zisman a.zisman@soi.city.ac.uk

RE-Readings

Reviews of recent Requirements Engineering events.

Requirements Day

31st March 2004, Thistle City Barbican, Clerkenwell, London

The city of London awoke on Wednesday, March 31st to enjoy the pleasures of a perfect Spring day. One in which all the senses rebel at the thought that the cold, dark winter days are finally over and we must all prepare to make the most of the year's Spring and Summer months. This provided an ideal setting for the numerous activities that took place at R-Day'04, which was held at the Thistle City Barbican hotel.

The activity was organised by the RESG committee who, after a run of successful smaller events, decided to go for another full-scale requirements day conference, in the tradition of the RE-Day and the

CEIRE conferences that took place a few years ago. As with CEIRE, the committee strove to make the meeting low-cost so as to encourage both industrial practitioners and academics to attend.

We had the special good fortune to have the RE'04 program committee here in London through our treasurer Neil Maiden's chairmanship of that conference (the fine programme is to be held in Japan's most beautiful city, Kyoto, in September – do go!). The R-Day program was put together by a combination of inviting people, mostly academics, from the RE'04 program committee, and soliciting contributions from the community, mostly British industrial practitioners.

To make the event as rich and exciting as possible, we invited two distinguished speakers to give plenary addresses: Peter Hruschka (Atlantic Systems Guild) and Michael Jackson (Independent Consultant).

The rest of the programme was divided into three streams. In the morning there were three parallel tutorials: Julio Leite (PUC, Rio de Janeiro) on Ontology Development; Didar Zowghi (University of Technology, Sydney) on Requirements Evolution; and Suzanne Robertson (Atlantic Systems Guild) on building bridges between businesss and development.

In the afternoon, there was a packed schedule of talks – three times too many to go to, of course, but guaranteed to offer something for everyone:

Anthony Finkelstein (UCL) spoke on Requirements Stability: an Open Question.

Don Gause (STNY) spoke (highly amusingly and provocatively) on problem definition - problem solution: an exploration of CONTEXT in creating FORM.

Roel Wieringa (University of Twente) discussed Research Methodology in RE.

Alistair Sutcliffe (UMIST) discussed RE Research Methods - Theory and Practice.

Elena Perez-Miñana (Philips) spoke on Use Case-based Testing of Concurrent Systems.

Jeremy Dick (Telelogic) spoke on the Integration of Requirements and Design using UML 2.0.

Ian Gibson (Data Systems & Solutions) spoke on Applying Use Cases throughout the Development Lifecycle.

Paul Krause (University of Surrey) spoke on New Directions in Software Quality Control.

Simon Hutton (3SL) spoke on Requirements Metrics in a Project Management Context.

Søren Lauesen (IT University of Copenhagen) spoke on Integration Requirements in COTS Acquisition - the Customer's View.

Oscar Pastor (Valencia University) spoke on Linking Organizational Modelling with Scenario-Based Requirements Engineering.

Alistair Mavin (Praxis Critical Systems) spoke on Scenarios in Rail Rolling Stock with REVEAL: a Case Study.

Ian Alexander (Scenario Plus) spoke on Where did those actors come from? A new look at Stakeholder Analysis.

Happily, most of the slides for these talks are on the RESG website.

As if all that were not enough, there were interesting demonstrations in the exhibition from Adelard, Cediti, rcm2, Telelogic, and 3SL, and some of the latest books were on show at the John Wiley stall. Springer verlag also sponsored the day.

The Thistle City Barbican hotel kept us all well fed throughout what was a very busy day. Over a hundred people attended.

Keynote Speeches

Peter Hruschka "Agile Methodologies: The Great Debate"

Peter Hrushcka very aptly set the stage for the rest of the activities that were planned to take place during the day through the presentation of the main issues associated with "Agile Methods", describing how they compare to other software development methods, and the role that requirements play in their application. A strong supporter of an agile approach to the production of software, he proceeded to explain a structured tactic that can help in the determination of the most appropriate techniques to apply so as to deploy an effective requirements activity in the software development process.

Overall, he made a fairly impassioned appeal for shorter cycles of feedback from requirements to delivered versions of systems, i.e. more 'agility'. He argued that this did not mean extreme programming without requirements, but being willing to use whatever techniques were necessary: i.e. agile analysts had to be expert in brainstorming, field research, simulation modelling, running workshops, modelling goals, scenarios, processes, state transitions, whatever. There was an obvious implicit challenge to practitioners and researchers to develop both the skills and a methodology capable of "a rational approach to selecting best practices" (ideally in all circumstances: a tall order!).

For example, in order to decide on the most appropriate techniques to handle the elicitation and specification of a system's requirements, it is necessary to look at all the risk factors, i.e. constraints, that influence the development of software. Subsequently, he has produced a table in which each column represents a requirements technique, either for

elicitation or specification, and each row represents one of the constraints (which can be either human, organisational, or system related). The contents of a cell in the table describe the suitability of the requirements technique to handle the relevant constraint. Its contents have resulted from the vast experience Peter and his team has had in developing systems.

Even though Peter is a strong advocate of the agile approach to development, the procedure he described for the deployment of an effective requirements activity has a strongly structured flavour. It provides an orderly framework in which to set the different aspects that affect software development to better determine the type of requirements technique that are most appropriate for the particular situation. Initially, he suggests looking at each of the vertices that make up requirements independently, providing for each of them the factors that affect its successful deployment.

During question time, Jeremy Dick wondered how this could be applied to aircraft carriers, to which Peter replied that each sub-project within such a large development could be delivered in stages, increasing assurance that the whole would be on time and to budget. Subsequently, Alistair Sutcliffe pointed out that his suggestions did not seem to fit into an "agile" development framework which was something Hruschka did not disagree, stating that the general way of working he was advocating was of an agile nature.

Michael Jackson "Problem Frames: their use and misuse"

Michael Jackson gave an equally lively talk on problem frames, arguing that they offer a very low-cost way of picturing what you need to think about and what domain properties you need to describe to set the boundaries of the problem. This means that frames provide an effective tool to develop systems as they provide a guide, a way to think about a problem and its potential solution.

On the key question of how you (de)compose problem frames in order to structure the 'machine', in the first instance, it is necessary to decide on the level of granularity that wants to be achieved. He argued that subproblems and subdomains are projections, not parts: they are like the familiar Cyan, Magenta, Yellow, Black colour separations of a picture made for printing in pigment inks. Each separation is the whole picture, seen in a certain way. Decomposition has to be guided by recognition: you look ahead for familiar patterns (ah, that's a database issue, that's a real-time control problem, etc) rather than blindly decompose top-down (hoping you'll end up with something workable). The point is to identify the key concerns, like safety/reliability, breakage, or initialisation.

Michael gleefully told an initialisation horror-story, concerning the handheld 'plugger' device used by the US Army to steer a bomb onto its target. A soldier noticed a voltage low indication while guiding a bomb. He replaced the power pack. This reset the target indication to the plugger's own location, which is where the bomb landed. Michael clearly did not find the US Army's response – to re-emphasise operator training – particularly satisfying.

Michael suggested two principles: 1) Problem Sensitivity: does my development approach address the problem concerns? and 2) Developer Responsibility, which does not consist of "This product (a tube of glue) is not guaranteed because the conditions of use are beyond our control". Enough said.

Tutorials

Suzanne Robertson "Requirements: The Bridge Between Business and Development"

The tutorial conducted by Suzanne Robertson stressed the importance of the Requirements process in any organisation, in the first instance, through the identification of the "gaps" that exist between the different activities that are conducted in any product development process and those that enable the smooth operation of a business. This fact is evidenced by the various roles that a requirements activity can play in an organisation. Firstly, it allows us to build bridges by providing mechanisms to develop linguistic integrity within all the business and product development tasks that any successful organisation must carry out. Secondly, it enables its users to discover the origin of the business events because they provide the means for optimising the communication that must take place between the stakeholders of any successful business. Finally, the requirements provide the most effective means to engender collaboration amongst the business stakeholders.

Once the large audience that attended this tutorial was tuned to the speaker's strength of feeling on this matter, Suzanne proceeded to describe a mechanism designed and used by The Atlantic Systems Guild group. One that helps its customers to build the bridges needed to achieve a successful business. It all revolves around the idea of building a "knowledge model", a representation of all the elements that influence a business and the development activities it uses. One of the first steps in the construction of the model consists of the definition of the boundaries of the various environments within which an organisation and its production activities operate. Subsequently, it is necessary to start producing the knowledge that is relevant to the context of work by figuring out which are the classes of knowledge affecting the project or projects under development, including those that influence the persons involved. The initial model reflects the knowledge that is necessary to produce the requirements that must be satisfied by the business or the product under development. It will change and evolve through time until it reaches a "stability point".

The knowledge model will provide linguistic integrity because all the stakeholders that are part of the organisation will learn to express their requirements by using the concepts reflected in the model. It will help in the process of discovering the business events because it constitutes a pivotal point for all the activities conducted in the business. Given that all the stakeholders must provide their input to the model through its evaluation and update, it stimulates and supports collaborative development.

The requirements that must be specified in a particular organisation, be it business requirements, system requirements or product requirements, will make use of the knowledge reflected in the model by using the same terminology and providing input to the fine-tuning activity designed to improve it.

Overall, Suzanne provided a very convincing argument in support of developing and implementing a reliable requirements process in an organisation, we can only hope that businesses worldwide find it convincing.

Julio Leite "Ontology Development"

During this tutorial, Julio's aim was to present a type of ontology which has been made possible thanks to the technological developments associated with the world-wide web.

In order to provide the attendees with a clear understanding of the potential of the Language Extended Lexicon (LEL), he divided his presentation into four parts. In the first section he gave a detailed description of meaning of the term "ontology" and its usability from its conception in the area of Philosophy.

In the first instance it can be viewed as an area of expertise in itself; the subject of study being the categories of things that exist or may exist in some domain. It involves extracting all the terms that have a meaning in a certain domain, and the relations that exist between these terms which result in further meanings.

He stressed the importance this type of analysis has for any particular domain by listing the benefits that result through its development:

- It enables the possibility of sharing a common understanding of the structure of information among the entities that participate in the tasks associated in the domain.
- It fosters the reuse of a domain language.
- It helps to make explicit the domain assumptions that hold in the domain.
- It separates the domain knowledge from the operational knowledge.
- It facilitates the analysis of the domain knowledge.

In the particular case of the world-wide web, given the amount of information that can be managed through the media for a particular domain, it becomes particularly important to make explicit the meaning of the terms that become available. We must conduct a correct integration of the web resources that handle the objects in the domain. To this end, it is necessary to put together the properties of each term through the description of the various features and attributes associated with it.

During the second part of the tutorial, Julio discussed the current trends that are practiced to develop ontologies. He identified two main approaches:

- IA – Knowledge Engineering. In this case, what takes place is a domain mapping activity that results in the creation of large knowledge bases that represent the human knowledge that is mainly built by experts.
- The Semantic Web. This approach is the result of specific applications that are generally built by non-experts using a top-down strategy. It is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in co-operation.

The development of ontologies in the Semantic Web has a high risk component. This increases the importance of specifying a standard as the best means to ensure the information represented in it is understood by the humans and the software that manipulates that information.

The need for this standard was very aptly put by James Hendler, an authority in the subject of ontologies, who suggested that the Semantic Web will not evolve through the careful construction of net ontologies that result from the work of expert AI researchers. Instead, he envisioned a complex Web of semantics ruled by the sort of anarchy that rules the rest of the Web. Instead of a few, large, complex, consistent ontologies that great numbers of users share, he sees a great number of small ontological components consisting largely of pointers to each other.

In order to determine the best form and content for a standard in the Semantic Web, it is necessary to take into consideration the useful set of ontology languages, which are currently in use. To this end, during the third part of the tutorial, Julio described a number of them, i.e. XML, RDF, OWL, highlighting in each case, their strength and weaknesses.

During the final part of the tutorial, Julio described a new discipline that has been created to support ontology development, i.e. Ontology Engineering which corresponds to his current research area. He described his work, which corresponds to a lexicon-based ontology constructed by non-experts. It is being developed through the application of a mature process and is being validated in real life projects.

The research focuses in the "language of the problem" and concentrates in developing a lexicon as a first step in the ontology construction process. This activity results in a hypertext based "language extended

lexicon" (LEL), mainly oriented towards the language of the problem which is constructed and validated through a continuous and iterative process.

Overall, Julio provided the attendees with a good overview of the current trends in ontology development stressing the considerable influences exerted by the present technological advances.

Didar Zowghi "Requirements Evolution"

Notwithstanding the technological difficulties the speaker and audience encountered (it was not possible to get the LCD projector in an operational state) at the beginning of this session, Didar Zowghi from the University of Technology, Sydney, faced the problem with a straight face and determination. Instead of walking out of the room because the promised resources were not available, she proceeded to provide an excellent description of her past and current work in the topic of requirements volatility and how it affected the effectiveness of the overall software development process.

Throughout the tutorial, Didar provided details of the type of research she has been involved with in the past few years, the results that it has produced, and how she hopes the research progresses in the near future.

The main motivation behind the research was to determine whether requirements volatility (RV) is as a determinant factor in the quality of the associated software development processes and products.

The first step in their work required a clear and unambiguous specification of the term "requirements volatility" as understood in the context of the research. To this effect, RV is understood as the tendency of requirements to change over time in response to the evolving needs of the customers, the stakeholders, the organisation, and the working environment.

To be in a position that will enable the realisation of a thorough analysis to test their hypothesis, it was necessary to quantify a project's RV. This was done by establishing an operational definition of the term RV, i.e. the ratio of change in requirements (addition, deletion, and modification) to the total number of requirements over a given period of time.

The work began in 1995 and has been produced in stages (5 in total) the last of which is still under development. The first interesting thing to note is that it seems to have had the support of quite a large number of Australian organisations. This is something worth noting, because this kind of research needs a large amount of data in order to produce really useful results. As far as I am aware, it is not common practice in the European continent.

The first and second stages involved a pilot study and survey in which various IT intensive organisations participated. The researchers conducted a number of structured interviews and collected a fair amount of information through the analysis of a sample questionnaire that a number of them completed. The study indicated that there is a positive correlation between RV and the size of the requirements document, the time a project is delayed, and the percentage of project costs exceeded.

Looking in more detail at the reasons behind the project delay and the higher project costs they found a high degree of similarity in the route causes: poor management, highly complex and ambiguous requirements, lack of expertise of the project team members, and problems encountered during the design and testing phases.

The study also showed that there is no relation between an increase in RV and a decrease in the productivity of software development. The productivity was measured from three different perspectives: level of expertise of the developers, quality of the project management, and the quality of the source code.

With regards to the customer satisfaction, it was clear that RV is a determining factor given that, in general, a higher level of RV results in a higher level of dissatisfaction and volume of change requests.

The study also indicated that the use of a formal notation for the specification of requirements, together with the application of a well-defined requirements methodology, result in a lower RV rate.

The third stage of the research aimed to determine whether there was a negative correlation between the exceeded project costs and the over-run in development time. To this end, a survey was carried out on a sample of software projects that were developed over a period of two years. The findings that were achieved with a response rate of 21% indicated that RV is negatively associated with both schedule performance, and cost performance. Furthermore the determining factors for this association were: the use of a defined methodology for analysis and modelling, the frequency of communication between users and developers, and the execution of a requirements inspection activity.

During the fourth stage of the study, which was conducted using a pilot project in a specific organisation, a qualitative method for characterising and evaluating requirements changes has been developed. The researchers hope that it will be used by other researchers and practitioners to identify causes and reasons for the changes in requirements so as to gain an improved understanding of the nature of RV during the software development lifecycle, and develop a comprehensive taxonomy of requirements changes.

For the last stage of their work, the researchers are aiming to determine how a number of improvements in the development process of the case study organisation, together with a series of better requirements practices impact their working

environment and their work results. They hope that the organisation will benefit from these changes so as to help other organisations to conduct similar changes and obtain similar benefits.

Workshops

Workshop 1: Perspectives on Non-Functional Requirements

This workshop was conducted under a slightly different format from that originally envisaged by the R-Day organisers. In the first instance only two presenters took part in the session. Don Gause, research professor at the State University in New York, and Anthony Finkelstein, professor at University College London. Don Gause presented a number of examples, which provided practical evidence of the need for non-functional requirements. Anthony Finkelstein took a more laissez-faire approach and invited the audience to raise the issues they considered relevant on non-functional requirements in the context of the development of open-ended software architectures.

Don Gause "Problem Definition – Problem Solution"

In a well-structured presentation, Don provided the audience with convincing evidence for the need of giving serious thought to the non-functional requirements in a system development process. In the second part of the talk, he listed a number of "heuristics" that would enable the development team to cater for the system's non-functional issues. The presentation began with a demonstration of the dramatic effect that changing stated non-functional requirements can have on software development, reporting an experiment in which teams were given identical functional requirements but different priorities for non-functional aspects, resulting in enormous variations in development time and program size.

It was clear from the example that the development of a good system in a satisfactory manner is not straightforward. Ideally, a development team should aim to produce a system with a minimum development time. The performance of the system's functions should be the fastest possible. Furthermore, each function should use the smallest amount of storage, and the smallest number possible of executable statements. Last but not least, the clarity of the source code, and the output generated must be of the highest standard.

Each of the aims listed above is difficult to achieve and, given their very nature, contradict one another. For that reason, it is necessary to integrate as part of the development process a series of measures that will allow the work team to think of the best ways of achieving them so that one is not sacrificed in an effort to attain any of the others.

Don's alternative to handle a system's non-functionality, is to put into practice a number of heuristics that would help the development team to gain a clear picture of the users' expectations together with the constraints, preferences and responsibilities that must be satisfied in order to achieve a successful implementation. The list of possible useful context heuristics that must be taken into consideration runs as follows:

- An approved problem statement.
- A manageable set of stakeholders.
- A comprehensive set of meta-questions.
- A complete and clear set of design assumptions.
- A clear and complete set of use scenarios.

Having stated the set of heuristics, he focused on the difficulty of identifying and prioritising all of the relevant non-functional requirements in real-life developments, made all the more difficult by the variety of perspectives and preconceptions of stakeholders.

Overall, the presentation highlighted the importance of starting with a clear problem statement at the beginning of the requirements elicitation process, and an understanding of the way in which the solution will change existing work practices, in order to identify all the critical stakeholders. In the absence of such concrete inputs for non-functional requirements it is likely that the developers' own priorities, such as efficiency, will dominate over less tangible, but vital aspects, such as usability.

Anthony Finkelstein "Requirements Stability: an open question"

Using a more unorthodox style, Anthony took over the central stage of the City Suite at the Barbican Thistle hotel, and opened the floor to a discussion of the various aspects associated with the task of establishing an architecture that must support evolving business requirements over many years. This is something which, for many members of the audience, constitutes an even greater challenge than that of identifying the critical non-functional requirements for a single product.

He began by noting that it is typically changes in non-functional requirements, rather than the addition of new features, that tends to break architectures. He introduced a discussion on this topic by proposing that it is essential to understand the way in which a business is likely to evolve, and the resulting changes in non-functional requirements, and the consequences of meeting them, when the architecture is first being defined.

To prove this point, Anthony suggested that if an architecture has to move to multiple servers for scalability reasons, the requirements for security within the system will also increase dramatically, even if there is no change in the security needs of the business. He suggested that, while these informal

observations appear plausible, there has been no structured analysis of the relationship between changing requirements and architectural stability, in particular towards predicting triggers for disruptive changes or the metrics that would characterise the degree of change that becomes critical.

These introductory statements led to an airing of opinions and experiences from the audience, which confirmed that there was indeed ample scope for more research in this field before it is well-understood.

Workshop 2: Integrating Requirements, Design and Testing

The aim of the work discussed during this workshop was to provide the audience with practical examples showing how a structured and well thought out requirements process benefits the other stages of software development.

Elena Pérez-Miñana presented a structured approach to the generation of test cases aimed at concurrent systems which guarantees a good level of coverage without incurring the insurmountable costs associated to the task of testing systems with a complex level of interaction. It attempts to maximise the exploitation of a structured and complete system specification.

Jeremy Dick looked at a generic development process and showed how it is possible to define the necessary links between the stages of development that enable traceability, impact analysis and modelling to a degree that guarantees a system that meets the necessary criteria of reliability and consistency.

Finally, Ian Gibson showed how use cases could provide useful input to the various stages of a software system development process.

Elena Pérez-Miñana "Use Case–Based Testing of Concurrent Systems"

During her talk, Elena Pérez-Miñana, described a structured approach designed by Philips Research to generate system test cases using as main source of input the system specifications, complying with the guidelines established in the V-model, which indicate that the system specifications should drive the system testing procedure.

The approach aims to solve the difficulties incurred during the design of tests for concurrent, embedded systems. It provides the means to analyse the requirements so as to identify test cases and assess coverage metrics. The main thrust is to handle the complexity of these systems in such a way that it is possible to generate an adequate set of test cases; one that guarantees a reasonable level of confidence in its correct functioning once the test process has been completed.

The approach aims to help the testers to exploit the considerable domain knowledge they usually have, and make an effective use of their insight into failure modes. It provides the means to structure and analyse a system's requirements in a way that supports testing and can be reviewed for completeness. The main input is the system's use case based specification, which is transformed to enable the visualisation of several classes of use case interactions. The first step described consisted of the extension of the UML use case map associations with stereotypes, representing temporal dependencies. The second step consisted of the analysis of these dependencies through the development of three different types of interaction matrices, i.e. the inter-use case interaction matrix, the CRUD matrix, and the concurrency matrix. In each case, the testers are required to think of the system's functions from a different perspective which when combined provide a reasonable understanding of the interactions that should take place, and form the basis for test case design and adequacy criteria.

The final step of the approach consists of transforming the flow of events of each use case using the scripting language of SofTest, a test tool that generates all the test cases necessary to validate a system. These test cases are extracted from a cause-effect graph that the tool generates and which reflects the system's behaviour as depicted in the scripts that were fed to the tool as initial input. It is possible to have confidence that the system has been thoroughly tested because the tool generates as many tests as are required to traverse all the paths of the cause-effect graph. Furthermore, it makes sure that each path is traversed only once.

The approach was explained in the presentation by showing how it was applied to the generation of test cases for the demo of a networked video and Internet access system that Philips Semiconductors presented at the 2001 IFA.

Members of the audience noted that the approach did not seem very cost effective which is something Elena agreed with, noting that when faced with the awesome task of having to guarantee the reliability of a highly complex concurrent system, the approach provided a very good alternative.

Jeremy Dick "The System Engineering Sandwich"

Jeremy Dick, from Telelogic, described an approach that brings together the discipline of requirements management and system modelling.

Jeremy began his talk by stating his understanding of the different aspects of a requirements activity that have a role in a development process, i.e. elicitation, specification, analysis, change management. He also listed those commonly associated with the modelling activity, i.e. formalisation, support in problem reasoning, aiding communication between stakeholders. Once the audience was clear on the main concepts that Jeremy was aiming to marry together he proceeded to describe a reasoning framework that enabled the system development team to integrate both activities making the most of the fact that they complement each other.

The main idea underlying his approach is to think of the different layers of requirements that should be specified during the development process, meanwhile thinking of which models provide the best aid to help reasoning at each layer. Making sure that the models and the requirements at each layer constitute the inputs and outputs of the subsequent and preceding layer, it makes the tasks of structuring the system, implementing traceability, conducting impact analysis, etc. much more straightforward.

During the development process, the results of the work carried out is reflected in the analysis and design documents. In the specifications the model itself is encapsulated and explained in the form of a readable document. The vocabulary used is consistent with the elements of the model. It contains organised rationale in the form of satisfaction arguments traced to higher level requirements, and design justification that can be traced to lower level requirements. It has internal tracing of rationale to elements of the model.

A member of the audience enquired how his theory tied in to the Telelogic RM product, DOORS, and Jeremy was very emphatic in his reply declaring that the mechanism was completely tool independent. Nevertheless, he couldn't resist ending the presentation by showing all those present how DOORS Data Modelling and DOORS Analyst support the integration of the requirements management and system modelling tasks following the approach described during his presentation.

Ian Gibson "Applying Use Cases throughout the Development Lifecycle"

The third presentation of the workshop on "Integrating Requirements, Design and Testing" was carried out by Ian Gibson from Data Systems & Solutions (DSS). It provided a perfect finishing touch to the issues discussed during the workshop, presenting an intelligent approach to a software development process which exploited the different aspects of a use case based specification.

The different phases of the DSS Integrated Systems and Software Process (ISSP) are derived from ISO15288. It therefore includes, Analysis, Software Design and Implementation, Verification, Integration, Transition, Operation and Maintenance, Disposal and Validation. A system is shaped in iterative development lifecycles built around work packages that are based upon logically related sets of use cases.

The analysis phase is conducted through the application of FOIL™, a key feature analysis. The key features, i.e. the significantly behavioural patterns of the system under development, are studied using a data model and use cases that complement one another. The domain model embodied in the form of a data dictionary defines the terminology in the use case descriptions.

The key features are organised in a key feature matrix which provide an anchor point for the other elements that are built during system development, i.e. the domain model the use cases, the functional and non-functional requirements. Message sequence diagrams, and robustness diagrams are produced during the SW development and verification processes that take a detailed view of subsets of system behaviour.

Ian concluded his presentation by making it clear that the step which must be taken when an organisation starts integrating use cases into its development process is not an easy one, but the benefits that can be reaped as a result make it worthwhile.

Workshop 4: Research Methodology

Alistair Sutcliffe from the university of Manchester, and Roel Wieringa from the university of Twente, in The Netherlands, jointly provided a very interesting discussion on the impact that university research has on society, and how society, the world general, drives the manner in which university research evolves, in particular in Requirements Engineering.

They collaborated with one another and made sure the presentations conducted in the workshop complemented each other, giving plenty of room for the audience to participate. Even though they indicated that both their positions could be considered to oppose each other and breed controversy in the audience, I personally feel that it wasn't the case.

Roel Wieringa "Research Methodology in RE"

Roel started the workshop by describing the plan for the workshop. He presented his understanding of what constitutes RE research through the description of various research examples conducted at the University of Twente.

In his opinion, research can be classified in two broad categories:

Curiosity-driven research, which aims at knowledge that, is thought to be interesting as determined by a community of peers.

Utility-driven research aims at knowledge that is useful as judged by one of two groups: by business in which case the research tends to be short term, or by other researchers which is what is usually labeled long term research.

In his opinion, each type of research tackles a different kind of problem. In the case of utility-driven research, the problems, which Roel referred to as "action" problems, deal with the conceptualisation of means to change the world through the design, implementation of a particular solution. Meanwhile, curiosity-driven research tends to deal with "knowledge" problems, i.e. the construction of elements that result in a change of the domain knowledge. This is usually achieved through the proposition and subsequent evaluation of the validity of a hypothesis.

For any organisation, be it a university or research centre, the main difficulty lies in how it should go about designing its research. First and foremost, it must be done in such a way that it will enable the successful funding of all its resulting research projects. In the first instance, this seems to indicate that the research should be mostly utility-driven because this aims to find specific solutions for industry and they're the main source of research funding. In his opinion, a way around this that will enable any research institute to conduct both utility-driven and curiosity-driven research is to dress up the research so that it becomes amenable to industry and provides the funding required by the institute to carry it out. Personally, I feel that this is a perfectly legal way of steering your research, although I would question the need to "dress it up". In my personal opinion, I feel that what is really happening is that maybe this fuzzy division that separates utility-driven from interest-driven research does not really exist. Research of any kind aimed at improving the domain knowledge of a particular research area is equally useful, and interesting which makes the label "curiosity-driven" a bit superfluous.

Alistair Sutcliffe "RE Research Methods – Theory and Practice"

Alistair presented the audience with a description of RE research which constitutes a complete different perspective from the one outlined by Roel. Nevertheless they were by no means contradictory. It is simply an alternative view to the same problem domain, i.e. the current state of RE research.

In Alistair's opinion, the research can once more be divided into two broad categories:

Direct RE research which addresses part of the process/ process support, can influence practice in the short term, and is mostly "utility" driven.

Indirect RE research which investigates questions raised by direct RE research, and may influence RE practice in the medium term whilst at the same time be more likely to influence the trends in the RE research domain.

One point that Alistair referred to more than once, was the increase in the multidisciplinary nature of RE research. As science, technology and society advances, it becomes easier to build bridges across different disciplines thereby enabling a greater number of interesting research problems that can only be solved through the effort of researchers from more than one particular area of expertise. In the particular case of RE research it is already possible to see the influence of results from, for example, Cognitive Science, Linguistics, or Sociology.

In the second part of the presentation, Alistair discussed the kind of problems that are tackled through research. He spoke mainly of two types of problems, action problems, and knowledge problems. The first ones provide solutions to specific situations in a similar vein to the type of problems solved through the utility-driven research described by Roel. The knowledge problems are those unanswered questions that are natural to any discipline and which when solved make a particular impact to the knowledge domain to which they are associated.

Alistair concluded his presentation by highlighting the wide spectrum of action and knowledge problems associated with RE research, and stressing the importance of the multi-disciplinary nature of many RE problems.

After listening to Alistair's presentation, my feeling is that, when compared to other, more traditional disciplines, e.g. civil engineering, the maturity level of RE research is much lower, but the very nature of the way in which it is growing indicates it is a healthy research area which still has a long way to go.

Workshop 5: Metrics and Measurement

The aim of the work discussed during this workshop was to provide the audience with practical examples on the sort of metrics and quantitative approaches that can be used to monitor the quality of systems.

Gordon Woods as an independent consultant looked at the process of producing requirements and showed how it is possible to measure the success or otherwise of the requirements.

Paul Krause (University of Surrey) spoke on New Directions in Software Quality Control using MODIST (Models of Uncertainty and Risk for Distributed Software Development).

Finally, Simon Hutton (3SL) showed how to collect Requirements Metrics in a Project Management Context.

Paul Krause "New Directions in Software Quality Control"

Paul introduced his presentation with a simple statement that testing is time consuming and error prone. In addition, typically informal assessments of critical factors are used during the software development to assess whether the final product is likely to meet its requirements. An example from a telecommunications project was cited in which those components that were rigorously tested and therefore showed a higher number of faults before release had fewer faults after release.

In order to examine this a model based on Bayesian network technology has been developed and validated using data derived from metrics collected from projects in Philips in Bangalore. Although there were some caveats, he reported that there was a high degree of consistency between the model and the data but in essence the model was too naïve and inadequate for managing the quality of complex software projects. This is where the MODIST method and toolset was introduced. MODIST enables the building of tailored Bayesian networks from predefined templates. It uses:

- indicator nodes to define measurable metrics
- synthetic nodes to act as "super-nodes" that can then be split into more manageable child nodes
- distribution of probabilities rather than hard values
- scenarios and "what if" alternatives
- multiple networks to show the evolution of the project over time
- templates for modelling specific project phases

I believe that this technique has some potential. Rather like a Goal Structured approach it instills some rigour into the measurement process and forces decisions at an early stage. What is the purpose of a metric? How is it measured and reported? These sorts of questions have to be answered in order to produce the model, and as with any technique the more you put into it in the design stage the more accurate the final results.

Simon Hutton "Requirement Metrics in a Project Management Context"

Simon began his presentation with a Dilbert cartoon, which became a recurring humorous theme throughout ensuring that an otherwise dry subject was both stimulating and refreshingly different. As with all good presentations he introduced his topic slowly laying the foundations by defining what is project management and what standard tools and techniques are available. Break down the project into manageable chunks, plan the activities and assign resources. Then monitor project progress and resources against the plan. Nothing new here for the experienced practitioner, perhaps. Monitoring by exception, he argued was key to success. Use the technique of Earned Value Measurement, which can give a more accurate realisation of progress against the plan, in terms of both time and cost. Also manage your risks to minimise the technical, business and context risks and maximises opportunity in a cost-effective way.

Having laid the foundations, Simon then proceeded to introduce an interesting Goal Question Metric approach using an example. This provides simple metrics that are used to satisfy the sort of questions asked by project managers in order to realise their goals. This technique requires adequate collation of the metrics and presentation. Simon explained that Cradle was capable of providing this information at all levels.

Finally he concluded that in order to make it work you need to find out what the project manager needs to know, identify suitable metrics to answer his questions, agree targets and report in a format appropriate to the audience. Above all else the requirements team must be involved in the planning and presentation of the metrics.

I can only agree.

PhD Poster Session

The PhD student poster session presented eight very interesting posters covering different requirements engineering topics. There were posters describing approaches on problem frames, requirements modelling, requirements ambiguity management, semiotics, software system procurement and goal-oriented RE. Students had the opportunity to present their research results and discuss their work with established researchers and practitioners in the requirements engineering community.

The RESG Committee awarded prizes for the best posters. The winner student was awarded with a free registration for RE'04. All three runners up won a one year subscription to the RE Journal. The first winner was Carina Alves from University College London, supervised by Anthony Finkelstein. Her poster presented the TAOS Method to support the selection of COTS packages. The approach focuses on the matching analysis and negotiation process between stakeholder requirements and COTS functionalities.

The second winner was Islam El-Maddah, Tom Maibaum's student from King's College. His poster described an interactive software tool, called GOPCSD that is an integrated development environment for constructing requirements and generating formal specifications. The main objective of the approach is to bridge the gap between informal and formal requirements specification.

The third winner was Luncheng Lin from the Open University, supervised by Darrel Ince and Bashar Nuseibeh. Luncheng's poster described an approach using problem frames to guide the analysis of security threats and vulnerabilities.

Engineering Organisational Solutions from Human Information Requirements

12th May 2004, University College London.

Upholding its reputation on the organisation of events which provide its members, and the general public so inclined, with surroundings and resources to thrash out ideas on the latest developments in RE, the RESG held its May event at the University College London.

On this occasion, two speakers conducted the workshop, Ronald Stamper from the University of Staffordshire, and Yasser Ades from the University of Greenwich. Both of them have been working from the late 1970s on the specification of a method for information system development, i.e. MEASUR. Through the workshop the attendees were presented with the main concepts underlying it, the principal components, and a description of an application that was developed using MEASUR.

The first important difference between MEASUR and other software development methods lies in the type of information model built in order to produce the

database that constitutes the heart of the definitive system implementation. The specification procedure is driven by a set of rules and guidelines that enable the work team to have a clearer idea of the meanings and ontological dependencies that exist between the concepts handled by the information system.

The authors wish to capture in the model a different perspective, or different level of abstraction of the information the system under development should manage, that varies somewhat from what has been used by other models, e.g. UML. In order to do this it is necessary to define the minimum unit of information, "affordances", i.e. an invariant repertoire of behaviour. In order to do this; a language is provided to specify a system's set of affordances. It includes a set of operators for building ontological constructs that are more complex.

The well-formedness of the model is guaranteed because the rules that must be followed to produce it ensure that the applicants build a well-formed model – one that is in semantic normal form (SNF). This is in the same spirit that enables an engineer to make claims on the well-formedness of a relational database depending on the normal form it complies with. In the case of MEASUR, the well-formedness is achieved if the information model developed satisfies the conditions stipulated in the SNF. Fields of norms, affordances, responsibilities, ontological dependencies, agents, surrogates, and irreducible modules constitute the key MEASUR concepts. In addition to the language, the key concepts, and the well-formedness rules that the final model must comply with, there are three analysis techniques which help the development team progress from a vague understanding of the needs that drive the system development activity, to the specification of an overall system architecture fully compliant to the SNF.

At this point of the workshop, it was easy to believe that the MEASUR approach does differ from conventional OO methods of system development. Nevertheless, it was extremely difficult to decide on the gains of using MEASUR instead of any of the other currently available methods. For example, the description they gave of affordances has a high resemblance to scenarios, one of the most common ways of describing snapshots of a system's intended behaviour and which has been shown many times to be quite effective.

The second half of the workshop aimed to show the benefits of using MEASUR through the presentation of an example which attempted to show how it is possible to produce a better system although I feel that they somehow failed in their intent.

The application described corresponded to a university administration system, which was developed and monitored over a six-year period. Yasser attempted to demonstrate how the system's performance was better in view of the circumstances, i.e. that the resulting system complied with the SNF. In a similar way, he showed how it becomes much easier to decide on the correctness of the system, although I think his argument was not terribly convincing as it all boiled down to the well-formedness and the restrictive view of the world needed to ensure that this was the case. He also briefly mentioned the considerable savings in development subject to the deftness of the developers in using MEASUR.

Overall, an interesting presentation conducted by people firmly convinced in the benefits of the approach they were describing. Nevertheless, I feel the audience did not completely buy into the idea.

Ronald and Yasser concluded the workshop by asking the attendees for suggestions on how to improve their description of MEASUR. The audience suggested that maybe providing more working examples instead of so much theoretical background is a better way of describing their work. I agree fully with this and would also suggest that they provide more examples describing what they mean by the new concepts they are introducing together with more detail in the comparison of the benefits gained by using their method because it was difficult to grasp how system developers are better off specifying fields of norms, affordances and surrogates in place of information flows, entities and OO objects.

As a final word, it might be worth pointing out that the speakers had prepared a number of examples in addition to the one described in the latter part of the presentation but, as is common with many of the RESG events, the discussions that arose throughout the afternoon did not allow time for this.

RE-Papers

Surrogacy

Ian Alexander
Independent Consultant

Practically the first thing the budding requirements engineer learns is to go and get requirements direct from "the users" (whoever that might mean). It's an odd thing, then, to discover that direct from-the-horse's-mouth requirements gathering is rare, difficult to organize, and quite often actually impossible.

Why is this? In a word, it's because in our complex society, people often stand in for other people, speak for them, or are chosen to represent them. Standing in is a defining characteristic of 'representative

government' (where not even speaking for the people is worse) and it is equally characteristic of business organization. This on-behalf-of relationship is surrogacy, and people who practice it are surrogates. There is not necessarily anything sinister about this – unlike the situation of the surrogate mothers in *Brave New World*; but the wide extent of the practice does suggest that there is something strangely unrealistic about current requirements engineering theory.

There are several reasons why we often can't meet "the users" directly:

- They haven't been born yet, or are still in nursery school for instance, the soldiers who will use the vehicles being imagined for 20 years hence. Of course the problem arises (in less acute form) for any future system: nobody yet knows exactly what it will be like, and nobody has used exactly it: so in truth, you can never talk to the users of what you haven't built yet.
- There are too many of them, for instance the millions who use portable music players or who drive family cars.
- They don't know the technical situation (or even the situation on the ground), for instance the public who are concerned in a general way about the level of safety of a factory, the environmental impact of a nuclear power station, or the financial probity of an investment company.
- They are too busy, cannot be spared from their work, are no good at expressing themselves, may not talk to outsiders as it is a security risk, etc etc.

One other kind of surrogacy must be mentioned: we ourselves stand in for stakeholders of all kinds when we help to express their requirements for them. We naturally hope that we are amplifying rather than attenuating channels, but as Shannon so righly observed, there's no such thing as a noise-free channel.

How can these obstacles to requirements engineering be overcome? Plainly the usual answer is to appoint surrogates: in which case the surrogate stakeholders (for we cannot in fairness now call most of them users) act both as our channel to the stakeholders they represent, and as a threat to our ability to find out what is actually required.

The unborn soldiers can be represented by some current soldiers, while the non-existent enemy and weapons can be modelled by domain specialists, simulation, and the creation of suitable scenarios.

The operators of the next, unbuilt, version of the system can often stand in for themselves: they can imagine what the extra functionality will be like, or can be helped to do so with prototypes, mockups, walkthroughs and so on. And I've given the game away by talking about versions: iteration in the form of evolutionary development, rapid prototyping, agile methods, what you will, enables people to glimpse future system behaviour in small imaginable steps, so present users insensibly grade into future ones.

The mass-market consumers (the role is a hybrid of purchaser, operator and functional beneficiary) can be represented by a statistical sample, i.e. a small group chosen to be typical in aggregate of the consumer population as a whole. A few tens or hundreds of randomly-selected people (possibly carefully matched for age, sex and purchasing habits with the intended population) speak not so much with their own voice but with the averaged voice of the market. They speak to people with other surrogate roles, marketing or product management (the two are different, but there are clear connections). Those people decide, on behalf both of the organization (itself a surrogate acting on behalf of financial beneficiaries such as shareholders and directors) and of the consumers what the sample of consumers indicate that the mass-market consumers will in future (there is present-for-future surrogacy in this too) choose to buy, and hence decide what products to develop. Did you follow all that at the back? Please pay more attention…. Surrogacy can get complicated.

The employees who can't speak for themselves are most often represented by a purchasing or procurement department. The people in those roles again represent both the employees who need to be provided with new hardware or software, and the organization itself. Each buying decision is a trade-off between the needs of the workers and the cost and risk to the organization. One effect of surrogacy is thus to merge requirement conflicts within the minds of individual stakeholders, whereas classical requirements engineering would look for different viewpoints from different people.

The public who don't know the details of the law or the doubtfully-pure insides of a business are represented by a salaried official such as a regulator or inspector, who is paid to speak up on or investigate matters financial, safety, environmental, or whatever. This role is special, as there are generally statutory or at least openly-agreed rules governing what regulators must do, and indeed empowering them explicitly to act as surrogates to protect the public. In this they are quite different from most other kinds of surrogate that requirements engineers might meet.

One such is the head of department who claims to speak for all departmental processes under his or her control. The problem with that – it's fine in theory, perhaps – is that whatever may be supposed to happen on paper, what people in the department actually do is always, and I mean always, different in practice. Even in the most highly proceduralised domains like air traffic control, I am assured by people who know that rules may and must be broken when there is good cause: if not, the system just wouldn't be able to handle the traffic.

This means that if the requirements engineer wants to find out what really happens, as opposed to getting the

party line, there is no option but to speak to (or observe) the people 'on the shop floor'. Perhaps there was a time when the boss knew everything, as in the rhyme about the famed Dr Jowett of Balliol College, Oxford:

I am the master of this college

And what I don't know isn't knowledge.

Lesser mortals, however, can rarely hope to know all the details of what everybody who works for them has to know, especially in rapidly-evolving fields like software and electronics. We are now familiar with the paradoxical situation that it is all too often the most junior people who have the practical operational skills; their managers have administrative and sales expertise, but little or even no knowledge of technical matters. This may be a sadness – how much better it was when an Isambard Kingdom Brunel could be at once entrepreneur, manager, and master engineer! – but it is everyday reality. It follows that if you want to find out the operational requirements of the people who will actually use whatever it is you are building, you must speak to them directly.

If, of course, you can get to speak to them – rather than a surrogate. Where you can't, then the question becomes 'how should I elicit requirements from these kinds of stakeholder, given what they represent and the authority, statutory position, and knowledge that they have. There are plenty of claims for methods and tools, but few of them come with advice on *who* to use them on. Perhaps it's about time they did. At least it's fairly plain that the requirements of, say, parliament, a safety regulator, marketing, and procurement are going to have to be approached differently.

This tangled tale reveals a surprising truth: surrogacy, far from being a rare or obscure phenomenon, is actually pervasive in society and business; and it is perhaps especially important in exactly the parts of a business that we ought to be most concerned about: development. The surprise is how little attention seems to have been paid to it: maybe its very pervasiveness has fooled us – the hardest things to see are often right under our noses.

Mother (in annoyance): Where's my handbag?

Children (in chorus): In your hand, mummy!

So, next time you're gathering requirements for a project and are wondering where the surrogate stakeholders are: start by looking in the mirror.

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 July 2004

FARE: A Requirements Engineering Method for Product Families

Muthu Ramachandran[1]
Leeds Metropolitan University

Introduction

The concept of a product family was defined as long ago as 1976 by David Parnas:

> "We consider a set of programs to constitute a family whenever it is worthwhile to study programs from the set by first studying the common properties of the set and then determining the special properties of the individual family members." [1]

Nowhere is the concept more clearly demonstrated than in the domain of consumer electronics products that incorporate both hardware and software, for example TV sets, VCRs etc., where all members of the family have a basic set of common functions with many variants and add-ons appearing as the product line is extended. Such product families typically evolve into complex trees.

Such product lines have a long and successful history of hardware component reuse, but large-scale reuse of software has been much more difficult to establish. The difficulties inherent in building software systems for product families have been widely recognized. The earliest work to address the problem of systematic large-scale reuse concentrated on design and documentation for reuse, the creation of repositories of reusable components and the use of architecture frameworks. The landmark ESPRIT REBOOT project [2] addressed all of these issues, and also the organizational changes necessary for successful reuse. SCV (Scope, Commonality and Variability) analysis is an approach which provides a systematic way of identifying and organizing the product family to be created [3]. SCV explicitly identifies commonalities, i.e. assumptions that are true for all family members, and variabilities, i.e. what can vary among members.

Most published examples of the use of SCV, like most of the earlier work, has concentrated on the task of creating designs from a well-engineered set of requirements. However, in the type of domain we are describing, it is a difficult task to formulate and track such a set of requirements for multi-site development teams dispersed across the globe, while maintaining the integrity of the product family and maximising potential reuse benefits.

[1] This work was carried out when the author was with Philips Research Labs, Redhill, UK

Our aim was to find a way of incorporating SCV analysis early in the product life-cycle as part of a requirements engineering process which could be used to transform product and stakeholder requirements into functional requirements. The potential benefits of capturing common requirements for the product line include:

- Reduced cost to new product from product line base
- Reduced cost to design
- Reduced cost to document
- Reduced cost to implement
- Reduced cost to design tests
- Reduced cost of maintenance
- Reduced cost for requirements management

This paper proposes a method for incorporating SCV analysis as part of a requirements engineering process, and describes its application to a real industrial case study.

The FARE method

In order to achieve a systematic way of preparing a requirements specification which would include SCV analysis, we developed the FARE (Family-oriented Analysis and Requirements Engineering) method. The FARE method incorporates a process for carrying out the analysis and a checklist for assessment. The key features of the method are as follows:

- It is specifically geared towards producing requirements for a family of products.
- It provides a process for conducting the analysis.
- A checklist is used to ensure that the SVC analysis has been carried out, and to identify areas for improvement.
- It can also be applied retrospectively to existing sets of requirements, to allow future reuse opportunities to be identified.
- Cost-benefit analysis is built into the method.
- It incorporates method-specific support to make the analysis more effective (for example, support for use-case analysis.

FARE is therefore a method which is used alongside existing analysis methods in use with the organization's software process, but extends that process to make commonalities and variabilities explicit, and to identify potential reuse opportunities at an early stage. In the following sections we examine some the above key features in more detail.

The FARE process

The process consists of five phases (as shown in Figure 1): Prepare, Plan, Commonality and Variability Analysis, Quantify and Review. Following the review phase, there may be one or more iteration of the process from the planning stage onwards to resolve any outstanding issues. In practice it has been found useful to appoint a moderator (this could also be a small team) to oversee the entire process, ideally someone who is not involved in any of the development teams.

The phases typically include the following activities:

Prepare: The scope of the analysis is established, and an initial analysis (feasibility study) is carried out.

Plan: The checklists for assessment are prepared, and these and the process are explained to the participants, facilities are set up, domain boundaries are identified.

C&V Analysis: This phase may run in parallel with other requirements analysis activities, or it may use as its input existing requirements documents. The commonality and variability analysis are carried out at this stage, and each commonality identified represents a potential reuse opportunity.

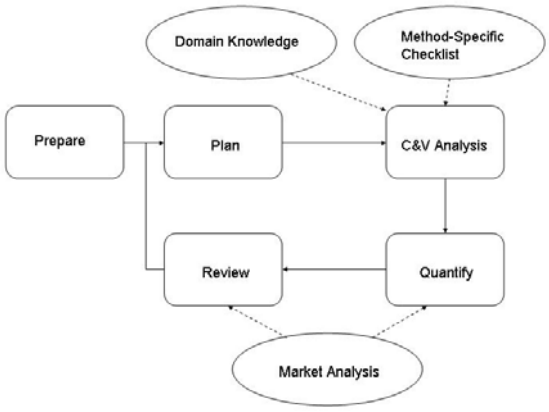

Figure 1: The FARE Process

Quantify: The parameters of variation are generated for each variability identified. For each commonality and variability identified a cost-benefit analysis is carried out (described in section 5), which involves assigning a priority to each variability based on the market analysis.

Review: The review process includes the application of checklists to ensure that the C&V analysis is complete and consistent, to ensure that it is consistent with the market requirements, and to highlight areas for improvement within the process itself. This can take place at the same time as validation of the functional requirements, or may be carried out later as a separate activity.

Domain knowledge and expert experience is an essential input to the process. This can be acquired from domain experts, product development experts, and product requirements documents. Engineers with specific responsibility for software reuse should also be involved in analysis and review activities. The market analysis is the key to assessment of potential benefits and risks.The role of the method-specific checklist is discussed in the next section.

Checklists for assessment

In order to make the process more systematic and process-centric, we have formulated a number of checklists and guidelines on how to categorize, assess and validate requirements for a product family. Some of these checklists are standard, while others are tailored to specific analysis methods being used. The checklists fall into the following categories:

- General C&V analysis checklist. These are requirements for any C&V analysis, e.g. **CVA3**: Prepare a glossary of common terminology, including abbreviations and acronyms
- Commonality analysis checklists (CA), e.g. **CA2**: Identify and list baseline product features
- Variability analysis checklists (VA), e.g. **VA3**: Bound the variabilities by placing specific limits-such as minimum and maximum values- on each variability.
- Cost-Benefits analysis (CB) checklists, e.g. **CB2**: Conduct or have the market analysis/study report ready before introducing new features into the product line

Cost-benefit analysis

For each requirement that is identified as a requirement for a product family we need to conduct a cost benefit analysis. This should include issues like what will be the cost of implementing this requirement and also how to maintain this requirement. For example, let us consider CVA3 listed above.

Cost: 1 person-month of work force. This should include producing the glossary and circulating amongst the development team and/or anyone involved within the organisation.

Benefit: Everyone understands what we mean when capturing product line requirements. The major profit is the time which is saved for years to come.

Poulin [4] describes the C-B models as the analysis of reuse problems by itemising all the cost factors influencing reuse, summing the values for these factors, and basing the business decision on the result. It also determines the value of reuse by first finding or estimating reuse costs and benefits. We then simply determine the total benefits and total costs by summing each benefits, $B_{i,}$ and each cost, C_i:

$$\text{Benefits} = \sum^{n}_{I=1} B_i$$

$$\text{Cost} = \sum^{n}_{I=1} C_i$$

Case study: a TV product family

TV systems are getting more software intensive due to demands for various services and the emergence of digital broadcasting has impacted this consumer market rapidly. A TV product line inherits from general display systems and then splits into two major categories such as analogue and digital systems. In our work we only consider the digital TV family. The digital TV splits further into networked systems and standalone (home) systems. The networked systems are often used in a hotel and museum, etc. A standalone digital TV system consists of more than 100 variations and which grows rapidly based on the market competition. These product variations include plasma TV, Flat TV, normal TV, DVD-combi, VHS-combi, and furthermore we also need to include TV sizes such as 14", 21", 28", 32" etc. The software need to be configured for each product variations. The user interface varies quite significantly. The UI management needs to consider languages, country, buttons, icons, and messages for internationalisation.

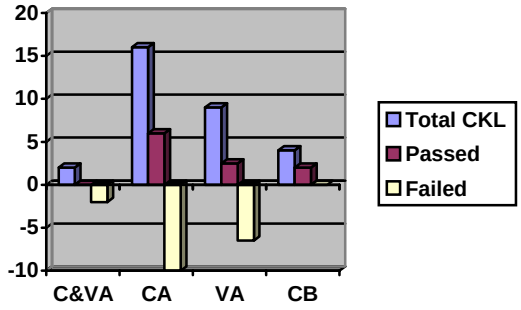

Figure 2 C&V assessment result for a TV product line

As shown in Figure 2, the TV product family successfully passed 35% of our checklist even without a formal process for conducting SCV analysis. However it has failed mainly on CA and VA checklists which are shown in the white bars in the negative scale.

Lessons Learned

This work was carried out as part of a major requirements engineering consultancy project. One of the aims of the product development group was to reduce development cost by reusing artefacts of a product line. This product line was akin to their current business. Hence the need for identifying and structuring requirements for product family has emerged. The lessons learned were:

- Always consider product family/line when collecting requirements
- Use a good structuring approach to categorise requirements which helps to conduct SCV analysis for a product line in the future if not immediately
- We found that the checklist we identified helped to validate and analyse existing requirements
- The moderator needs to be a domain expert for conducting SCV analysis in the context of FARE method

- The requirements that failed our checklists revealed that the initial or existing requirements were not represented precisely or are ambiguous
- The improvement in product development was noticed for the following product immediately after our FARE analysis
- Method-specific checklist (in our case Use Case) helped to improve existing requirements for immediate product development which can be used as requirements validation method

Conclusions

In this project we have learned how to customize various RE techniques and best practice guidelines on the effective use of RE techniques. We have tried to merge concepts from RE techniques and commonality & variabililty analysis techniques. It is one of the challenging area for researchers both in product family as well as RE.

Acknowledgements

The author would like to thank Dr Paul Krause at Philips Research Labs, Redhill, who supported this work. Also we would like to thank colleagues at the School of Computing at LMU who have provided valuable comments on this paper.

References

[1] Weiss, D., Commonality Analysis: A Systematic Process for Defining Families. Second International Workshop on Development and Evolution of Software Architectures for Product Families, February 1998.
[2] E-A. Karlsson (ed.), Software Reuse: A Holistic Approach, Wiley, 1996.
[3] J. Coplien, D. Hoffman, and D. Weiss., Commonality and variability in software engineering, IEEE Software, November/December 1998.
[4] Poulin, S. J. Measuring software reuse, Addison Wesley, 1997.

RE-Publications

The Requirements Engineering Handbook

Ralph R. Young
Artech House, 2004
ISBN 1-58053-266-7

Ralph Young is the author of the popular *Effective Requirements Practices* (2001), and he has now followed up that guide to project activities with a handbook for people who look after requirements (perhaps especially in large organisations). There is obviously some overlap - both books discuss process issues and offer a taxonomy of requirements, for instance; but this book is both more up-to-date and oriented more towards the practitioner, and it tells the 'how to do it' (at the level of project activities, not of the detail of analysis and modelling) where the earlier book said what had to be done.

I should declare an interest here: Young quotes and references several aspects of my work, including on Stakeholders and the *Writing Better Requirements* book. In fact he is consistently generous and collegiate in style, quoting and praising numerous named sources such as Ivy Hooks' *Customer-Centered Requirements* and Ellen Gottesdiener's *Requirements by Collaboration* on different aspects of the field. Therefore, despite my deputy midwife's role in bringing this book into the world (with suggestions and comments on the manuscript), I think I can still write a reasonably objective review of the published book.

The book is organized straightforwardly into ten chapters.

The first introduces The Importance of Requirements, but in contrast to other books (including mine) where a similar argument is addressed mainly to managers wondering why they should bother, the message here is for the Requirements Analyst (RA) -- for whom indeed the whole book is written. In the section 'Factors affecting your career decisions', for instance, Young at once recommends "that you meet with your PM very early ... to allow you to conclude whether you can be effective in your role." In other words, there's no point taking on the job if you won't be allowed to do it as described in this handbook. This is not a book, clearly, which is going to waffle on about academic theories or the intricacies of modelling.

Chapters 2 and 3 look at the Roles of the RA and the Skills and Characteristics of an Effective RA, respectively. The roles and skills are presented both in the text and as matrices: e.g. facilitation is needed in all life-cycle activities, whereas advising on tools and techniques is needed in system initiation, analysis & design, and during operations. The skills are presented for 3 levels of analyst, from entry-level to senior, and a detailed job description is presented in a table that takes up just over 2 pages -- covering the skills, knowledge, responsibilities, and measures of performance. This is far more systematic (and dare I say it, better) than anything in the current literature, though Sommerville & Sawyer's *RE: a good practice guide* is a worthy precursor (but it focuses on process improvement rather than on practitioners). Chapter 7 revisits this core theme of the Handbook by discussing

the RA's Specialty Skills (it doesn't mean 'on specialty requirements' but 'what you specially need to be able to do', like facilitating, estimating, and configuration control).

Chapter 4 looks at Types of Requirements. These are grouped into customer needs and expectations, analyzed system requirements such as functions, '-ilities', and performance, allocated subsystem / component requirements, and checks such as qualification requirements. To prevent the chapter from getting too dry, there is a welcome case study story to round things off. Young maintains a constant sharp focus on what needs to be done, by 'you' the reader and RA. He refuses to be side-tracked into details such as what attributes you should use to document your requirements, but provides plenty of pointers and references to sources in books and websites.

Chapter 5 covers Gathering Requirements (aka eliciting, discovering, capturing). As Young says 'A lot of time and effort is wasted' and he gives detailed reasons for this. He does not make the mistake of supposing that an 'effective proven procedure' is the sole solution: indeed, the lack of one is just reason #7! The others are crucial practical matters, like the fact that the project is 'just getting organized and things are confused' and tellingly 'There isn't much pressure to meet the schedule yet'. You can tell that he has been there and seen that. The RA has to work in the real imperfect world; tools, techniques, and even processes are only a part of the solution. There is a daunting but on inspection highly sensible 28-step checklist for project requirements gathering activities: 'perform requirements gathering' is step 22. There are plenty of pitfalls for the unwary: Young doesn't say that fools rush in, but the implication is there. There's even a suggested 'topics for training for RAs', i.e. an outline syllabus. As you'd expect, this covers not only 'the mechanisms, methods, techniques, and tools' but also 'having and using a requirements process' and 'reducing rework on the project'. After all, if systems engineering is about anything, it is the carpenter's maxim: measure twice, cut once. Young is right to emphasize the reduction of rework; few other authors even mention it.

Chapter 6 looks at best practices: something easily said and less easily done, because "it's a lot of work". 30 best practices are listed and described in a paragraph or two each, but first there's some practical advice on how to get people to adopt them, and at the end there's again a case study which tells a detailed story of a real project experience, and not a comfortable one at that.

Chapter 7 has already been mentioned. It again looks at a list of topics. These are framed as Socratic questions, like "Why are requirements errors so devastating and how can RAs help address the problem?" These are answered at length, quoting various authorities, and discussing the difficulties and risks of the proposed solutions. For instance, use cases are not accepted uncritically: "Be sure to digest [Kulak & Guiney's *Use Cases: requirements in context*] considered list of problems related to using use cases before making a decision"; Tim Korson's *Misuse of Use Cases* is also applauded.

Chapter 8 discusses 'An Integrated Quality Approach'. As with standards, the nice thing about quality approaches is that there are so many to choose from; Young mentions TQM, 6Σ, Quality is Free, Zero Defects and ISO standards among others. Since "quality initiatives are difficult to sustain" (the enthusiasm fades when the guru walks away), Young rightly emphasizes the integral connection between requirements and quality, and the essential role of management. He then stresses some simple Guiding Principles, the first of which is "Customer satisfaction is imperative for our continued existence." The others are just as necessary. Using and valuing these helps "create a sense of purpose for employees". After that, quality improvement techniques have a chance of working.

Chapter 9 presents A Vision for Requirements Engineering. It asks how we should support project managers, customers, and developers through better requirements work. Young answers this in typically pragmatic fashion: "This is not a job for the most junior member of the team or the least talented member of the group." Indeed not.

Finally Chapter 10 looks at Moving Forward. Young recommends "not trying to change everything at once", and suggests selecting "three to six practices or areas for your own personal commitment list." The task, in other words, is not something theoretical, but a matter of learning how to do the job better -- and doing it.

This is an industrial handbook with a constant focus on 'how to'. The perspective is, as with *Effective Requirements Practices*, largely from the point of view of technical management, responsible for overseeing the slippery and complex activities that do much to determine the success or failure of system development projects, small and large.

The reader is helped by a full list of acronyms, a good and detailed glossary, a long list of references and an excellent index.

Requirements people in industry, whether their firms are as experienced as Northrop Grumman or relatively new to systems and software engineering, will find this a practical and thorough guide to what to do.

RE-Sponses

Karl Wiegers' article in the March edition of Requirenautics correlates very closely with my own experience in large business systems IT. In the section 'What To Do', he states that '…the real challenge is to give the bulk of practising requirements analysts the practical knowledge, skills and judgment they need to be effective on a difficult job.'

There are two serious problems here: firstly, the majority of those currently carrying out RE are just not interested in improving their skills, and secondly, their managers simply do not care. They have settled, comfortable lives and get paid every month, so why should they be bothered, especially as when things go pear-shaped? It is the designers and programmers who tend to get blamed anyway.

There is an element of 'second-level ignorance' here, in that the managers in particular do not know enough to even realize that they do not know enough to ensure RE is performed adequately for their organizations.

A frightening prospect is that model-driven development directly from an object-oriented domain model will become a reality as soon as a major reputable provider releases a toolset. This will mean that RE (with acceptance testing) will be the only game in town, as systems analysts, data analysts, programmers, etc. will no longer be needed to build and deploy large IT systems. This, coupled with Linux and open-source tools, will have a cost model that will be absolutely compelling. A happy-go-lucky approach to this form of model-driven development will inevitably result in disasters, some with high profile. The potential for this situation to arise will be made worse by analysts and programmers trying to move into RE in order to stay in employment.

Unless IT development managers from board-level down are either educated, forced or plain scared into realizing that RE must be performed by knowledgeable specialist practitioners, I do not see this situation improving in the future; projects will still overrun their timescales and budgets, and will fail to satisfy their Users. The concomitant waste of money will continue, especially in the public sector and the larger organizations where the above attitudes seem to me to be the most entrenched.

Martin Lawrence, RESG member.

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:
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/

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-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: Dr 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.