Summary of the 12th Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'06)

Vincenzo Gervasi
University of Pisa
Italy

Pete Sawyer
University of Lancaster
UK

Barbara Paech
University of Heidelberg
Germany

Abstract. The twelfth edition of the Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'06) series took place in connection with the International Conference on Advanced Information Systems Engineering in the buzzing city of Luxembourg, Grand-Duchy of Luxembourg, on the 5th and 6th of June, 2006. The conference was organised by Vincenzo Gervasi, Pete Sawyer and Barbara Paech, with Eric Dubois, Andreas Opdahl and Klaus Pohl serving on the REFSQ Advisory Board. This summary gives an overview of the changes in the conference format which were introduced in this edition, as well as an account of the presentations and lively discussions that took place at REFSQ'06.

1. Introduction

The push for *quality* in *requirements* has not decreased in the last decade – instead, it has sharply increased, with software-intensive, computer-based systems becoming more and more pervasive in our social and economic structures, as well as in our everyday life. The importance of defining high-quality requirements prior to building such systems is universally recognized in industry and academia. *Requirements engineering* (RE) is the discipline that studies the process of eliciting, negotiating, documenting, verifying, and validating requirements, applying techniques from computer science, psycho-social sciences, economics, and engineering. The REFSQ workshop series was established in 1994 to foster research in requirements engineering, and has published, to date, almost 200 papers on the subject – many of them presenting ground-breaking and seminal work – as well as a number of special issues of several journals. The format of the workshop, focusing on interactive, detailed discussions among participants over presented papers, has been highly successful in promoting active involvement and encouraging the emergence of innovative ideas.

REFSQ'06 was the 12th edition in the series, and introduced a number of changes to the event format aimed at making participation more enjoyable to attendants and to promote widespread diffusion of the research presented at REFSQ.

The call for papers invited submissions on any aspect of RE and its relation to other fields, including:

- Estimation in RE
- Best Practices in Specific Domains
- RE for Autonomic and Self-conscious Systems
- Encoding, Transferring, Analyzing and Applying RE Knowledge
- RE in Large Projects: Compositionality and Scalability

Case studies, experience reports and industrial problem statements were particularly encouraged.

In response to the call for papers, 37 papers were submitted, of which 15 were accepted for presentation and discussion in Luxembourg: 12 full papers, 1 position paper, and 2 industrial problem statements. This represented a healthy interest in REFSQ and – for the fourth year in a row – an increase in number of submissions. Each submitted paper was reviewed by 3 program committee members; most of the reviews were – as per long REFSQ tradition – very detailed, with many suggestions for improvement. More suggestions and opportunities for improvement of the papers before their final publication came from the plenary discussions at the end of each session, to which two thirds of the entire time budget for the event were reserved.

The fact that over half of the papers (20 out of 37) were submitted from extra-European countries testified the international relevance of REFSQ, unhindered by its permanent location in Europe.

Thirty-one people attended from Belgium, Canada, France, Germany, Italy, Japan, the Netherlands, Norway, the UK and the US. As described below, 2006 was the first time that REFSQ had been opened to attendees who were not co-authors of any of the presented papers and approximately one third of the attendees were in this category. The experiment seemed to work, with all attendees contributing to the REFSQ tradition of full, open, constructive and friendly discussion.

2. Tradition and Innovation

REFSQ has always had a tradition of being more than anything else a forum for in-depth discussion among specialists of current and innovative work, and has enjoyed an excellent reputation in this respect. However, it was felt that some of the constraints of the formula (e.g., that only authors of accepted papers could attend the event) were becoming too restrictive now that the worldwide RE community has grown to respectable proportions. It was thus decided to introduce a number of changes to the organization which, while respecting and maintaining the original spirit of the series, could improve participation and contribution – especially from industry. These changes are detailed below:

- Attendance to REFSQ was opened to non-authors;
- A special effort was made to invite industrial papers (both in the industrial problem statement category and as regular papers) and attendants;
- The program was extended and more varied, including a panel session and a special “greenhouse” category for papers describing maturing research, with shorter presentations;

This new formula motivated the new label of “Working Conference” (as opposed to “Workshop”) for REFSQ. Another change was introduced, based on participants' comments from previous editions of the workshop: the traditional “first slide”, which started each presentation with a diagram showing the context of each work with respect to a fixed set of concepts (*people:* User/Customer, Requirements Engineer, Software Architect/Developer; *artefacts:* Needs, Specification, Design/Architecture; and the *environment*) was replaced by a new diagram, based on keywords indicated by authors upon submission, showing all the papers in a session and their relationships with an open set of key concepts. With the new slide, presenters were asked to comment not only on the context of their work, but also on how it was related to the other papers in the same session (of which they were often discussants). An example of the new first slides is shown in Figure 1.

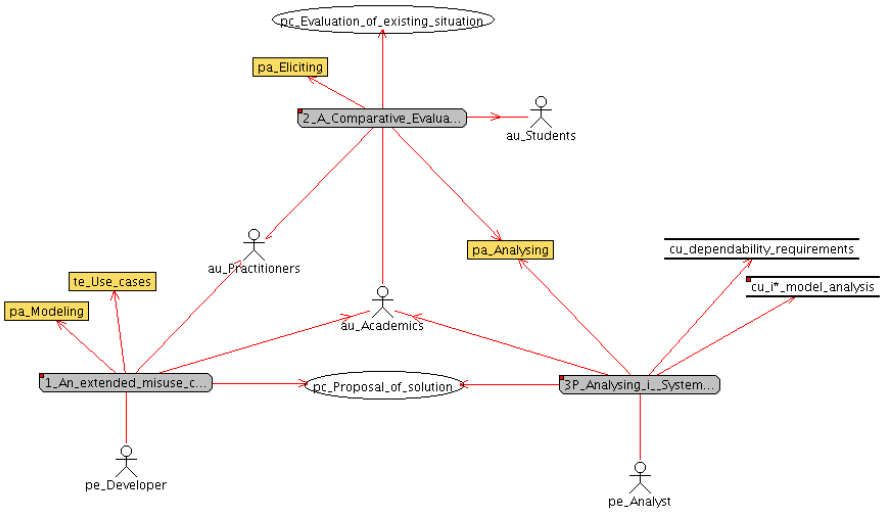

Figure 1: An example of the new first slide

In the new format, papers (in grey boxes) are linked to keywords from several groups: *Paper Class* (based on R. Wieringa et al.s’ classification [1][2]), *Process Area*, *Performer*, *Artefact*, *Technique*, and *Intended Audience*. Custom keywords could also be specified. The full list, as well as a tabular index of all the papers, is provided in Section 6 of this report.

3. Conference Structure

We organised the conference in 6 main sessions, which were devoted to *Quality Requirements* (chaired by Sjaak Brinkkemper, with discussion facilitated by Camille Salinesi); *Case Studies* (chaired by Pete Sawyer, with discussion facilitated by Patrick Heymans); *Quality of Requirements* (chaired by Barbara Paech, with discussion facilitated by Andrea Herrmann) on the first day, and a short session on *Formal*

Methods (chaired by Nazim Madhavji), followed by one on *Elicitation* (chaired by Pete Sawyer with discussion facilitated by Cornelius Ncube) and one on *Complex Systems* (chaired by Andrea Herrmann, with discussion facilitated by Nazim Madhavji) on the second day. Moreover, in parallel with the session on Elicitation a *Panel on the Interplay between Requirements Engineering and Process Management* was held in an adjoining room.

To ensure the effectiveness of the format, each full paper presentation was limited to 15 minutes and followed by 20 minutes of discussion. Furthermore, each paper discussion was initiated by three discussants — usually other paper presenters from the same session. At the end of each session, the major topics raised by the talks or the related discussions were elaborated after introductions by the session discussion facilitators. Presentations of short papers were restricted to 10 minutes with 15 minutes set aside for discussions initiated by two discussants.

As already said, each presenter was asked to start his or her talk with a slide putting the work into context and relating it to other papers in the session. Additionally, each presenter and each discussant was asked to summarise his or her own views on the talk by answering the following questions:

- Which quality features are addressed by the paper?
- What is the main novelty or contribution of the paper?
- How will this novelty or contribution improve RE practice or RE research?
- What are the main problems with the novelty/contribution and/or with the paper?
- Can the proposed approach be expected to scale to real-life problems?

The conference was closed by a general discussion, including an evaluation of the event itself by the participants. Details of the various sessions and other events are provided in Section 4 below.

4. Session Summaries

Vincenzo Gervasi welcomed the participants and explained the format of the conference. After presenting the statistics of the submissions everyone introduced themselves. Because of the time constraints of some speakers the presentations started with Session 2 followed by Session 1.

Session 1: Quality requirements

Sjaak Brinkkemper, the session chair, introduced the first paper **An extended misuse case notation, including vulnerabilities and the insider threat** presented by Lillian Røstad[1].

A misuse case (originally introduced by Andreas Opdahl and Guttorm Sindre) is an inverted use case, where the actor is an attacker and the goal is a threat to the system. The proposed extension introduces an inside attacker (or "insider") in addition to the traditional outside attacker, and an internal *vulnerability* alongside the traditional threat. Insiders can exploit vulnerabilities to facilitate attacks. The author motivated this extension of the notation with the need to identify the "attack surface" and

[1] In the following, we will underline the name of the presenter in the author list of each paper.

supporting communication with the customers/stakeholders to discuss the consequences of attacks and countermeasures. A straightforward process of identifying attackers, insiders, misuse cases, threats, vulnerabilities and thus security requirements as countermeasures (possibly iterated) was suggested. An example about access control was presented, where an *emergency access* was misused thus presenting a vulnerability. Another example, of an insider in the development team injecting a backdoor or a bug, was also presented.

Discussants focused on the notation, questioning whether it was really necessary to have a different notion of insider w.r.t the general attacker. Particular problems were foreseen with large diagrams and the lack of a clear definition of the concept of vulnerability.

The general discussion moved away from notation and directly to the heart of the security problems which the proposed notation intended to model. The usefulness of extended misuse cases for the purposes of *predicting* attacks was also discussed. "Inviting" attackers and insiders as (!) stakeholders during elicitation meeting was mentioned – and in fact, this is what happens with former hackers establishing their own security consultancy firms: an approach which seems to be successful in industry (for example, producers of anti-virus software have hired former virus authors).

The second paper, **A Comparative Evaluation of Three Approaches to Specifying Security Requirements** was authored by Mamadou H. Diallo, Jose Romero-Mariona, Susan Elliott Sim and Debra J. Richardson. The presenter started by observing that there has been an exponential increase in threats and vulnerabilities in software systems. Current practice often addresses the vulnerability *after* the system is built; the authors concentrate instead on addressing security in the requirements phase. In particular, the work reported concerns the comparative evaluation of three different approaches to specifying security requirements: Common Criteria (CC) which capture the environment, the security objectives and the derived requirements, Misuse Cases (MC) which focus on threats and attackers, and attack trees (AT), which capture attacks in a hierarchical tree structure. The three approaches were compared on the basis of a common case study: a wireless hotspot system in a public place. Five typical attacks were considered (deauthentication/deassociation attack; power saving mode attack; time window attack; virtual carrier sense attack; "evil twin" attack), and each of them was modelled in the three techniques mentioned above (the three models were also shown). The results were compared w.r.t. five categories: Learnability, Usability, Solution inclusiveness, Clarity of Output and Analyzability. None of the techniques resulted to be a clear winner, hinting that combining them might increase the confidence in specifying security requirements. Of course, combining the techniques raises issues about the increased cost and how to guarantee consistency of the three models.

The authors are aware that the study is very subjective, being based only on their own experiences with modelling a single system; yet, the results are consistent with those reported in the literature, and can be expected to scale to real-life problems.

Discussants raised a few questions about the comparison method and whether the three approaches are really comparable or not. The example used did not concern software in particular, but a general system. The lack of guidance on *the situation in which* one of the approaches was preferable to the others was also mentioned as a weakness of the work. In particular, it may not be sensible to provide "use all three" as advice in real-life situations. A fault in the paper was corrected: CC was not developed by NIST, but only adapted. Finally, the evaluation criteria were chosen

arbitrarily, and may not be the most important ones in practice. However, research in comparing methods was in itself very welcome, and the authors were encouraged to continue in this line of work.

The authors replied that the goal of their work was to compare the methods (the work started in fact from a survey of the field of security requirements); the "troublesome" advice to combine all of them was offered just as a hint and a prospect for future work, and was not a product of this research. Discussion continued on related themes, e.g. how to integrate security requirements with functional requirements. More discussion ensued on the fact that there are a number of techniques for addressing security requirements in literature, but they are seldom used in practice.

Questions about how the criteria were measured (e.g., "learnability") were also raised from the audience, and more details provided by the presenter. The presenter also reported that they have not actually combined the three methods (as they suggest), so the costs of applying all of them together has not be determined (nor the difficulty of ensuring consistency).

Sjaak Brinkkemper suggested that metamodelling the three approaches and then comparing/mapping the resulting metamodels could be a more effective way of comparing the approaches. The theme of metamodelling also came up for the previous paper, so the issue was postponed to the general discussion.

The third paper **Analysing i* System Models for Dependability Problems: The Überlingen Accident** by Neil Maiden, Namid Kamdar and David Bush unfortunately could not be presented due to the presenter's illness.

The general session discussion was facilitated by Camille Salinesi, who started by presenting summaries of the two papers presented. Overall, four approaches to model security requirements were discussed; the question of *which one is good when* and *how to integrate them* are both important research issues.

Security by itself is becoming more and more important (e.g., in BASEL II), and yet adoption of methods is very sparse. The methods we are researching are not a silver bullet, yet the problems are a real challenge for industry – a wonderful opportunity to test these methods in the field. We also need to standardize on an *experimental standard* to use when evaluating methods. Finally, there is a strong need to *formalize* the concepts we are discussing (e.g., "threat", "vulnerability"), and the issue of how to *identify* security requirements was not addressed at all in the papers – despite being very important in practice.

The issue of formalization was discussed at some length; there was a general consensus that concepts like "attacker" or "threat" do not need a separate formalization – it is just a matter of changing the *utility function* used in traditional techniques, e.g. from "our goal is to keep the system running" to "our goal is to crash the system", and then apply completely standard elicitation and analysis techniques.

Both presenters were invited to contact companies (e.g., banks) and try to run an experiment in an industrial context. The issue of security clearance, though, could be a show-stopper as this cannot be provided by universities. Nevertheless, there are lots of industries with less critical security demands who are still potential victims of attacks, so cooperation with such industries should be easier than, say, banks or the military.

The importance of showing *some* benefit was also stressed. Universities can provide a case study report and hints for improvement, as well as defining a benchmark. In most cases, the costs of supporting an industry-academia cooperation

on such issues would be very small compared to the cost of the entire projects, and entirely bearable as long as there is some advantage to be gained.

Sjaak Brinkkemper also reported on the experiences his group has had with industrial cooperation on running case studies, and on the importance of speaking the partners' language. There are excellent examples of successful long-term cooperation between academic research groups and industry, so this is certainly feasible in the current situation – an improvement over the way industry received RE research (and academia received industrial problems and solutions) years ago. There was no shortage of negative reports, though, particularly with trying to cooperate on security issues (for example, Pete Sawyer reported about his failed attempts with railway companies in the UK). Others reported a long period of cooperation (1 year and a half) needed just to establish trust between the parties; once trust is established, work can proceed much more easily.

The extensional nature of case study research was also discussed, and compared with the academic inclination to intentional analysis (the “five minutes of genius”). Nazim Madhavji also reported on his experiences with a sizeable project (“on a scale we would have never imagined”).

Having been on both sides of the trench, Sjaak Brinkkemper told about its experiences, and advised that case studies should *never* be run on *confidential* or *controversial* issues. The point of view of the person in charge on the industrial side is simply “will this improve or ruin my career?”. For example, a paper on “design erosion” was rejected and not authorized by the industrial partner until the subject was changed to “design preservation” – with the same data. The research was in fact studying how a product developed by the industrial partner had degraded over time, something the partner certainly did not want to expose to potential customers.

Another issue was that of meta-evaluation; the way research in certain medical fields is reported (essentially, by filling up a publisher-provided form) was suggested as an interesting alternative for collecting experiences and building repositories of case studies in RE.

Session 2: Case Studies

The session was opened by the chair, Pete Sawyer. The paper **From Requirements Maintenance to Program Maintenance: Using BPM Notation as the Missing Link** by Rafael Gutierrez, Joel Goy and Jean-Marc Latinus was a very good start for the conference (as Session 2 was actually held before Session 1) as it gave an illustrative example of RE practice in industry. It reported a 3 year project involving all departments of the agency for French education in foreign countries. The main challenge was to keep requirements and programs evolving at the same pace. This was achieved by modelling the business processes according to a simple pattern (document reception, document process, decision making, decision notification) and mapping that to a business process platform.

The first discussant emphasized that this solution is new in its domain, but would not work for technical domains (which have more complicated processes). Another discussant emphasized that using business processes to ease maintenance is a novelty. It was seen as a contribution to have achieved a common language between business and IT. Research is stimulated to supply tools which keep business processes and code synchronized. However, the authors were requested to try a more rigorous case study.

During the discussion it became clear that in addition to the processes, the business rules are very important, because they also capture requirements relevant to several processes. Also, it was noticed that quality requirements were not so important here, since the architecture was fixed at some time and the main question was how to support the business processes.

The following paper **Requirements-oriented Problems While Architecting: An Empirical Study** by Remo Ferrari and Nazim H. Madhavji reported a student experiment. The idea was to improve the communication between RE and architecture through studying the problems software architects identify in requirements. 16 student teams (consisting of 4 members each) were requested to derive an architecture from a requirements document consisting of roughly 80 requirements. Problems were reported by them in weekly feedback sessions with their advisors and through email. 35% of the problems were requirements-oriented. This suggested the need for further analysis. The requirements-oriented problems were classified by 5 experts into problems related to identifying the key architectural drivers (15%), problems with the different levels of abstraction of the requirements (14%), with deciding whether an architectural solution would satisfy the quality requirements (21%) and with understanding specific requirements (17%). Many quality requirements problems related to performance and availability, maybe because they are not as tangible as e.g. security requirements which are often functional. Scenarios could help here to improve documentation. Some of the teams had RE backgrounds, while others didn't. However, there were no statistically significant differences between the performances of the different teams. The authors acknowledged difficulties in generalizing the results to industrial teams; similar experiences however are difficult to replicate in industry (especially the RE-vs-non-RE test).

Discussants highlighted the positive contribution of the paper, and especially the way it stressed the importance of improving RE to better inform architecture. It was emphasized later by the presenter that this does not mean that there should be more RE upfront, but rather that it has to be performed in parallel. It was pointed out that an interesting additional study would be the investigation of the *sources* of requirements-oriented problems experienced while architecting. This point was not addressed in the paper (how to "fix" the problems was not addressed, either), and could constitute a good improvement. Also, the difficulty in generalization was unanimously considered a major factor challenging the validity of the results in industrial contexts.

In reply, the author explained that such a study would be designed differently in industry w.r.t. the way it was designed in an academic environment. With students, the cost factor of having 16 teams is manageable, while in industry it would not. Academic cases can still serve to assess the *cost factor* (which costs can be saved in industry by applying "good" techniques), and this cost factor could serve as a way to convince a company to run a single case study. The results would then either confirm the previous findings or reject them; this could be repeated with several companies to get a larger base.

The paper discussion opened on the details of the study. It was clarified that architects and modellers were the same persons in the teams, and that there was no special emphasis placed on quality while architecting. More information on the experiment design and the way it was conducted were requested by the audience, and provided by the presenter. It was a multi-case (16 team) exploratory (no a-priori hypothesis to validate), convenience (no way to ensure accurate sampling) study of

architecting banking applications. All the teams architected the same system, using the ADD method; architectures were documented and process data captured through defined templates. Help was provided on a request basis to teams during the process. The teams had an unlimited amount of time to accomplish the task, and there was neither direct cooperation nor competition across the teams. Overall, the design experiment was *very* rigorous (every recorded session was fully transcribed and analysed systematically, classification of problems was done by 5 people etc.).

Also, the relation between very high-level requirements (strategic goals) and architecture was briefly discussed. No specific analysis was run on how the particular input artefacts (the ADD method was given, the application was given, the framework supporting the architecture was given, etc.) influenced the results and the problems which were identified. This opened the way to threats to validity: for example, having used ADD may have simplified certain problems and made other worse; the results of the study may depend on the specific inputs and methods.

The last paper in the session **A Product Software Knowledge Infrastructure for Situational Capability Maturation: Vision and Case Studies in Product Management** by Inge van de Weerd, Johan Versendaal and Sjaak Brinkkemper focused on industrial RE again. The general idea was to present methodical guidance to improve RE processes in terms of a product software knowledge infrastructure (PSKI). This infrastructure provides a method base from experiences and allows the analyst to search for methods to apply in certain situations. The general idea of incremental RE process evolution (without a tool) was tried in 2 companies over a period of 1-2 years focusing on release management. In one company 24 employees were in charge of a software product with a rate of 30-50 new requirements each month (from 600 customers). The method for release planning should be improved. A capability matrix was used to identify new process capabilities (in this case prioritization) and then a specific method was suggested.

The first discussant appreciated the possibility of using the proposed roadmap in his own project. Nevertheless, from an industrial point of view the long duration of the process can be a challenge – not all organizations can afford to spend two years on such a project. Also, the cost of comprehensive documentation can be too high for overworked people. Despite these shortcomings, the discussant was positive about the real-life applicability of the approach. The second discussant also remarked on the good potential synergy between the papers in the session. The CMM-like role of the contribution was highlighted, and the associated practical wisdom praised as potentially very useful. The choice of the specific maturity levels was questioned. Further questions were raised on the progression of tables (relationships between quantification [on projects] and classification [on companies], usefulness of the matrix in Table 2 [since almost every box is checked at every level], etc.). Also, the research procedures and description seemed to be under-developed. The contribution was certainly useful as an initial approach, and the authors were encouraged to provide better validation and further develop their ideas.

The authors replied to many of the observations. They stressed that the improvement feedback in general will not take 2 years. They agreed to the observations about the usability of the matrix, but remarked that this is considered only an entry point to the improvement process, and more detailed information will be provided on a specific level.

During the general discussion, the comparison between the proposed model and the way SEI has introduced CMM were considered. In particular, CMM could draw on a

large repository of published literature, and could thus motivate the model based on observations. On the other hand, the presented work seems mostly given "a priori". The interested shown by the industrial attendees was considered a good indication of the potential of the work.

Another issue which was discussed was the relationship between method fragments, PSKI and process patterns. PSKI has the concept of "situational constraint" which could be used to characterize applicability of process patterns.

The general session discussion, facilitated by Patrick Heymans, opened with the question "when to do what?", which seemed to be the general theme of the three presentations. Two positions were posed in the papers: "Better have all requirements fixed so that architecting is less painful" and "Intertwine requirements and architecture". The third alternative, having architecture before doing RE, should also be considered for completeness. In fact, the paper "Why is RE for web applications easier" in an earlier REFSQ had this position. In view of the subject of another session, the question "why is it so difficult to deal with quality requirements" was raised. One possible answer is: they are fuzzy/cross-cutting/different/too many/too solution-dependent...

A call was also made to the RE community to start collecting results of empirical studies and experiences as "situational method fragments", as done in the last paper, and start building a common method base. The first question would be: how should this repository be structured? What should its contents be? A study of these subjects would produce a meta-model for process fragments.

The audience took the first theme first. Sjaak Brinkkemper pointed out that architecture is mostly frozen at "release 1.0", so in product lines development it can be assumed that most requirements work will happen with a stable architecture in mind (of course, problems with that architecture would make for difficult RE in the future). Nazim Madhavji observed that traditional architecture could/should be replaced with "strategy". Indeed, we are moving towards open architectures, whose components can be changed at will. The widespread adoption of components and services is changing our very concept of "architecture", and its relations with RE.

Vincenzo Gervasi observed that the richer your basic framework (e.g., web applications, SOA, etc.), the easier RE becomes, due to the reduced solution space to explore, and the more important it is to get the "right" requirements for that particular architecture. The trade-offs between support offered by and flexibility of a given architecture were also discussed considering the business processes that the application running on a given architecture should support.

Pete Sawyer reported on a workshop on ubiquitous computing that he attended recently, where the relationship between requirements and emerging (ubiquitous) computing was discussed at length.

Session 3: Quality of Requirements.

In contrast to Session 1's topic of "Requirements for Software Qualities", this session, chaired by Barbara Paech, was about analyzing and assuring the quality of the requirements themselves. The session comprised two papers addressing contrasting but complementary problems. The first of these, **A New Quality Model for Natural Language Requirements Specifications** was presented by Antonio Bucchiarone (co-authors Dan Berry, Stefania Gnesi, Giuseppe Lami and Gianluca Trentanni) and addressed the evaluation of the linguistic quality of requirements written in natural

language (NL). The motivation for this work is that linguistic defects can render the meaning of a requirement ambiguous and hence risk erroneous interpretation of the real intention of the requirement.

The work represents an incremental extension to the quality model (QM1) that forms the basis of the QuARS tool. QM1 is a member of the linguistic (as opposed to statistical) family of NL models of RE that exploits lexical, syntactic and structural properties of NL to infer meaning. QM2 extends QM1 to include knowledge of *ambiguity indicators* identified in recent work by some of the authors and their collaborators. The authors' next step is to extend QuARS to support QM2. This will permit QuARS to detect (e.g.) implicity, optionality, subjectivity or vagueness, any one of which may result in an ambiguous requirement. QuARS may even be able to prompt the author to select from a set of possible alternate interpretations. QuARS would still be imperfect, because of the sheer complexity of NL, but the work demonstrates that the model is extensible and scalable.

Discussants noted the incremental nature of the work, but appreciated the extension to ambiguity, remarking that building a significant repertory of indicators, while being an unexciting, extensional work in itself, will contribute to a better understanding of ambiguity. However, lack of any validation and the arbitrary choice of indicators were cited as weaknesses in the work.

The presenter replied by noting that the extension of QM1 was originated by feedback obtained from industry, so there was some form of a-priori validation. QuARS, implementing QM1, was used in real-life case studies, and although the number of false positives was very high, it was judged positively overall. The extension to QM2 will use the same technology, and there are no reasons to assume it would not work in the same way.

Questions from the floor were focusing on two linguistic issues: how much (linguistic) ambiguity is language-specific (e.g., what it would take to extend QM2 to – for example – Italian), and how effective it is to track down ambiguity on a purely lexical-syntactic level, i.e. without considering semantics. It was noted that sometimes ambiguity is a wanted feature. The scalability of the approach to very large documents also raised some concern. More technical details were then discussed about QuARS and the way it would implement QM2.

The second paper, **Sequence Constraint Consistency Checking in Requirement Specifications**, which was presented by Manu De Backer (co-authors Monique Snoeck, Wilfried Lemahieu, Guido Dedene), addressed the quality of behavioral models. This work was motivated by the need for consistency in sequence constraint specification. In particular, domain models need to be consistent with business process models both *vertically* (where there are dependency relationships between the domain and business process models) and *horizontally* (where different process actors' business processes must be consistent within the domain), which introduce two different types of consistency. The concept of *strong semantic compatibility* was defined and taken to mean that two business process models define the same behavior over the same set of domain events. An interesting problem with the definition of strong semantic compatibility is that two actors can still collaborate successfully even if one partner's behavior is not identical to what the other partner expects. To handle this, a more relaxed definition, that of *weak semantic compatibility*, was introduced. Here, all that is required is that at least one among the processes of an actor is supported by one partner processes. The authors are developing a technique based on

translation of business models to Petri nets to verify these classes of behavioral consistency.

There was some skepticism among the audience that a formal technique would ever prove acceptable to the industrial practitioners the work is aimed at. However, Sjaak Brinkkemper immediately refuted this by showing a Petri net model of a Baan business process, developed by and for industry, that he just happened to have on his laptop! Further defense for the approach came when it was observed that the plan was for Petri nets to be used as an analysis tool that the process stakeholders would never need to see or understand. The issue of how to trace back from inconsistencies identified in Petri nets to user-significant elements that can be shown to the user was also discussed.

Andrea Herrmann's summary of the session teased out the contrasting foci of the papers on the essential pragmatics of dealing with NL specifications versus the targeted use of formality to model complex behavior. The subsequent general discussion first considered the differing fields of applications of NL and formal or semi-formal notations. Some members of the audience noted that NL (and UML) was the prevalent choice on information systems, and formal methods (FM) were gaining acceptance for real time, safety critical systems. Others countered that the difference between the two approaches was mostly based on economic considerations (particularly, on the relative ROI for NL and FMs), not on the class of system.

The "good side" of ambiguity and inconsistency was also discussed: ambiguity can be a means to express abstraction linguistically, and most people agree now that we have to "live with inconsistency". Ambiguity is actually always present – for a good purpose – in calls for tender, and often in product line descriptions (e.g., to support variability). Ambiguity can also be a very useful tool in reaching agreement in negotiations. This didn't negate the value of being able to identify ambiguity where it is present, however. The discussion, and the day, were concluded with the slogan "defects have to be *managed*, not necessarily eradicated".

Session 4: Formal Methods

Day 2 began with a session on Formal Methods which comprised a single paper: **Design Exploration and Experimental Validation of Abstract Requirements** by Roozbeh Farahbod, Vincenzo Gervasi, Uwe Glässer and Mashaal Memon. Vincenzo began with a brief tutorial on abstract state machines (ASMs). Following this, Vincenzo presented the research goal which is to investigate what it takes to make ASMs usable for RE. The answer that has emerged form the authors' work is that they permit the design of languages and tools for domain-specific specification and high-level design. In particular, since ASMs can be interpreted or even compiled, they permit specifications expressed in an ASM to be executed, hence supporting the validation of specifications. This was characterized by Vincenzo as model-based engineering of abstract requirements.

However, most existing ASMs are targeted at detailed specifications. CoreASM, by contrast, seeks to make ASM techniques available earlier in the development cycle. To achieve this, CoreASM supports the writing of behavioral specifications by writing pseudo-code over abstract data. The CoreASM language is untyped, although types *can* be used and will be checked. Non-determinism can be enforced and multi-agent behavior can be expressed. These features make the writing of 'quick and dirty' specifications feasible, and hence requirements can be validated relatively easily and

cheaply. Work on CoreASM is still needed to reduce the cost of encoding specifications and this is being tackled by using plug-ins to provide domain-specific sub-languages.

The subsequent discussion raised three crucial issues. The first was the question of whether ASMs could tolerate the presence of ambiguity in the early phases of RE. CoreASM can use its abstraction mechanisms to represent ambiguity by tagging concepts as *unresolved*. The second issue was raised by a delegate from industry who questioned ASM's ability to acquire industrial acceptance unless they had a graphical representation. It appears that there is no satisfactory graphical representation of a full ASM. However, Vincenzo's envisioned users of CoreASM are engineers validating requirements rather than analysts engaged in elicitation sessions with stakeholders. He felt that the programmatic nature of CoreASM was likely to prove acceptable to this target user group. The final issue was the desirability of supporting an incremental transition to implementation. CoreASM's goes some way to achieving this but the challenges posed to writing a full compiler mean that it is better suited to prototype generation.

A number of unexpected factors meant that there was a vacant slot in the programme following the Formal Methods session. The slot was filled by a presentation of **Benefit Estimation of Requirements Based on a Utility Function** by Andrea Herrmann and Barbara Paech. This paper was accepted as a "maturing research" paper, and was thus a revised and considerably shortened version of the submitted paper. Instead of providing a solution to benefit estimation it used the notion of *utility function* borrowed from economics to explain several phenomena related to estimation. One can define the utility of a requirement only relative to a reference system: this explains the fact that often, dissatisfaction with non-realization of a requirement is not the opposite of satisfaction with realization of a requirement. Often, different reference systems are used to define satisfaction and dissatisfaction. Furthermore, this explains why it is not correct in general to just sum up the benefit estimates of individual requirements. Adding one requirement after the other changes the reference system incrementally. Of course, it is not practical to estimate the benefit for each requirement with respect to many reference systems. Thus, one has to make simplifications such as collecting groups of interdependent requirements that are related to *features*. The benefit estimations of different features are then largely independent of each other and can be summed up to obtain a good approximation of what the "real" utility would be. It was also illustrated that approximating utility function values by few discrete values (typically 1,2,3) bears the risk of compounding estimation errors. This can be alleviated by using utility intervals instead of fixed values. The authors were aware that these were just initial thoughts (as reflected by the "maturing research" tag), and further research is needed in developing reliable estimation methods.

The discussion remarked that these thoughts are also helpful for the estimation of requirement attributes besides utility, such as cost. Furthermore, it was remarked that such a utility function can be scrutinized through sensitivity analysis to determine how sensitive a given total estimation is to errors in the individual estimates. It was also pointed out that the question of benefit estimation is really important for industry, and that just using priorities is too simplistic. Benefit should be related to business goals and estimations should be based on data from experience. One could also use a multi-criteria approach considering different business goal dimensions. Thus, it was

agreed that the authors should strive to develop a more comprehensive method based on their current thoughts.

Session 5: Elicitation

Three papers had been selected for the elicitation session but unfortunately the curse of REFSQ'06 had struck again and the presenter of **Creating a Best fit between Business Strategy and Web Services Capabilities using Problem Frames Modeling Approach** by Anju Jha, Karl Cox and Keith T. Phalp was ill and unable to attend. Fortunately, the presenters of the other two papers were in good health, although even they were shivering in the room's icy, air-conditioned temperature.

The session kicked off with **A Requirement-driven Approach for Designing Data Warehouses** by Ines Gam and Camille Salinesi. This work was motivated by the fact that few data warehouse projects consider high-level requirements. They are driven by the data modeling needs, but typically fail to adequately consider the needs of users. Since the users include key decision-makers who need to make decisions that are informed by the data, this is a serious shortcoming. Ines described their new method, *CADWA*, that integrates modern elicitation practice and model reuse to guide data warehouse analysts in the design of data warehouses that satisfy the needs of the key-decision-makers. An example based on a study of a large retail organization illustrated CADWA's derivation of data models in a process that starts by identifying decision-makers' goals and proceeds with a series of refinements of business models until the decision-makers' key data requirements can be understood and modeled.

Ines noted that the method needed to be formalized and this was reinforced by the discussants who felt that, as presented, the refinement process used by CADWA was unclear. The discussants also speculated about how easily data warehouse analysts would adapt to the application of RE in their domain.

The presentation of **The Role of Creativity in Achieving High Quality Requirements Ideas: Classifying Requirement Ideas Generated for Web Applications According to a Quality Model** by Luisa Mich, Mariangela Franch and Daniel M. Berry created a sense of déjà vu in several members of the audience, since Luisa began by reminding us of work presented at REFSQ'04. This earlier paper had described some of the authors' work on *EPM create*, a creativity-fostering technique (CFT) for requirements elicitation whose goal is to generate novel requirements. The focus of the authors' REFSQ'06 paper, by contrast was on measuring the quality of the requirements of web sites that can be generated by the application of CFTs. This typically needs domain expertise and is hence expensive. The authors are investigating the extent to which this cost can be reduced by distinguishing between quality dimensions that are dependent on the semantic and syntactic properties of web sites.

To investigate this, the authors' have developed the *7loci* meta model of web site quality. 7loci classifies website quality according to the seven loci of Ciceronian rhetoric. These equate to questions about the identity, content, services, location, maintenance, usability and feasibility of web site requirements. The authors had re-analyzed the data collected from the industrial case studies reported in their REFSQ'04 paper to evaluate 7loci's effectiveness in classifying the quality of the requirements gathered. The interpretations of the results must be tempered by the limitations imposed by the scale of case studies, but appear to indicate that web site qualities that derive from syntactic dimensions can be evaluated effectively and

cheaply using 7loci, although the costs remain high for those that derive from the semantic dimensions.

Two key issues emerged in the discussion. The first was due to an assertion that the real key to the acceptance of CFTs was ensuring that only good requirements were generated. Luisa refuted this by pointing out that creativity was fundamentally dependent on the participants being free to generate ideas without being constrained by feasibility or other quality issues. The second issue segued into the general session discussion and returned to the theme of using real industrial data in case studies and experiments. This arose from an objection to the authors' use of data that had already been analyzed and published; an alleged form of self-plagiarism. This data had been extracted from industrial case studies and had proven, as it nearly always does, very costly to acquire. The cost of acquiring new data was therefore felt to be unreasonable when existing data was available and was in any case being mined for different information. The session concluded with Cornelius Ncube expressing enthusiasm for applying 7loci to the results of the creativity workshops run by City University.

Session 6: Complex Systems

The first paper in this session **GOMOSCE: Scenario-driven Goal Modeling for Complex Systems** by Cornelius Ncube and Neil Maiden described a process to do trade-off analysis and make choices between architectures. GOMOSCE is divided into 3 stages. At the first, i*-models are developed. These are then detailed by scenarios. Finally, walkthroughs (via Visio) are used in further elicitation meetings to suggest candidate actors, goals, soft goals, tasks, resources and actor dependencies. GOMOSCE (GOal MOdelling with SCEnarios) integrates scenario-driven requirements discovery (in ART-SCENE) and goal modeling of i* (REDEPEND). As for other research efforts by the same group, the process is fully tooled – generation and compilation of scenarios, presentations of scenarios, etc. are supported by several tools (mostly from the ART-SCENE and REDEPEND projects). Previously, scenarios and goal models from the two approaches were separated, and the respective tools were not collaborating. GOMOSCE bridges this gap and integrates the two in a single process. An example from a military domain was presented, showing how the tools work.

Discussants were generally positive about the idea of linking scenarios and goal models. Moreover, the relationships between these two and the decision-making process leading to the choice of an architecture were mentioned as interesting aspects. Nevertheless, the lack of evaluation was considered a drawback, and some discussant suggested that too much emphasis was placed on the details of how the tool worked (almost at the level of a user manual) and too little on the usefulness and applications of the techniques.

The unwieldiness of i* models was also cited as a difficulty by several discussants; large models tend to be difficult to follow, and having to browse them and interact with each node in order to get to the corresponding scenario appeared to be a weak point.

The author agreed with the discussant observations (in fact, he expected them). Lots of background is needed to properly appreciate the contribution, which could not fit in the paper. Also, the scalability problems with i* are known and recognized. Improvements to the notation (by adding proper support for layering, hierarchical structure and multi-resolution specifications) were considered desirable.

Attendants mentioned the results of the CREWS project, which conducted a survey of 200 companies about the way they were using use cases. Other questions were posed at the goal discovery stage of GOMOSCE: it was observed that goals should drive development, not be discovered. What GOMOSCE calls “goals” are actually activities, and the whole spirit of the work was reminiscent of results in the field of data flow analysis from the '70s (under different names). The presenter didn't recognize the parallel with data flows, but remarked that subgoals and activities are indeed related. Goal dependency and activity delegation, for example, are very similar phenomena; resources needed by a goal are the same resources consumed by an activity, etc.

There was still disagreement about the exact correspondence: the RE process through which goals are identified, and the semantic significance of goals, are different from those of activities. The animated discussion had to be postponed to the general session discussion.

The second paper **Dealing with “Map Shock”: A Systematic Approach for Managing Complexity in Requirements Analysis**, by Daniel L. Moody, was about the problems customers face when shown overly complex, terribly intricate RE notations. Even UML, with its 13 types of diagram and dozens of symbols, is a significant challenge to most observers. Analysts appear to be particularly fond of intricate diagrams, especially if they can fit *everything* in a single diagram by using minuscule icons and unreadable labels. Software is indeed one of the most complex artifacts ever created, and the complexity is somehow entrenched in the domain. Yet, a number of techniques can be used to simplify and humanize the notation. RE is also different from other disciplines: engineers, for example, show engineering diagrams to other engineers, whereas analysts are supposed to show their diagrams to customers, not only to other analysts.

The psychological phenomenon known as “map shock” has been studied in cognitive psychology, and a number of studies have reported on various facets of the problem. For example it is well-known that not more than 20 elements should fit on a sheet of A4 paper, and that not more than 7±2 elements can be recognized in one step. This, together with e.g. insights from cartography suggests several guidelines and organizing principles. A set of 9 principles (e.g. decomposition, summarization, redundancy, signposting, indexing, navigational map) were discussed, and some exemplified showing their application to an RE notation (a variant of data modeling). The paper has the full list of these. Applications of the same principles to other fields (social networks) were also briefly mentioned.

Discussants generally appreciated the paper, albeit the reported principles were not really novel by themselves. Their application in RE, however, would be very welcome, and certainly useful in improving the industrial acceptance and scalability of other techniques. The need for tool support for some of the principles was also mentioned – for example, when the same information is repeated in several diagrams to improve context awareness.

The presenter replied that indeed the principles were imported from a wide variety of other disciplines, yet there is too little attention paid to the results of other disciplines in RE. Points about application to i* and tool support were agreed, too.

Camille Salinesi expressed his intention of running a benchmark of different RE notations, and was interested in using the given principles as measures of quality of a notation. The distinction between ontological approaches to evaluation of notation (what a notation can express) and its understandability was also highlighted. Other

notational devices (e.g., the use of colors) were also proposed as hints at further research. Again, the analogy with cartography came up – users nowadays expect maps to use colors to convey meaning and manage complexity of representations. One participant from industry remarked that – in spite of the usefulness of these principles – one should also strive to educate users to read diagrams. There was no agreement whether this is feasible or even desirable. In any case there will be newcomers for whom the adherence to these principles (and thus the improved understandability of the models) are of utmost importance.

The last paper **Requirements Engineering in the Automotive Development: Known Problems in a New Complexity** by Martin Becker, Carsten Böckmann, Erik Kamsties and Thomas Wierczoch reported on the authors' position on challenges of RE in the automotive context.

First, a description of the field of RE in automotive development was given. It mostly concerns the development of Electronic Control Units (ECUs), of which modern cars have around 50 (steadily increasing year on year). In this domain, there are vehicle-level requirements (customer requirements are determined by marketing, whereas high-level technical concepts come from technical development, and vehicle functions are defined by function experts), electric/electronic component level requirements (defined by function experts, and expressed also in terms of voltage, times, etc.) and finally software requirements.

The ecosystem for automotive development is populated by original equipment manufacturers (OEM) and suppliers. OEMs provide vehicle level specifications, subsystem design and system integration, whereas suppliers develop ECUs and are responsible for specific components.

The position put forward in the paper is that specific challenges are not being adequately addressed by research. In particular, handling requirements of different types and levels of abstraction are not well-supported by any known technique; complexity and variance in the product portfolio means that growth of problems is outpacing the scalability range of many methods; the final market is heavily segmented, with each customer group having different requirements, and each member of a product line (i.e., each variant of each model of a car) must satisfy the requirements of a specific subgroup. Moreover, OEM-supplier collaboration is a continuous challenge, with OEMs specifying ECUs and suppliers developing them. The lack of a common presentation form for passing information back and forth between the two is a hindrance to the development effort.

In summary, the following challenges were identified:

- definition of practical requirement abstraction levels
- development of requirement information models
- concepts to connect component product lines with vehicle product lines
- collaboration structures for OEMs and suppliers

The discussants identified abstraction, variability and product lines, and OEM-supplier collaboration as important topics; yet, how portable the concerns of automotive development were to other areas was unclear. Nevertheless, while problems may be different, solutions developed to address those may well be reformulated for other applications. The potential of triggering new research was also indicated as a major contribution of the work. Some of the claims in the paper about the complexity of cars was contested by claims that there are many other more complex systems around, but the fundamental conclusions were judged valid.

The audience was also interested in knowing how large and pervasive abstraction problems were, since from other experience, it was noted that industry often prefers flat, extensional models to hierarchical, intentional ones.

The general discussion was facilitated – or better provoked – by Nazim Madhavij. He noted that two papers, GOMOSCE and the automotive one, revolved around domain complexity, whereas Moody's paper was about presentation complexity. He suggested that tools will only be used if they can be proven to be effective, which is not obvious; that scalability to really large systems is not to be taken for granted (it would be like using carpenter's tools for building a skyscraper), and that evolutionary approaches may never reach the "right" solution, simply oscillating from version to version around it. He illustrated this with a diagram reproduced in Figure 2.

Nazim's proposal for what we should do in our research is:

1. think big
2. know our limits
3. plan for tomorrow.

The point of "how large can it grow?" raised a lot of comments. One participant suggested we should know when to stop growing, but (1) who should decide this? and (2) when all parties have a vested interest in growing, there is no practical way to stop growth.

Other attendants remarked that modularity may be the general technique and design principle that should be applied thoroughly. However, modularity is just a way of re-factoring complexity; even with perfectly modular systems, complexity increases with the size of the system, although less dramatically. The same goes for goal models – even when modularized, at a certain point complexity would still reach the "painful" threshold. These problems will not be solved by marginal improvement – some rather radical breakthrough would be needed, otherwise our general expectations about software should be lowered.

The same kind of problems with hierarchical decomposition was reported on project management. On a project with over 650 people which one of the attendees had observed, there were managers, and super-managers, and super-super-managers overseeing various level of the project, with an all-seeing chief architect at the top. The key in managing complexity was not in adding more levels, but in separating management of requirements from actual requirements content – an idea which proved successful in practice whereas previous attempts based on decomposition had failed.

As a final remark, the need for hand-in-hand collaboration with industry was once again indicated as *the* way of researching such issues.

Panel: "Requirements Engineering and Project Management"

The panel session on Requirements Engineering and Project Management was introduced by Andrea Herrmann, who linked the panel to the REProMan workshop held with RE'05 in Paris. The panelists were Barbara Paech and Sjaak Brinkkemper.

Barbara started by commenting on the situation "at it used to be", with a process model which saw requirements being produced by RE specialists and then flowing to design and project management (PM).

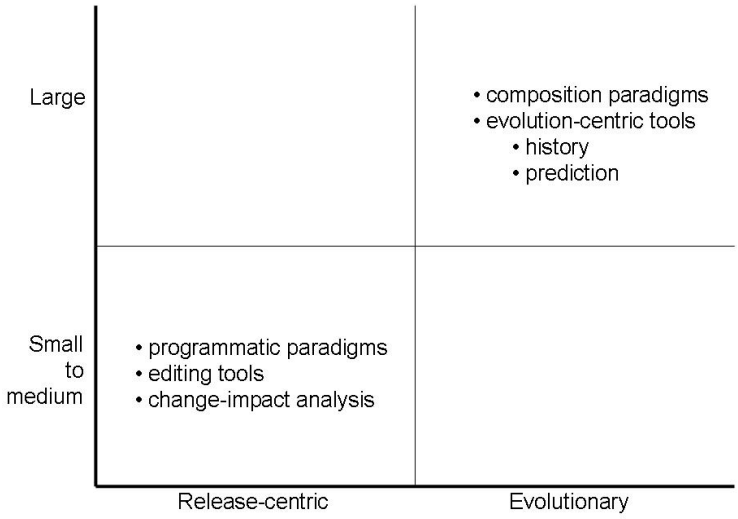

Figure 2: Size and Evolution Challenges

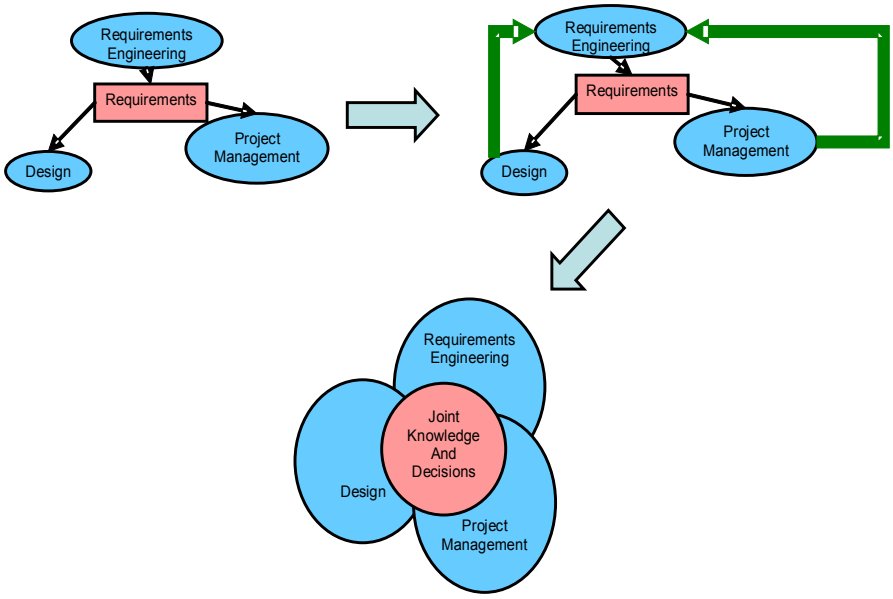

Figure 3: Integration of RE and PM

This view is now being replaced by an iterative approach, but essentially we should strive for an integrated approach where joint knowledge and decisions are taken at the intersection of RE, design and PM (Figure 3). In a sense, requirements (problem space), product (solution space) and process (management space) become a single space to explore while continuously re-evaluating or re-estimating benefits and risks.

In fact, the costs associated with requirements can only be determined knowing the general system architecture by which these requirements have to be satisfied, and in turn this is a factor in process management; on the other hand, management factors (i.e., schedule or personnel available) are a critical factor in deciding whether a given set of requirements is feasible or not; and which architecture can be realistically adopted may be influenced both by such managerial constraints and by which requirements are deemed more important.

The following topics were considered particularly relevant for the integration of RE and PM:

- PM: Process for RE: staffing, effort, schedule, stakeholder selection
- RE: Requirements benefits, risks
- Development: Architecture cost estimation
- PM: Prioritization
- PM: Process for the Project: Staffing, effort, schedule

Barbara reported on her experiences interacting with project managers from industry at the last REProMan workshop. In practice project managers often work as requirements engineers but without any specific guidance in RE. In terms of research, PM and RE research are almost totally separated.

The audience pointed out that PM is in some way more standardized than RE, and that by trying to "merge" RE and PM we could actually scare PM people out of cooperation. Barbara replied that Project Managers actually do a lot of RE in industry, so even if that is not immediately recognized, there are benefits to be reaped by cooperating. Sjaak reported that another major distinguishing factor is "who is in charge". In software-oriented companies, the most powerful figure is typically the software architect (projects aim at producing novel technology), whereas in other situations, the most powerful figure is the project manager (projects result in customer-oriented adaptation of some base technology). The issue of relative scarcity of resources (what is more important, technical or managerial skills?) was also mentioned as one of the factors conditioning the relative importance of RE and PM.

Sjaak's presentation revolved around interrelationships between RE and PM. His position was that PM is a *generic* activity at high level, and thus independent of RE as such; however, it is *specific* at the detail level, and thus *can* be integrated with RE.

Most activities (budgeting, scheduling, risk avoidance, etc.) are highly standardized, and applying them to software project is no different than applying them to any other project (e.g., naval construction).

In particular, Sjaak reported on a study he had conducted on the adoption of standards on a sample of 106 companies in the Netherlands. For System Development, new methods (RUP, SDM, etc.) are slowly *replacing* earlier standards which were widely deployed. For Architecture, there are few standards that are *emerging* (IAF, MDA, etc.), but not replacing any earlier standards – in fact, no older standard had gained significant acceptance. For Project Management, there are two standards (Prince/Prince2 and PMBOK) which are used by a large majority of the companies; Sjaak considered this as an indication that PM is a more *mature* field.

As a typical PM activity, Sjaak described in detail Release Planning, i.e. the decision about which subset of the requirements should be implemented in the next release of a product from a product family.

Requirements for product software

1. are numerous,
2. are difficult to prioritize,
3. come from different stakeholders.

Sjaak and his colleagues tackled this problem by using *Integer Linear Programming* (ILP). Various project estimates (costs, utility, capabilities) and constraints (domain, availability, deadlines, etc.) are formalized as ILP equations. The model considers such pragmatically relevant issues as vacations, market evolution, transfer of personnel from one team to another, learning and adaptation times, hiring temporary help, outsourcing, etc., as well as fundamental issues such as requirements interdependencies. Several formalizations based on different set of assumptions were presented: (1) with a single pool of interchangeable developers; (2) with multiple teams of specialized developers, and (3) with multiple teams with team transfers and "up to steam" times and losses of efficiency in case developers are transferred from one team to another to accommodate variable needs.

Once all the constraints are entered in an (off-the-shelf) ILP system, solutions can be sought, possibly maximizing a given utility function. Examples based on real data (from Baan) were shown. This was an example of how PM can be made RE-specific at the level of detailed release planning and resource assignment.

Sjaak's experiences (both working in and consulting for industry) were that solutions found in this way were vastly superior to hand-crafted ones, especially of initial assignment of resources.

Discussion moved to how sensitive these methods are, as extreme optimization may make solutions more sensitive to errors in estimation. It is a common case that companies want to catch technological or marketing waves (or hype!), and this leads to *critical choices*. In fact, it may be more convenient to choose a sub-optimal, but more robust/resilient, solution rather than the optimal one. Robustness and ill-conditioning may indeed be subjects for future research.

Other questions concerned how readable or understandable the output of the tool was to PMs. The question was also raised of how general the lessons learned from the release planning study were, and whether they could be extended to other aspects of the PM/RE liaison. There was a general consensus that the details are too situational to be generalized, and that the same happens in other industries. Politics, power structure, organization goals, etc. all interact in shaping the notion of "good" PM. In other areas (e.g., ERP projects), some of the assumptions of the ILP study simply do not hold: for example, the fact that the list of requirements is given at the beginning. In certain types of projects, scheduling is done *before* requirements are collected, and in fact requirements elicitation is in itself an activity which has to fit in the given schedule – which makes the approach inapplicable in these cases.

Returning to detail, it was suggested that "computed" (calculated based on varying market factors) estimates also offered a possible extension, and the human implications of using such techniques were discussed. Sjaak reported how PMs in companies showed high skepticism towards the method – especially because excessive reliance on technical advancements would put them out of the political decision-making process. So, politics can actually hinder the adoption of an optimal

solution because decision-making which is too transparent (being derived directly from raw data) leaves too little maneuvering space for power games.

In summary, the PM-related aspects of RE (or RE-related aspects of PM) look like a very promising area of research, which is currently not well developed, is waited for by industry (although with some political implication to take into account), and provides a number of challenging problems which deserve study.

The general floor discussion started on the important topic of whether there is more in the intersection between RE and PM in addition to release planning. The consensus was reached that there are certainly other applications, with the typical example being initial projects (which are to be managed even before elicitation, which is itself an RE activity). Even understanding the kind of information the PM will need in such a situation in order to make good, informed decisions about scheduling and resource allocation is in itself a significant challenge. Communication between REs and PMs was also cited as a major problem.

The issue of who takes decisions was discussed again, pointing out that it is difficult to definitely answer this question. Even in the same company, in the same process, for the development of the same product, significant decisions will be taken both by the RE and development side ("we are not ready yet, postpone release by ...") and by the PM and marketing side ("we cannot postpone any longer, drop the entire set of difficult requirements") – examples about the development cycle of Windows Vista, with its alternating decisions of postponing release to accommodate more functionality, and of dropping promised functionality to meet the released deadline, were noted.

In most companies, the social structure inside the company is skill-based, and decisions are taken by whoever is more competent about the issue at hand. In such cases, PM is seen just as a single area of competence. In other cases, the PM can be the team leader (or product manager), and take all overriding decisions. The different positions of PM and RE in the three cases can explain how certain decisions are taken. In general, there is no "right" organizational structure – it mostly depends on where the company gets the most value from, either from technology development or good project management (or good marketing, or...). It was also noted that agile processes like SCRUM change the environment for the interplay of RE and PM completely.

The discussion was concluded by participants exchanging their experiences with working for or in various companies of different sizes and structure.

5. Conclusion

Honouring its long tradition, REFSQ generated many discussions and good feedback. Several interesting and interconnected conclusions can be drawn from this year's papers. First, we observed a **comeback of quality features**, in particular product quality in the form of requirements for security and safety, and quality of the requirements themselves, particularly the issues of consistency and ambiguity. Second, **empirical work** emerged as a constant theme, particularly regarding the importance of working with real industrial data and the limits of lab-based experiments. Third, the **fitness** of different classes of methods for industrial applications was identified as an area requiring more work, and a number of issues of

urgent importance to industry emerged. Finally, **notational issues** stimulated much discussion, particularly, again, with respect to fitness for use by industry.

A general wrap-up discussion was held at the end of the conference, during which suggestions were sought for interesting themes to propose for REFSQ in 2007. The following themes were suggested by the audience:

- RE for large systems
- RE for evolving systems
- RE and architecture and conformance
- Handling multiple levels of requirements
- Interoperability between RE methods
- Benchmarks and standardised validation and research approaches
- Research methods for (e.g.) situational and domain-specific RE
- Communicating and documenting requirements

As always, the co-chairs asked for feedback on the conference and a number of perceptive and useful ideas were suggested. Further improvements of the new first slide were suggested. The panel session was considered successful but several people felt it was a pity that it had to be scheduled as a parallel session.

The 12th edition of the conference was closed with a big "thank you" from the co-chairs to all participants for a very successful REFSQ'06. In turn, participants generously thanked the co-chairs for their work. 2006 marked the final year of Vincenzo Gervasi's service on the organising committee. Much to his embarrassment, Vincenzo's many admirable qualities were enumerated and he was given a very warm send-off.

6. Paper-Keyword Mappings

In the following tables we show how papers were classified by their own authors according to the criteria presented in Section 2. In particular, Table 1 list the papers presented at REFSQ'07, together with their reference number. These numbers are then used in Tables 2-7 to classify papers according to the seven criteria *Paper Class*, *Process Area*, *Performer*, *Artefact*, *Technique*, and *Intended Audience.* Finally, Table 8 lists the custom keywords assigned to the papers by some author. In order to encourage focusing on the most important concepts in each paper, authors were limited to a maximum of 6 keywords per paper.

The reader can use these tables as an alternate form of indexing for the proceedings volume. For example, readers interested specifically in Problem Frames can flip to Table 5 (Techniques) and find out which papers discussed the issue; practitioners can turn to Table 7 (Intended Audience) to identify recommended readings, and so on.

The same data, sliced by sessions, is presented in graphical form at the beginning of each section of the volume.

Table 1: Paper Numbers

Paper #	Authors	Paper Title
1	Lillian_Røstad	An Extended Misuse Case Notation: Including Vulnerabilities and the Insider Threat
2	Mamadou H. Diallo, Jose Romero-Mariona, Susan Elliott Sim and Debra J. Richardson	A Comparative Evaluation of Three Approaches to Specifying Security Requirements
3	Neil Maiden, Namid Kamdar and David Bush	Analysing i* System Models for Dependability Properties: The Uberlingen Accident
4	Rafael Gutierrez, Joel Goy and Jean-Marc Latinus	From Requirements Maintenance to Program Maintenance
5	Remo Ferrari and Nazim H. Madhavji	Requirements-Oriented Problems While Architecting: A Case Study
6	Inge van de Weerd, Johan Versendaal and Sjaak Brinkkemper	A Product Software Knowledge Infrastructure for Situational Capability Maturation: Vision and Case Studies in Product Management
7	Antonio Bucchiarone, Dan Berry, Stefania Gnesi, Giuseppe Lami and Gianluca Trentanni	A New Quality Model for Natural Language Requirements Specifications
8	Manu De Backer, Monique Snoeck, Wilfried Lemahieu, Guido Dedene	Sequence Constraint Consistency Checking in Requirement Specifications
9	Roozbeh Farahbod, Vincenzo Gervasi, Uwe Glässer and Mashaal Memon	Design Exploration and Experimental Validation of Abstract Requirements
10	Anju Jha, Karl Cox, and Keith T. Phalp	Creating a Best fit between Business Strategy and Web Services Capabilities using Problem Frames Modeling approach
11	Ines Gam and Camille Salinesi	A Requirement-driven Approach for Designing Data Warehouses
12	Luisa Mich, Mariangela Franch and Daniel M. Berry	The Role of Creativity in Achieving High Quality Requirements Ideas: Classifying Requirement Ideas Generated for Web Applications According to a Quality Model
13	Cornelius Ncube and Neil Maiden	GOMOSCE: Scenario-Driven Goal Modelling for Complex Systems
14	Daniel L. Moody	Dealing with 'Map Shock': A Systematic Approach for Managing Complexity in Requirements Analysis
15	Martin Becker, Carsten Böckmann, Erik Kamsties and Thomas Wierczoch	Requirements Engineering in the Automotive Development: Known Problems in a New Complexity

Table 2: Paper class

Paper # / **Keywords**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Evaluation of existing situation		X			X							X			X
Proposal of a solution	X		X			X	X		X		X		X	X	
Validation of proposed solution										X					
Philosophy															
Personal experience				X											

Table 3: Process area

Paper # / Keywords	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Eliciting		X									X	X			
Modeling	X			X		X			X	X			X		
Analysing		X	X				X			X					
Communicating														X	X
Agreeing															
Evolving															
Prioritizing															
Reusing															X
Tracing															

Table 4: Performers

Paper # / Keywords	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Final user														X	
Customer														X	
Tester															
Developer	X				X										
Specifier				X					X				X		
Analyst			X				X					X		X	
Domain expert										X				X	X
Autonomic system															

Table 5: Techniques

Paper # **Keywords**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Goal-driven											X		X		
Problem frames										X					
Use cases	X														
Viewpoints															
Linguistics							X								
Process modeling				X		X									
Patterns															
Inspections															
Formal methods								X	X						
UML															
Ethnomethodology															

Table 6: Artefacts

Paper # **Keywords**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Domain knowledge sources				X											
Domain models								X		X					
Formal specifications									X						
Test cases															
Design											X		X		
Source code															
Large requirements bases															
Business needs											X				

Table 7: Audience

Paper # / Keywords	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Academics	X	X	X		X	X			X						
Students		X													
Practitioners	X	X		X	X								X		X
Regulators															
Business decision-makers															
Public policy makers															
General public															

Table 8: Custom Keywords

Paper # / Keywords	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Dependability requirements			X												
I* model analysis			X												
Software architecting					X										
Requirements-oriented problems					X										
Product management						X									
Method engineering						X									
NL-SRS							X								
Ambiguity							X								
Quality												X			
Web applications												X			

7. Acknowledgements

First of all, we want to thank Eric Dubois, Andreas Opdahl and Klaus Pohl who served on the REFSQ Advisory Board.

We would also like to thank the members of the program committee who acted as anonymous reviewers and provided valuable feedback to the authors: Ian Alexander, Thomas Alspaugh, Aybüke Aurum, Patrik Berander, Daniel Berry, Antonio Bucchiarone, Sjaak Brinkkemper, Pierre Jean Charrel, Karl Cox, Alan Davis, Eric Dubois, Anne Etien, Joy Garfield, Stefania Gnesi, Michael Goedicke, Tony Gorschek,

Patrick Heymans, Frank Houdek, Natalia Juristo, Erik Kamsties. John Krogstie, Giuseppe Lami, Peri Loucopoulos, Kalle Lyytinen, Neil Maiden, Raimundas Matulevicius, Johan Natt och Dag, Andreas Opdahl, Anne Persson, Klaus Pohl, Norah Power, Jolita Ralyté, Lucia Rapanotti, Björn Regnell, Colette Rolland, Matti Rossi, Camille Salinesi, Kristian Sandahl, Andres Silva, Guttorm Sindre, Maurice ter Beek, Mario Winter, Claes Wohlin, Eric Yu, Didar Zowghi. To their dedication all the readers of these proceedings should feel indebted.

8. References

[1] Wieringa, R., Maiden, N., Mead, N., Rolland, C.: A classification of RE papers: Are we researching or designing RE techniques?, Proc. *Third International Workshop on Comparative Evaluation in Requirements Engineering (CERE'05)*, Paris, 2005.

[2] Wieringa, R., Maiden, N., Mead, N., Rolland, C.: Requirements engineering paper classification and evaluation criteria: A proposal and a discussion, *Requirements Engineering,* 11 (1), 2006.