Requirenautics Quarterly

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

RE-Locations

RE-Soundings

Editorial

For those of you new to RQ – welcome!! As well as sending this out to our members, we're distributing this issue free at the Third International Conference on Requirements Engineering. If you're attending the conference, and have just picked this up, please take it away with you for later use. If you want more, you can join the RESG and have this newsletter sent to you four times a year - there's a membership form at the back. You'll also get discounted entry into RESG events, of which there are many throughout the year. If you have the misfortune of not living in Britain, don't despair – these events are worth travelling for! Oh, and don't forget to enjoy the conference!

Copy deadline for the next issue is 26th June 1998

Steve Easterbrook,
NASA IV&V Facility, Fairmont WV

Chairman's Message

I'm writing this as I prepare to travel to the 3rd International Conference on Requirements Engineering (ICRE-98) in Colorado Springs, where over half the technical papers are by authors working in UK organisations. I also note that all these authors' organisations are members of the BCS RESG. A coincidence? Nah..I'm sure there is a correlation! If

you're reading this and are not a member, why not complete the membership form at the end of this newsletter?

We're well into our programme of events for 1998. We started in York in February with an afternoon meeting on Industrial Experiences in RE, and followed this in March with a half-day mini-tutorial on the Unified Modelling Language. In May, we're moving up to a full-day event: a series of presentations and demos on the use of Scenarios in Requirements Engineering; while June see us teaming up with the IEE for a full-day colloquium on Managing Requirements Change - a business process re-engineering perspective.

Details of all these events are in this issue.

As always, please let us know if there are any topics you'd like us to cover in our programme of events. We aim to please!

Bashar Nuseibeh,
Imperial College, London

PS: This issue coincides with *that* time of the year: membership renewal. Please help us (Sara!) survive this process by returning your completed renewal forms promptly. We're offering a two-year membership this year to save time, effort and trees!

RE-Treats

Forthcoming events organised by the group.

How to Use Scenarios and Use Cases in the Systems Development Process

Thursday May14th, 1998
Location: Senate Suite, City University, London
Time: 0930-1630
Cost: £120+VAT per person
Format: A one-day symposium
(See Registration form on the next page)

Despite the considerable recent interest in the use of scenarios and use cases in the systems development process, there is a lack of scenario-based methods, techniques and guidelines available to practitioners, or even agreement about the definition of use cases and scenarios in the first place. Jacobson has done much to introduce use cases into the systems development process, but even he has left us with some loose ends. As a result, discussions about use cases and scenarios often lead to more questions than answers.

This symposium aims to provide some concrete answers. It brings together practitioners, vendors and academics with different interests in scenarios and use cases in the systems development process. Experts from the United States and Europe will present and explore the diverse uses of scenarios and use cases, examine effective methods and guidelines, report on experiences and good practice, and propose a common path for the development and application of new scenario-based systems development methods.

Who should attend? Practitioners, vendors and academics with interests in scenarios and use cases in the systems development process. In particular, you should attend if you use or want to know more about:

• complex operational scenarios to acquire requirements for large, integrated systems;
• how to use scenarios for acceptance testing of large, procured systems;
• the OBJECTORY method for object-oriented analysis and design;
• the UML approach for use cases to acquire and validate system requirements;
• task/scenario analysis for user interface design and evaluation.

Speakers at the symposium including leading consultants, practitioners and academics in requirements engineering, systems engineering, object-orientation and use case-driven approaches:
  Ian Graham, Chase-Manhattan Bank;
  Ron Krubeck, Rational Software Corporation;
  Ian Alexander, Independent Consultant;
  Andrew Daw, GEC-Marconi;
  Matthais Jarke, RWTH-Aachen;

  Eric Dubois, Universite de Namur;
  Colette Rolland, Universite de la Sorbonne, Paris;
  Alistair Sutcliffe, City University London.

It is also possible that Colin Potts from Georgia Institute of Technology will be able to speak at the seminar.

Panel discussions will give ample opportunity to discuss key issues. Software tool demonstrations will also be available throughout the day.

The symposium is limited to 100 delegates. Interest so far has been keen, so please register early to avoid disappointment. Information and registration details are available from:

Liz Bromley, Centre for HCI Design, City University, Northampton Square, London, EC1V OHB. Tel: 0171-477-8427. Fax: 0171-477-8859. <E.M.Bromley@city.ac.uk>

Registration will include a buffet lunch, light refreshments and a delegate pack. Registration is deemed to have been accepted on receipt of registration details. Substitute delegates may be made at any time. Registrations can be cancelled by providing written notification at least 10 working days before the symposium, when a full refund, less £23.50 administrative charge, will be made. Cancellations after this date are liable for the full fee.

Registration fees are:
  full registration: £120.00+VAT = £141.00
  BCS members 20% discount: £96.00+VAT = £112.80
  2nd delegate from same organisation: £96.00+VAT = £112.80
  Fulltime students 50% discount: £60.00+VAT = £70.50

Abstracts for the invited speakers follow:

Validating Requirements: Beyond UML

Ian Graham, Chase Manhattan Bank

UML is put forward as a standard for object-oriented development but is deficient in areas such as development process and requirements engineering. In the latter area problems arise principally because of the weak semantics of the class modelling language (bi-directional associations and lack of rulesets), lack of theory and non-OO semantics for use cases, lack of concurrency for sequence diagrams and too early use of the latter. This talk will show how all these problems can be addressed and go further to show how an object model can be validated as necessary and suffient against a set of business objectives.

Ian is a Vice President at the Chase Manhattan Bank with responsibility for object-oriented software development practices. His books include Object-Oriented Methods, Migrating to Object Technology and The OPEN Process Specification. Ian is a Fellow of the British Computer Society.

Scenarios and their Application for Requirements Definition or What Do You Want To Do with the System?

Andrew Daw, GEC-Marconi

The paper presents a framework for, and definition of, activities for Requirements Definition, in which Scenario Generation and Animation play a fundamental role. The paper discusses, within a Top Down Systems Engineering perspective, the need for operational contexts of the system (in these examples complex whole ship naval platforms) from which to answer the question - 'What Do You Want to Do with the System? The use of scenarios is shown to offer a consistent and coherent framework, not only for requirements elicitation but also for effectiveness and capability modelling, stimulation of structured functional design representations, and Human Factor MMI Design and Human Operational analyses. In such circumstances, the scenarios offer an immediate vehicle for answering the second half of this question - '...and How Well Do You Want it To Do It?'

The paper draws together these issues and perspectives, and discusses the distinct points of focus offered by Scenarios, whereby the operational context of the platform can be defined and analysed; the management of change in that context can be assessed; design trade offs can be identified and prioritised across multiple design paradigms; equipment comparisons can be achieved for cost effectiveness and performance analysis; and the definition of through life support models can be initiated through appropriate scenario definition and management.

Use Cases: A Reusable Way To Represent Requirements Information

Ronald L. Krubeck, Rational Software Corporation

Software requirement information has typically been specified via traditional techniques such as a Product Requirement Document and a Software Requirements Specification. Such techniques show requirements in a static, declarative way and offer few, if any, ways to promote reuse of these requirements by other functional groups such as Quality Assurance and Documentation. Use cases, used as a way to capture software requirements, promise greater reuse by other functional areas within an organization.

Ronald L. Krubeck is a Principal Engineer with the Requirements Management Business Unit in the Rational Software Corporation.

Task Models to Elicit Requirements and Generate Scenarios

Ian F. Alexander, Independent Consultant

Scenario Plus is a specially-designed tool for creating, editing, measuring, and exploring task models. A task model is a detailed analytic description of a task that might be carried out by a future system, or by a system together with

---

How to Use Scenarios and Use Cases in the Systems Development Process
Thursday May14th, 1998

REGISTRATION FORM

Please return the completed registration forms to Liz Bromley, Centre for HCI Design, City University, Northampton Square, London, EC1V OHB. Tel: 0171-477-8427. Fax: 0171-477-8859. E.M.Bromley@city.ac.uk

Full registration:	£141.00 incl VAT
BCS members:	£112.80 incl VAT
Second delegates:	£112.80 incl VAT
Fulltime students:	£70.50 incl VAT

Please attach duplicate registration forms if more than one delegate is attending.

I enclose a cheque for £_____

Cheques payable to BCS Requirements Engineering Specialist Group

Delegates will receive seminar pack, lunch and tea/coffee.

Dr/Mr/Ms_____   First name_____

Surname _____

Position_____

Organisation_____

Address _____

_____

_____

_____

Post code _____

Tel _____

Fax _____

Your purchase order no _____

Vegetarian lunch required: **yes/no**.

Special needs: _____

Please invoice: **yes/no**

Invoice address if different to above:

Department_____

Organisation_____

Address _____

_____

_____

Post code _____

Date _____

other agents including human users. Task models are visualised primarily as hierarchies of subtasks, but tasks can be combined in a variety of ways. A task can be given a type, such as Sequence, Alternatives or Parallel, indicating the relationship between its children. Looping is permitted by means of an explicit Periodic type, as well as a Link type which corresponds to an unconditional jump. Scenario Plus can animate task models, enabling users to step through planned tasks quickly or deliberately. During animation, the user is offered a choice when a set of Alternatives (etc) is encountered. Generation of a path is thus semi-automatic, enabling users to select just the paths of interest. Metrics indicating the complexity of the model, in terms of the minimum number of paths it can generate, can be displayed at any time. Any generated path through a model can be saved as a Scenario, for later use. Scenarios can be replayed, or used to filter the task model, displaying only the relevant tasks. These can form the basis of requirement documents, safety analyses, and test specifications. Scenario Plus runs on all DOORS PC and UNIX platforms.

CREWS: A European Project on Scenario Management

Matthias Jarke, RWTH Aachen, Germany

The European ESPRIT project CREWS (Cooperative Requirements Engineering With Scenarios) is concerned with systematic support for the usage and management of scenarios in the development of complex systems. The project has developed a classification scheme for scenario-based techniques which was then applied (a) to gain a structured overview of the state-of-the-art in the field, and (b) to structure a broad survey of current scenario practice in Europe. The talk summarizes the result of both studies and gives an overview of the work done in CREWS to remedy some of the identified issues.

Matthias Jarke is professor of Information Systems and chairman of the computer science department at Aachen University of Technology. Prior to joining Aachen in 1991, he served on the faculties of New York University's Stern School of Business and of the University of Passau, Germany. His research interests focus on information systems support for design processes in business and engineering. Related to requirements engineering, he has been coordinator of three European ESPRIT projects, DAIDA, NATURE, and CREWS. He is editor-in-chief of the journal Information Systems and served, in 1997, as program chair of the International Conference on Very Large Data Bases (Athens, Greece), and as General Chair of the German National Computer Science Conference.

Requirements Elicitation with Real World Scenes

Klaus Pohl, RWTH Aachen

Scenarios are an excellent means for eliciting and validating requirements. A scenario represents a concrete example of current or future system usage. We call a persistently recorded usage of the current system a real world scene (RWS). This talk presetsso called abstraction guides which

support the requirements-engineer in eliciting requirements from RWS and in validating requirements against RWS. During the elicitation and validation, the abstraction guides relate the conceptual definition of the requirements to the parts of the RWS which have caused their definition or which have been used to validate the requirement. Thereby a fine-grained interrelation between the conceptual models and the RWS is established. These interrelations significantly improve the traceability and understandability of conceptual models. The interrelation provides the basis for: explaining and illustrating conceptual models to, for example, new team members; comparing conceptual models defined by different stakeholders; comparing different observations using computed annotations; and refining or detailing a conceptual model during later process stages.

Klaus Pohl is senior researcher with the Information Systems group at Aachen University of Technology, where he also obtained his doctoral degree in 1995. In 1996 he was a visiting professor at the University of Namur, Belgium. Klaus Pohl is/was work-group leader in the ESPRIT Reactive Long Term Research project on cooperative requirements engineering with scenarios (CREWS) and in the ESPRIT Basic Research Action on novel approaches to theories underlying requirements engineering (NATURE). He is initiator of the international workshop series on Requirements Engineering: Foundations of Software Engineering (REFSQ), member of the editorial board of the Requirements Engineering Journal and the vice-chair of the GI requirements engineering interest group.

Guiding Goal Modelling Using Scenarios

Colette Rolland, Universite de la Sorbonne

Since a few years, scenario based requirements engineering approaches have gained in popularity. Textual scenarios are narrative descriptions of flows of actions between agents. They are often proposed to elicit, validate or document requirements. The CREWS experience has shown that the advantage of scenarios is their easiness of use, and that their disadvantage stands in the lack of guidelines for authoring. In this article, we propose guidance for the authoring of scenarios. The guided scenario authoring process is divided into two main stages : the writing of scenarios, and the correcting of scenarios. To guide the writing of scenarios, we provide style and contents guidelines referring to a conceptual and a linguistic model of scenarios. To guide the correcting of scenarios, we propose a set of enactable rules. These rules aim at the clarification, completion and conceptualisation of scenarios, and help the scenario author to improve his scenarios until acceptable quality in the terms of the former scenario models. The paper presents both style/contents guidelines, and correction rules, and illustrates them with the London Ambulance Service example.

Requirenautics Quarterly   Issue 14 (April 1998)

The Newsletter of the BCS Requirements Engineering SG   Page 5.

Scenario-Based Requirements Validation: The CREWS SAVRE Method and Tool

Alistair Sutcliffe, Centre for HCI Design, City University

The presentation will introduce the CREWS-SAVRE method and briefly review the tool demonstration that will be available. The method uses scenarios in two senses; examples or stories of use taken from the real world and threads of interaction generated from use cases. Different validation treatments are linked to the two scenarios types. First, real world scenarios are used as test data to check dependencies between input and output to the system from the environment. Requirements are elaborated to deal with normal and abnormal input while the impact of system output on different stakeholders is assessed. Secondly scenarios are generated from use cases by an interactive dialogue. Checklists of possible errors and their causes are used to suggest when user-system interaction may go wrong, and generic requirements are proposed to deal with these problems. The method, implemented in the SAVRE tool assists the requirements engineer to explore the possibilities of future system behaviour and then detects possible problems by validation frames. The frames act as pattern recognisers for different combination of agents and events which may give rise to obstacles to normal system operation. The tool uses an extensive database of problems that it matches to solutions (i.e. generic requirements) and iteratively builds use cases and requirements specifications documented in the Requisite Pro tool. Use of the method and tool will be illustrated with a banking case study.

Animation of Scenarios for Requirements Validation

Eric Dubois, Universite de Namur, Belgium

Albert II is a formal language designed for the purpose of expressing requirements inherent of distributed real-time systems. The language has been the subject of several technology transfer initiatives. In particular, it has been used by industrial partners in the context of the development of two large, distributed, software-intensive, heterogeneous systems (a video-on-demand application and a satellite-based telecommunication system.

We are currently developing a number of tools for supporting the use of the language as well as the verification of the produced specifications. At the validation level, checking the adequacy of a formal requirements specification towards the needs expressed by stakeholders is far from being a trivial issue. To overcome this problem, we work on a so-called animator that the stakeholders can use interactively and cooperatively in order to explore different possible behaviors (or instance-level scenarios) of the future system allowed by the formal requirements specification. The purpose of the tool comes down to test if a given scenario proposed by one or several stakeholders is compatible with the requirements specification.

Managing Requirements Change: a business process re-engineering perspective

Date : June 11th 1998
Format: Full day colloquium organised by IEE Professional Group A1 (Software Engineering) jointly with BCS Requirements Engineering Specialist Group (RESG)
Location: IEE, Savoy Place, London.

The requirements of software-intensive systems in organisations often change in response to advances in technology and changes in the business environment in which this technology is deployed. An understanding of business processes and the ability to adapt them are therefore crucial to the effective management of changing requirements. This colloquium, organised jointly by the IEE Professional Group A1 (Software Engineering) and the BCS Requirements Engineering Specialist Group (RESG), will explore problems, solutions and issues in managing requirements changes from a business process re-engineering (BPR) perspective. Speakers include both researchers and practitioners drawn from a variety of industrial and academic backgrounds.

Provisional Schedule

09:45 Registration & Coffee

10:15 Chairmen's Welcome & Introduction
       Prof. Peter Henderson (University of Southampton)
       Dr. Bashar Nuseibeh (Imperial College)

10:30 IT-Driven Business Change
       Lawrence Regan (Barclays Group Plc.)

11:00 Demanding Change: How to remain in business despite IT
       Richard Veryard (Ind. Consultant, Veryard Projects)

11:30 The role of process based IT in supporting business change
       Bob Snowdon (ICL)

12:00 Lunch

13:00 Transferring success in managing requirements change: patterns achievement
       Dr. Rob Pooley (University of Edinburgh)

13:30 Legacy System and Software Evolution at the Centre for Software Maintenance
       Dr. Malcom Munro (University of Durham)

14:00 Agile System Design and Build
       Prof Richard Weston (Loughborough University)

14:30 BPR Through SSM: An Incremental Approach
       Prof. David Bustard (University of Ulster)

15:00 Coffee Break

15:30 Complexity: Partial Support for BPR?
       Dr Eve Mitleton-Kelly (London School of Economics)

16:00 Panel Discussion

16:30 Close

Conference on European Industrial Requirements Engineering (CEIRE'98)

Date : 19th - 20th October 1998
Location: Novotel, Hammersmith, London.
Format: Two-day industrial conference, organised by the BCS RESG in association with RENOIR.

CEIRE'98 is the second (following RE Day, held in London on the 30th September 1997) in a series of European industry-oriented requirements engineering symposia organised by the BCS RESG in association with Renoir.

CEIRE'98 will provide a 2-day forum for the dissemination of experience and practice in RE between European software and systems engineering organisations and researchers. The programme will include tutorials and workshops on issues relevant to RE. It will also include a special track of papers giving the industrial perspective on RE. Papers are invited from industry or business organisations which focus on the practice of RE. Papers are particularly encouraged on, though not restricted to, the following:

* Examples of pressing RE problems faced by industry.

* Case studies of the application of particular RE techniques.

* Reports of good RE practice.

* Reports of the integration of RE into quality management and software process improvement programmes.

* Reports of issues concerning RE in specialist domains - e.g. dependable systems.

* Reports of how changes to the business or competitive environment impacts on RE.

Submission procedure:

Papers should be in English and of no more than 3000 words in length. They should be submitted by the 29th June. Authors will be notified of whether their paper has been accepted by the 10th August. Accepted papers will be published in the workshop proceedings.

Address for submissions:

Pete Sawyer
CEIRE'98 Industrial Programme Chair
Computing Department
Lancaster University
Lancaster
U.K. LA1 4YR
tel: 44 1524 593780
e-mail: Sawyer@comp.lancs.ac.uk

Programme committee:

Ian Alexander
Wolfgang Emmerich (coordinator)
Galal Galal (workshop organisation)
Orlena Gotel (exhibition organiser)
Sara Jones (tutorial organisation)
Gerald Kotonya
Neil Maiden (treasurer & local organisation)
Shailey Minocha
Cornelius Ncube
Bashar Nuseibeh (chair)
Pete Sawyer (industrial programme chair)
George Spanoudakis

Special industrial advisory committee:

Pere Botella
Sjaak Brinkkemper
Silvana Castano
Eric Dubois
George Grosz
Andreas Opdahl

Summary of important dates:

Conference dates: 19th - 20th October 1998
Submission deadline: 29th June 1998
Notification of acceptance: 10th August 1998

Distributed Requirements Engineering
November 98
Details to be announced

$\mathcal{RE}$-News

RESG Membership to Run for 2 years

After consultation, we have decided to increase the period after which your membership of RESG needs to be renewed from one year to two. Our aim is to reduce the administrative load, both on us, and on you! Details are enclosed with this issue.

New Journal

Starting early in 1999, Springer-Verlag will publish a new quarterly journal called "Knowledge and Information Systems: An International Journal." This journal has grown out of increasing demands of such a rapidly growing field in Asia, Europe, and North America.

Instructions for authors and the aims and scope of the journal can be found at *http://www.sd.monash.edu.au/kais/* Electronic submission is strongly encouraged, to expedite processing, and Springer-Verlag will offer the journal in both electronic as well as print form. The editors will emphasize minimal delay in processing and publication, and plan to review papers quickly and advise authors of their paper status with a target turnaround time of 3 months from submission (4 to 6 weeks for short papers).

SEA Workshop Postponed

In view of the impending launch of a new SEBPC-funded network (the Software Engineering and Information Systems Network), it has been decided to postpone the UK Software Engineering Association (SEA) Workshop, scheduled for May 1998, until the autumn. The new network will bring together academics and industrialists, software engineers and information systems practitioners, and will provide an ideal forum for the distillation and dissemination of new ideas. For information about SEA, see *http://uksoft.co.umist.ac.uk/*

Calendar

April 1998

Second European Conference on Cognitive Modelling (ECCM-98), Nottingham, UK, 1 - 4 April 1998. *http://phoenix.herts.ac.uk/~rmy/eccm98/programme.html*

Third IEEE International Conference on Requirements Engineering (ICRE'98), Colorado Springs, Colorado, USA, April 5-10, 1998. *http://www.cs.technion.ac.il/~icre98/*The 1st UK Colloquium On Object Technology And System Re-Engineering, Oxford University, 6-8 April, 1998. *http://www.cms.dmu.ac.uk/STRL/cotsr/cotsr.html*

Ninth IEEE International Workshop on Software Specification and Design (IWSSD9), Ise-shima, Japan, April 16-18, 1998. *http://salab-www.cs.titech.ac.jp/iwssd9.html*

The Eleventh Workshop on Knowledge Acquisition, Modeling, and Management (KAW'98), Banff, Alberta, Canada, April 18-23, 1998. *http://ksi.cpsc.ucalgary.ca/KAW/KAW.html*

The 20th International Conference on Software Engineering (ICSE'98), Kyoto, Japan, April 19-25, 1998. *http://icse98.aist-nara.ac.jp/*

ICSE98 International Workshop on Component-Based Software Engineering, Kyoto, Japan, April 25-26, 1998. *http://www.sei.cmu.edu/technology/dynamic_systems/cbs/icsewkshp.html*

International Workshop on Human Dimensions in Successful Software Development, Kyoto, Japan, April 19-25, 1998. *http://btwebsh.macarthur.uws.edu.au/san/hudworkshop*

ICSE98 Workshop on Precise Semantics for Software Modeling Techniques (PSMT), Kyoto, Japan, April 20th. *http://www.forsoft.de/~rumpe/icse98-ws/*

The 1st IEEE International Symposium on Object-oriented Real-time distributed Computing (ISORC '98), in conjunction with The 4th IEEE Workshop on Object-oriented Real-time Dependable Systems (WORDS '98) and The 4th International Workshop on Object-Oriented Real-Time Systems (WOORTS '98), Kyoto, Japan, April 20 - 22, 1998. *http://dream.eng.uci.edu/isorc/*

May 1998

Fourth International Conference On Configurable Distributed Systems, Annapolis, Maryland, USA, May 4-6, 1998
http://doubletap.cs.umd.edu/CDS

Quality Week '98, 26-29 May 1998, San Francisco. *http://www.soft.com/QualWeek/QW98*

Third International Conference on the Design of Cooperative Systems (COOP'98), Cannes, France, 26-29th May, 1998
http://zenon.inria.fr/acacia/Coop/Coop98/

June 1998

Unified Modeling Language: Beyond the Notation (UML'98), Mulhouse, France, June 3-4, 1998
http://www.essaim.univ-mulhouse.fr/uml/evenements

"Modelling and Design". 5th International Eurographics Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS'98). Abingdon, UK, June 5-8, 1998.
http://www.dcs.qmw.ac.uk/research/hci/dsvis98

International Conference On Formal Ontology In Information Systems (FOIS'98), In conjunction with the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR'98), Trento, Italy, June 6-8, 1998. *http://mnemosyne.itc.it:1024/fois98/*

Fourth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'98) Pisa, Italy, June 8-9 1998 (Preceding the CAiSE*98 conference)
http://www.ifi.uib.no/konf/refsq98/cfp98.html

Third International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD'98), Pisa, Italy, 8-9 June 1998
http://www.ait.unl.edu/doc2/faculty/siau/home.htm

Conference on Advanced Information Systems Engineering (CAiSE*98), Pisa, Italy, 8-12 June 1998
http://www.pianosa.cnuce.cnr.it/caise98

IEE Colloquium on "Managing requirements change: a business process re-engineering perspective", IEE, Savoy Place, London, 11th June 1998.

5th International Conference on Software Process (ICSP5), Chicago, Illinois, USA, 15-17 June 1998
http://www.bell-labs.com/user/dep/prof/ispa/icsp5/

IEEE Seventh International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE '98), Stanford University, CA, June 17-19, 1998.
http://www.cerc.wvu.edu/WETICE/

Tenth International Conference on Software Engineering and Knowledge Engineering (SEKE '98), June 18-20, 1998 at the Hotel Sofitel, San Francisco Bay, USA.
http://www.ksi.edu/seke/seke98.html

Software Process Simulation Modeling Workshop '98, Silver Falls, Oregon, June 22-23, 1998
http://www.cs.pdx.edu/conferences/prosim98/

The Third International workshop on the Language Action Perspective on Communication Modelling (LAP98), Stockholm June 25-26, 1998
http://www.ida.liu.se/labs/vits/lap98/

July 1998

4th International Conference on Information Systems Analysis And Synthesis (ISAS '98), Orlando, Florida July 12-16, 1998
http://www.iiis.org

Second ECOOP Workshop on Precise Behavioral Semantics (with an Emphasis on OO Business Specifications), Brussels, Belgium, Monday, July 20th, 1998
http://www.forsoft.de/~rumpe/ecoop98-ws/

First ECOOP Workshop on Tools and Environments for Business Rules, Brussels, Belgium, Tuesday, July 21st, 1998
http://ecoop98.vub.ac.be/workshops/bizrules/biztools.html

August 1998

Sixth International Conference On Conceptual Structures (ICCS'98), Montpellier, France, August 10-14, 1998
http://www.lirmm.fr/ICCS98

International Workshop on Large-Scale Software Composition (in conjunction with the 9th International Conference on Database and Expert Systems Applications, DEXA'98), Vienna, Austria, August 24-28, 1998
http://www.iro.umontreal.ca/~keller/Workshops/DEXA98

Third IFCIS Conference on Cooperative Information Systems (CoopIS'98), New York City, August 20-22 1998
http://www.kean.edu/~mhalper/coopis98.html

September 1998

HCI'98, Sheffield Hallam University, 1-4 September 1998
http://www.shu.ac.uk/hci98

The 5th International Conference on Object-Oriented Information Systems (OOIS'98), Paris, France 9-11 September 1998.
http://panoramix.univ-paris1.fr/CRINFO/OOIS98

IFIP 13.2 Working Conference on Designing Effective and Usable Multimedia Systems, Stuttgart, Germany, September 9-11, 1998.
http://www.swt.iao.fhg.de/deums98

3rd Annual International Conference on Software Process Improvement - Research, Education and Training (INSPIRE '98), University of Sunderland, UK, 10th-11th Sept. 1998.
http://osiris.sunderland.ac.uk/INSPIRE98/

Third Northern Formal Methods Workshop, Ilkley, UK, September 14-15 1998
http://www.comp.brad.ac.uk/research/nfmw/

IFIP Working Conference on Engineering for Human-Computer Interaction (EHCI'98), Heraklion, Crete, Greece, September 14 - 18, 1998.
http://iihm.imag.fr/EHCI98

5th International School and Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'98) Lyngby, Denmark, Sept 14-15 (School) and Sept 16-18 (Symposium) 1998.
http://www.it.dtu.dk/~ftrtft98

Sixth European Workshop on Software Process Technology (EWSPT-6), near London, UK, September 16-18, 1998.
http://www-dse.doc.ic.ac.uk/~ban/misc/ewspt98.html

October 1998

The Third International Conference on Practical Software Quality Techniques (PSQT'98), St. Paul, Minneapolis October 5-7, 1998.
http://tcqaa.org/psqt/index.html

International Workshop On Current Trends In Applied Formal Methods, Boppard, Germany, 7-9 October, 1998.
http://www.dfki.de/vse/fm-trends/

13th IEEE International Conference on Automated Software Engineering (ASE'98), October 13-16, 1998, Honolulu, Hawaii, USA
http://www.ics.uci.edu/~ase98

Workshop on Industrial-strength Formal specification Techniques (WIFT'98), Boca Raton, Florida USA, October 21-24, 1998.
Info: chengb@cps.msu.edu

November 1998

Sixth International Symposium on the Foundations of Software Engineering (FSE-6), Orlando, Florida, USA, November 1-5, 1998
http://www.ics.uci.edu/fse6

3rd IEEE* High-Assurance Systems Engineering Symposium, Washington, DC, November 13-14, 1998
http://kel.eecs.uic.edu/hase98/

17th International Conference on Conceptual Modeling (ER'98), Singapore, Nov 16-19, 1998
http://www.iscs.nus.edu.sg/~er98

METRICS 98: Fifth International Symposium on Software Metrics, Bethesda, Maryland, November 19-21, 1998.
http://aaron.cs.umd.edu/metrics98

Requirenautics Quarterly   Issue 14 (April 1998)

The Newsletter of the BCS Requirements Engineering SG   Page 9.

December 1998'

Asia Pacific Software Engineering Conference 1998, Taipei December 1-4
http://www.selab.org.tw/apsec98

Software Process Improvement 98 (SPI98), Monte Carlo, 1-4 December 1998.
http://www.unipaderborn.de/cs/SPI98

The 19th IEEE Real-Time Systems Symposium, Madrid, Spain, December 2-4, 1998
http://www.cs.umd.edu/~rich/rtss98/

Second IEEE International Conference on Formal Engineering Methods (ICFEM'98), Brisbane, Australia, December 9-11 1998.
http://svrc.it.uq.edu.au/icfem98/

5th Maghrebian Conference on Software Engineering and Artificial Intelligence, Tunis, Tunisia, December 8-10, 1998
http://www.irsit.rnrt/

January 1999

Data Semantics 8 (DS-8): Semantic Issues in Multimedia Systems, Rotorua, New Zealand, 5-8 January 1999.
http://zulu.cs.rmit.edu.au/~ds8

February 1999

Third IFIP International Conference on Formal Methods for Open Object-based Distributed Systems (FMOODS'99), Florence, Italy, February 15th - 18th, 1999.
http://www.dsi.unifi.it/fmoods/

The First Working IFIP Conference on Software Architecture (WICSA1), San Antonio, Texas, USA, February 22-24, 1999.
http://www.bell-labs.com/usr/dep/prof/wicsa1/

International Conference on Process Modelling, Cottbus, Germany, 22-24 Feb 1999.
http://www.processmodelling.tu-cottbus.de

September 1999

FM'99: Formal Methods 1999, The World Congress on Formal Methods in the Development of Computing Systems, Toulouse, France, Late September 1999.
http://www.it.dtu.dk/~db/fm99/

$\mathcal{RE}$-Readings

Reviews of recent Requirements Engineering events.

Industrial Experience in Requirements Engineering

University of York, 4th February 1998

Report by **Pete Sawyer**

This event, organised and introduced by Andy Vickers (formerly of York, now at Praxis), explored the issues facing RE from an industrial perspective. A full lecture room suggested that there is a healthy interest in the practical problems faced by industry within the RE research community.

There were three speakers in addition to Andy: Neil Hudson, an independent systems and software engineering consultant; Jim Armstrong from British Aerospace Dependable Computing Systems Centre; and Jane Smith of the Rolls-Royce Systems and Software University Technology Centre. Each had a different perspective on the problem and each had some useful lessons to pass on.

In his introduction, Andy set the scope of the event by outlining what he viewed as the central problem: why is there a mismatch between academic theory and industrial practice? Andy observed that the problem with RE was fundamentally one of complexity. In Andy's view, the two sides of the community understand this as different things: researchers are interested in product complexity while industry's most pressing problems arise from organisational complexity. While researchers agonise over modelling properties of abstract systems, project managers have to cope with multi-disciplinary development teams, awkward customers, volumes of information and legacy systems. Even though industrial practice often falls short of researchers' idealised standards, industry by-and-large copes. However, it is critically dependent on key people with experience and domain knowledge - people who are in short supply.

Neil Hudson followed with an overview of his work on investigating the causes of accidents in "human/technical" systems. It is accepted that several well-known accidents - such as the Therac-25 - can be seen, at least in part, as failures of the requirements process. In his talk, Neil focused on what can be learned from accident analysis in order to understand how failures which, in safety-critical systems may lead to accidents, could be headed off at the requirements phase.

Neil's case was that this has wider significance than safety-critical systems since every system has requirements which are critical. In safety-critical systems these happen to be concerned with safety but, more generally, a critical requirements is simply one for which failure to meet it will prevent effective deployment of the system.

In his analysis, Neil has identified two classes of problem which appear to be dominant in the set of accidents which he analysed: knowledge and communication (e.g. the quality and availability of the information needed by operators), and

recording of operational data to help learn from operational experience (this seems to be particularly undervalued by system developers). Neil concluded by summarising some lessons for RE: that critical requirements must not only be identified, but identified as critical and treated accordingly through (e.g.) the allocation of resources, and V&V planning; that it is a mistake for RE to treat systems as a black box since this can lead to (e.g.) crucial environmental constraints being overlooked; and that the system's behaviour in the presence of erroneous environmental conditions should be explicitly specified.

Jim Armstrong spoke next from the standpoint of a formal methodologist working in industry. The key theme of Jim's talk was technology transfer of formal methods to real projects. Jim is piloting the use of formal methods in BAe based on the Omega approach - pragmatically generating formal representations of graphical specifications. The motivation for this approach comes from the need to address 2 of Anthony Hall's 7 Myths of Formal Methods: #4 Formal methods require highly trained mathematicians; and #6 Formal methods are unacceptable to users.

Most engineers use domain-specific (e.g. operational) models but find it hard to work with more abstract models. Domain-specific models are good for informally reasoning about systems. Abstract models, however, support formal reasoning and hence can support the proof of system properties (e.g. liveness) and specification properties (e.g. completeness). Jim's approach is to generate formal models from specifications written using engineers' favoured graphical notations. For example, the Statechart notation is popular for developing behavioural models. Although a Statechart is a "graphical formalism", it is not tractable to formal reasoning. By selectively generating a corresponding model in an axiomatic formal notation which supports proof, value can be added to a Statechart specification in a relatively painless way.

Using Jim's approach, the engineer can continue to develop models using the notation which best supports the informal reasoning necessary at the early stage of the specification process. Then , however, selected properties can be formally verified using a theorem prover without prematurely forcing the engineer into the straightjacket of an abstract formal notation.

This is intended to convince engineers of the value of formalism and to encourage them to explicitly identify system properties which should to be proven. The key point is to provide appropriate formalism. The work doesn't aim to completely insulate practising engineers from formalism but to control their exposure to it. Beyond sensitising engineers to the utility of formal methods, Jim's work has also had some useful side- effects. In particular, the unfeasibility of proving large Statechart specifications and the need for simplicity has led to the development of tailored graphical language subsets which are proving better suited to both formal and informal reasoning.

The last to speak was Jane Smith who described an RE process improvement project at RR. This was motivated by shortcomings in the existing RE process which most people will recognise: it was dominated by large volumes of documentation; traceability information was lost; visibility of the information was poor; and requirements were often ambiguous.

The project began with the definition of a new, more technically- focused and less document-bound process. This includes a simple and improved traceability model. A new information model has been designed to help structure requirements information to make it more manageable and manipulable. Tool support for the new process is being prototyped to provide management of the process through a structured database of requirements information. This stores requirements in a standard form which is intended to support the use of a structured English subset in order to counter existing problems with (e.g.) ambiguity. The tool also associates attributes with requirements (e.g. risk, stability), maintains traceability links, and supports document generation.

In the longer term, the project aims to help improve both informal and formal specification, support safety case development, help with technology transfer, and support a continuous process improvement programme. Jane identified lessons from the project for industry wanting to exploit research and for academia wanting to influence practice. Industry should examine their processes before introducing new techniques and be cautious before committing to a particular tool in case it fails to support their needs. In particular, automation will necessitate adaptation of the process and this needs to be recognised. Academia needs to be careful to acquire an understanding of industry's real problems and only attempt to introduce changes incrementally - ambitious changes are too risky for industry.

In the discussion which followed, this last point was echoed but a dichotomy pointed out: to be successful, change has to be cautious and incremental, yet managers look for dramatic improvements - an example of the organisational politics which have to be handled when trying to do RE technology transfer. Tool selection was also identified as problematic for industry because of the need for openness. Tools need to be seen as long-term investments because requirements information often needs to persist for decades. Tools and techniques for RE also need to be flexible enough to support widely varying processes and the different roles played by customers, procurement agencies and developers. The discussion ended on a hopeful note. Summing up, Andy concluded that maybe the gap between industry and academia isn't quite as wide as we feared. The key technology transfer problem is getting easier as researchers get a better understanding of what industry's problems are and what the industry will accept in terms of change.

Introducing the Unified Modeling Language

A tutorial by Dr Stephen Morris of City University, held on Wednesday 25th March 1998 at Imperial College.

Report by Ian Alexander

UML 1.1 consists of 365 pages of documentation on the Rational website (http://www.rational.com). UML is the product of the 'Three Amigos' Booch, Rumbaugh, and Jacobson, so called from the moment when Rumbaugh accompanied himself on his guitar at a conference. The three are all now employed as gurus by Rational to help to push the concept of a single language which can serve the very diverse needs of requirements, analysis, design, and implementation (but curiously, not verification, which one might have expected of something organised around use cases).

Their aims are certainly ambitious, as Dr Morris explained in his quietly understated way:
  to model systems and software using an Object-Oriented technique
  to link to both conceptual and executable artefacts
  to scale up to mission-critical systems
  to provide a language usable by both humans and machines

or in their own words "specifying, constructing, visualizing and documenting ... systems" independent of any particular programming language, offering a formal basis for modeling, and to encourage the basis of the tools market (laughter at this last item). Rational Corp. is assuredly not a public standards organisation.

UML is certainly not a software development process, nor a method, nor connected to anyone's favourite programming language: but tool vendors will very likely make all three of this links in the next year or two, thinks Morris.

The seminar proceeds with a combination of matter-of-fact description of UML as it now is, some verbatim restatement of the party line, and a certain amount of enjoyably drily humorous observation. For instance "almost everything in UML is a regular polygon" -- as opposed to the complex range of graphical notations replaced (broadly) by UML, including Jacobson's Objectory.

The three fundamental relationships which are handled by UML are generalisation (taxonomic), dependency (functional) and association (quite general). There was some feeling in the room that the offered semantics were weak and ambiguous. Morris said that the semantics in the Notation Guide were "a bit unfortunate" and that the coverage of individual items varied "from perfunctory to extremely lengthy". Some parts of the documentation could be improved: the indexing, for example, required users "to take 6 off the number in the index and it'll work quite well".

Association is widely used in UML models to link classes in a manner reminiscent of entity-relationship diagrams: the labels on the relationships are arbitrary and chosen by the user, so the semantics are weak. Quantifiers such as 0..1 are optional, as are markers of directionality, so it is permissible to construct models which are little more than sketches: this can be useful early in the systems life-cycle.

Early versions of UML were heavily criticised for their lack of semantic precision. The addition of the Object Constraint Language and the UML Semantics documentation has gone some way to strengthen UML's position. But it remains impossible to move from semantics to (uniquely determined) syntax. This situation is partially explained by the presence of the very diverse methods of the three amigos: in October 1995 a Unified Method (0.8) was released to join Booch and Rumbaugh's approaches, but by September 1996, with Jacobson also on board, the 0.91 addendum coyly drops the word Method in favour of Modeling Language, implying that while the artefacts (diagram constructs) are unified, the underlying methods cannot be.

The semantic model in fact permits a wide range of variation, including such catch-alls as the ability to modify the meta-model by creating Stereotypes to classify and mark elements, and the ability to describe Subsystems with new functional behaviours. In fact, many key features of UML such as System and Use Case Model are constructed using the Stereotype extension mechanism. It is no surprise, then, to discover that Jacobson's Objectory is catered for: users can draw a large range of types of diagram and symbol and can use these to mean more or less what they want. Morris remarks that Stereotypes "allow rampant destandardization".

In fact, the View Element is stated to be a textual or graphical presentation of model elements ... "proper to a graphic editor tool": in other words, UML does not try to restrict how tools may choose to present UML diagrams. The replacement of 3 methods' presentations by one was warmly commended by Morris, and welcomed by some academics present, but the implied commonality of diagram types may not last long.

Some of UML's diagram types are surprising. The sequence diagram, for instance, represents time on the vertical axis and seeks to show each object as a bar; methods that create objects (pretty much like procedure calls) are shown as arrows running horizontally, giving scope for showing both branching and recursion graphically. Conditions can be written in directly. The effect is strikingly procedural, and there was quite extensive discussion of the representation.

Other aspects of UML have clearly given its authors trouble: some 20% of the semantics section is concerned with defining state machines for state transition diagrams!

UML has certainly achieved a measure of standardisation of notation to support object-oriented software development. It has rather complicated semantics for such a notation, and a proliferation of diagram types. Its value is questioned by Morris - though the popularity of the seminar shows that Rational have succeeded at least in attracting a lot of interest. Morris concluded that "the next 12 months will be crucial - the impetus will die unless [other] tool vendors bring out really useful tools." It will be interesting to see.

CORE-Blimey!

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

They Do It Their Way … Unfortunately

In previous contributions I have indicated features necessary for support for specification. There are approaches supporting subsets of these features, but none supports the whole set. This time I use three existing approaches to illustrate limits on individual approaches. I also note reservations about the way combined approaches appear to be developed. The examples I use are Relational Databases, Logic (in particular, Prolog) and Object Orientation (OO).

The **Relational Database** approach assumes that set theory, normalisation and referential integrity are sufficient to describe everything of practical interest. It assumes that each record describing a set member can/should be described in terms of the same one-dimensional properties (fields). For efficiency purposes it uses keys/indexes for quick access. Combining sets, normalisation and key/indexes leads to both the strengths and weaknesses of the Relational Database approach. I am interested here only in the weaknesses for specification support.

It is necessary to identify uniquely each set member, but the relational approach encourages using mnemonic or numeric codes to form keys. That introduces redundancy, obscurity and opportunities for error. This can only be partially reduced by front-ends which manage generation and use of keys. The way keys/indexes are implemented and used conflicts with the very principles to which the relational model is supposed to adhere.

When searching/reporting on database contents it is often impossible to avoid learning a lot about the underlying schema and key/index formation rules. Natural language front ends I have seen do **not** remove this problem and can lead to invalid conclusions since they assume common interpretations of words and their association. As the specification audience widens this becomes a progressively less valid assumption. If (as I believe) flexibility of expression is required then each time a new form of expression is added the words/associations must be extended without introducing contradictions.

As a small example arising from the problems of relational databases, it is not, in general possible simply to ask the question "Tell me all that is known about X". The reason is that there is no mechanism for tracking all references to X … (oh no there isn't!). Even if each item is stored in a table with a unique key, all relationships between that table and other tables must be known by the query writer and each related table must be explicitly accessed by the query.

If a special query is produced to access all current related tables, then when new tables are added the query will have to be re-written to access them. Moreover the query must be re-cast for each distinct type of item the user might want to ask about and the user must know/remember which query to use for which item type. This means that users of the database must have a detailed knowledge of the information it contains and the way in which it is contained. In other words, only experts in that particular database can be allowed to specify efficiently – cutting out 90% of the people who actually know the details from which the specification must be derived.

In addition, relational database schemas are too complex and unwieldy to manage without hitting flexibility and immediacy. Changing them later frequently requires knowledge only held by a few people (if any, after staff turnover). Normalisation emphasises the problems of obscurity and potential errors with keys and introduces a proliferation of linking tables which users can't understand. Referential integrity is still partly in the hands of the schema definer – meaning that dangling references are still all too probable.

The **Logic (Prolog)** approach describes an environment via rules expressed in a form of logic. A support tool for such an environment expression merely exercises the rules, using external stimuli to "fire" rules which generate responses. Validation of the rules is achieved by observing the stimulus-response behaviour and (optionally) using an explanation facility to show how a particular stimulus-response mapping arose from the rules.

For a logic oriented community Prolog provides better support for immediacy than databases – but is obscure to the point of despair for any other community. The idea is that exercising the rules is all that is needed to convince users/customers that they cover all the cases of interest. However, who determines what are all the stimulus/response cases necessary to demonstrate coverage? It is not sufficient just to sit users at the tool and let them play. They must play according to a well thought-out plan. Such a plan can only be made if they have some form of specification from which to work and the only specification they have is in Prolog - which they cannot necessarily understand.

Calling for them to be trained in understanding Prolog is not, in general practicable. I know of one user group who walked out in the middle of being introduced to the entity-life history notation used in JSD – and that is little more complex than understanding BNF, so I am confident they would have formed a lynch mob if faced with training in Prolog.

Prolog also fails at answering the question "tell me everything known about X". For example, if you have a construct like p (a, b, c (d, X, e), f) the embedded presence of X cannot reliably be found by executing the rules. The Prolog explain facility can show that it has appeared after

applying the right stimulus, but in the absence of finding the right stimulus the presence of that reference to X is detectable only by either a text editor search or a Prolog program which inspects other Prolog programs for such references and extracts and lists the rules it finds.

The editor search/ special Prolog program might be viable for a single file of rules, but frequently many files are needed and not solely related to the environment's rules of behaviour, There are other things such as project management (time scales, budget, resource allocation, etc.) which might relate to the "X" about which information is sought. These additional details can be defined in Prolog, but would not in general sensibly belong in the same Prolog specification file as the technical specification, nor in the same location or host computer.

Hence the program-reader which finds references to X must now inspect all relevant specification files and all related files in multiple locations. The major strength of Prolog, unification, is also a weakness its information base grows and becomes distributed. Grabbing as much RAM as is currently available postpones the problem, but does not remove it and the problem of secure, complete access across distributed hosts is not one solved by Prolog. I do know how to overcome some of the memory and distribution problems, but it requires a modification of how unification is applied – sacrilege to the high priests I suspect.

The **OO approach** uses a classification scheme, extendable via inheritance and composition and maps the real world onto it. Much is made of abstraction, whose purpose is to hide/protect internal detail and polymorphism, which allows the same name to be used for the same concept when dealing with different classes, even if the realisation differs between classes. Many assume OO is enough – i.e. that any environment should be specified only in terms of the existing classification scheme – with allowance for inheritance and composition. There are two problems with this.

First, the well-established legal precedent of identifying the danger of what is called a *Leading Question*. The idea is that, by asking a question which suggests the answer or limits the scope of the answer to a small selected set of possibilities you can persuade the witness to give evidence which seriously distorts the truth. In law this is only allowed when there is serious reason to suppose that the witness will not otherwise arrive at the truth – e.g. has a prior bias for or against one side in the case.

The assumption that everything should be specified in terms of a pre-defined classification scheme is one which is quite likely to result in users (witnesses in the legal context) being led into describing an environment which fits the class model but conflicts with reality. The result may well be a reliable, well-specified, well-designed system which is not what the users meant.

Second, the principle of inheritance and composition from a basic classification scheme assumes the basic scheme

RE-Bites...

As a requirements engineer, one develops an acute awareness of the frailties of the English Language. Many requirements engineers become collectors of examples of garbled requirements from everyday life. The best examples have multiple ambiguities. Of course, if you are a serious requirements engineer, you will be able to figure out what the real requirement is straight away, and model it using a precise specification language:

On a canned fruit juice: **"Shake well before enjoying"**. Not only is it not clear whether the 'well' attaches to 'shake' (as in 'shake well') or to 'before' (as in 'well before'), but also what exactly is the instruction? If you don't plan on enjoying the drink, should you not shake it? Are they saying that shaking is a precondition to enjoying, or just that it must precede it?

In some restaurants: **"Shirts must be worn"**. Let's ignore the double meaning of the word 'worn'. You get a feel for the deeper ambiguity here by considering the juxtaposition of two signs on a set of escalators: "Dogs must be carried" and "Shoes must be worn". (*thanks to Michael Jackson for this one*). Worse still, the sign in the restaurant does have not have the simple meaning that if you're not wearing a shirt you won't be served, as clearly all sorts of other clothing types would suffice...

Sign on US freeways: **"Maintain Speed"**. It can't just mean 'don't slow down', as I might not be going fast enough in the first place! As a requirements engineer, I'm obliged to apply the testability criteria: how the hell would I know if I'm meeting this requirement?

captures all necessary concepts. This is both philosophical and practical nonsense. We are still seeking a grand unification theory in physics and have yet to prove that features such as thought and creativity really emerge from such a theory, so we cannot safely assume we can derive everything from an existing classification scheme. That is the path to a modern Ptolemaic universe, where we twiddle our existing models to "explain" reality, building a gradually more elaborate and risible house of cards.

Though the OO approach to specification is a highly valuable one, there are aspects to specification which require that an environment can be described in other forms. The OO approach becomes most valuable when those other forms of specification can be shown by a mapping to conform accurately with elements of the existing classification scheme or derived from them by inheritance and/or composition.

There are other limitations I believe apply to each of these approaches, but I will ignore them here. Together with the problems I have illustrated here I conclude that, though each approach as flexible, it is still too limiting when applied alone. A boy scout penknife is flexible, but I would not want to use it for running a timber yard.

That suggests the possibility of improvement by combined approaches. My problem there is that combinations seem invariably to be attempts by advocates of one approach try to bolt on a bodge to deal with its weaknesses by using other approaches as a subservient tools. This misses the fact that these are distinct theories and combination must make them interact, not make one subservient to the other.

For example, I believe that the data models inherent in Relational Databases and Prolog specifications are different enough that it is not sufficient to have Prolog call up records from Relational tables, using them simply as repositories for fixed length Prolog facts (atoms). That throws away the power of the relational model to do (for example) set and sorting operations. Equally, providing a Prolog interface only to operate on a subset of the information stored in a set of relational tables throws away the much greater expressive flexibility of Prolog.

I have corresponding problems with the little I have read about attempts to combine relational and OO database approaches. What appears to be happening is that one set of advocates are embedding the other technology as subservient to their own and then assuming that the embedded technology's querying capability need only be called up after discovery of an embedded component by their own enveloping query technology. I have not seen an attempt to combine OO and Prolog approaches, but given the quasi-religious nature of each community I would expect it to be based on the embedded servant idea rather than a peer to peer idea.

The point here is not that the approaches are without merit, nor that it is invalid to combine them (including combination with other unmentioned approaches). My point is that each has merits, but none is so dominant in merit that it can safely be considered the dominant partner. Specification support tool builders should be moving to peer-to-peer linking of approaches.

That may require that information is passed/shared between tool data sets, infringing normalisation which is so beloved in the relational database community. If proper support is provided for referential integrity, redundancy and thence normalisation does not matter. This relates to a concept I introduced 19 years ago, called constructive redundancy – but I will save discussion of that for another time.

Geoff Mullery
geoff_mullery@dial.pipex.com

$\mathcal{RE}$-Calls

Recent Calls for Papers

Second ECOOP Workshop on Precise Behavioral Semantics (with an Emphasis on OO Business Specifications), Brussels, Belgium, Monday, July 20th, 1998

http://www.forsoft.de/~rumpe/ecoop98-ws/

The Workshop will be held in conjunction with ECOOP'98 on Monday, July 20th at Brussels, Belgium.

Workshop themes

Business specifications are essential to describe and understand businesses (and, in particular, business rules) independently of any computing systems used for their possible automation. They have to express this understanding in a clear, precise, and explicit way, in order to act as a common ground between business domain experts and software developers. They also provide the basis for reuse of concepts and constructs ("patterns") common to all - from finace to telecommunications -, or a large number of, businesses, and in doing so save intellectual effort, time and money. Moreover, these patterns substantially ease the elicitation and validation of business specifications during walkthroughs with business customers, and support separation of concerns using viewpoints.

Precise specifications of business semantics in business terms provide a common ground for subject matter experts, analysts and developers. If business specifications do not exist, or if they are incomplete, vague or inconsistent, then the developers will (have to) invent business rules. This often leads to systems that do something quite different from what they were supposed to do.

Business specifications (the "what"s) are refined into business designs (the "how"s), from where creation of various information system (software) specifications and implementations based on a choice of technological architecture are possible. In this context, precision should be introduced very early in the lifecycle, and not just in coding, as it often happens. In doing so, "mathematics is not only useful for those who understand Latin, but also for many other Citizens, Merchants, Skippers, Chief mates, and all those who are interested" (Nicolaus Mulerius, one of the first three Professors of Groningen University, early 17th century).

Precise specification of semantics - as opposed to just signatures - is essential not only for business specifications, but also for business designs and system specifications. In particular, it is needed for appropriate handling of viewpoints which are essential when large and even moderately sized systems, both business and computer, are considered. Viewpoints exist both horizontally - within the same frame of reference, such as within a business specification - and vertically - within different frames of reference. In order to handle the complexity of a (new or existing) large system, it must be considered, on the one hand, as a composition of separate viewpoints, and on the other hand, as an integrated whole, probably at different abstraction levels.

Requirenautics Quarterly   Issue 14 (April 1998)

The Newsletter of the BCS Requirements Engineering SG   Page 15.

Many concepts and constructs used for all kinds of behavioral specifications - from business to systems - have common semantics and thus are good candidates for standardization and industry-wide usage. Various international standardization activities (such as the ISO Reference Model of Open Distributed Processing and OMG activities, specifically the more recent ones around the semantics of UML, business objects, and other OMG submissions, as well as the creation of OMG semantics working group) are at different stages of addressing these issues. OMG is now interested in semantics for communities of business specifications, as well as in semantic requirements for good (business and system) specifications.

It is therefore the aim of the workshop to bring together theoreticians and practitioners to report about their experience with making semantics precise (perhaps even formal) and explicit in OO business specifications, business designs, software and system specifications. This is the 8th workshop on these issues; we already had 7 successful workshops, one at ECOOP and six at OOPSLA conferences. Reuse of excellent traditional "20-year-old" programming and specification ideas (such as in [1,2]) would be warmly welcomed, as would be reuse of approaches which led to clarity, abstraction and precision of such century-old business specifications as [3]. Experience in the usage of various object-oriented modeling approaches for these purposes would be of special interest, as would be experience in explicit preservation of semantics (traceability) during the refinement of a business specification into business design, and then into a system specification.

[1] E.W.Dijkstra. On the teaching of programming, i.e. on the teaching of thinking. In: Language hierarchies and interfaces (Lecture Notes in Computer Science, Vol. 46), Springer Verlag, 1976, pp. 1-10.

[2] System description methodologies (Ed. by D.Teichroew and G.David). Proceedings of the IFIP TC2 Conference on System Description Methodologies. North-Holland, 1985.

[3] Charles F.Dunbar. Chapters on the theory and history of banking. Second edition, enlarged and edited by O.M.W.Sprague. G.P.Putnam's Sons, New York and London, 1901.

Topics

The scope of the workshop includes, but is not limited to:

- Appropriate levels and units of modularity
- Which elementary constructs are appropriate for business and system specifications? Simplicity, elegance and expressive power of such constructs and of specifications.
- Using patterns in business specifications
- Making Use-Cases useful
- Discovering concepts out of examples (Generalization techniques)
- Providing examples from specifications

- What to show to and hide from the users
- How to make diagram notations more precise
- Equivalence of different graphical notations: "truth is invariant under change of notation" (Joseph Goguen)
- Semantics above IDL
- Rigorous mappings between frames of reference (e.g. business and system specifications)
- Role of ontology and epistemology in explicit articulation of business specifications
- Formalization of popular modeling approaches, including UML
- On complexity of describing semantics

Organizers

Haim Kilov
Merrill Lynch
Operations, Services and Technology
World Financial Center
South Tower
New York, NY 10080-6105
email:haim_kilov@ml.com or haim_kilov@omg.org

Bernhard Rumpe
Institut für Informatik,
Technische Universität München
80333 Munich, Germany
email:rumpe@forsoft.de

Important Dates

Deadline for submission: May 1, 1998
Notification of acceptance: May 26, 1998
Final version: June 8, 1998
Day of workshop: July 20, 1998

Please note that workshop participants must register at least on that day at ECOOP conference. Deadline for early registration at the ECOOP conference is Friday, June 19th.

Proceedings will be printed as technical report of the Munich University of Technology and will be available at the conference. As in the year before, we will select the best papers to be published in the workshop reader (probably Springer, LNCS).

Submissions

Workshop proposals should be about 5-10 pages and highlight the main contributions of the author(s). Interesting papers will be selected by the organizers and their authors will have the possibility to present them in about 20-30 minutes. Furthermore, each author is encouraged to present open questions and one or two main statements that shall be discussed. Workshop participants are invited to submit their contribution as Postscript or Word as email to rumpe@forsoft.de.

Requirenautics Quarterly   Issue 14 (April 1998)

The Newsletter of the BCS Requirements Engineering SG   Page 16.

The 21st International Conference on Software Engineering (ICSE 99), Los Angeles, USA, May 16–22, 1999

Software pervades the growing web of computers, communications, personal and commercial services, financial and public services, and entertainment services, evolving into the global electronic community of the 21st century. Providing timely, cost-effective, high-quality software for the diverse stakeholders of this community will require the best possible technology and practices the software engineering community can provide.

ICSE 99 provides a number of linked forums for bringing together software engineering practitioners and researchers to present and discuss the most promising approaches for meeting these challenges. The forums include the main conference May 19-21, 1999, and associated special-topic workshops, symposia, and tutorials May 16-18 and May 22. The ICSE 99 main conference will include refereed technical papers, invited industry presentations, panels, exhibits, and leading keynote speakers. Featured keynote speakers will include Dr. Alan Kay, Vice President and Disney Fellow, Walt Disney Imagineering; Dr. Butler Lamp-son, Architect, Microsoft Corp.; and Dr. William Wulf, the first software engineer to serve as President of the U.S. National Academy of Engineering.

The conference will be organized around four main sub-themes:

  Quality and Service to People. Scaling up software engineering technology for global use; ensuring safety, security, helpfulness, and quality of service for diverse sets of users; providing safe and powerful user-programmability.

  Product Line Architecture and Componentry. Domain engineering, software architectures, reuse, reengineering, and evolution support.

  Integrating Process and Product Engineering. Con-current process/product engineering, management, and control; new processes supporting reuse, rapid and high-quality development and evolution; pro-active process/product support tools and environments.

  Advanced Applications. Case studies and lessons learned in meeting 21st century challenges: rapid applications development, COTS integration, high-tech/high-touch product tailoring, systems of systems, networks of net-works, hypermedia, autonomous agents.

Besides these topics, contributions are also encouraged in the full range of software engineering topics.

Conference Structure

The International Conference on Software Engineering (ICSE) is the leading international forum for the exchange of ideas on software engineering. The conference features a number of coordinated forums, including presentations on new research results, reports on industrial case studies and experiences, panels on topics of current interest, system demonstrations, industrial exhibits, tutorials, and a doctoral symposium. These activities aim to provide a comprehensive, informative program for software engineering professionals, researchers, managers, and students.

ICSE 99 invites you to contribute to its broad-based program by submitting in one or more of the categories outlined below. Topics of interest span the full range of software engineering issues, including those listed at the right.

Technical Papers - Report on new research results and on practical experiences.

Panels - Discuss and debate issues of current interest to the community.

Case Studies - Provide in-depth description of industrial systems and lessons learned in building them.

Formal Research Demonstrations - Describe and exhibit working systems or prototypes.

Poster Sessions and Informal Demonstrations - Present late-breaking results and ongoing work.

Doctoral Symposium - A forum for Ph.D. students to present their research.

Tutorials - Short courses on software engineering practices, techniques, and theories.

Workshops - Focused discussion of selected topics in a small-group setting.

Exhibitions - Demonstrations of commercial tools and technologies.

Submission Deadlines

Workshops	June 5, 1998
Technical paper abstracts	Aug 14, 1998
Technical papers	Aug 28, 1998
Panel proposals	Aug 28, 1998
Case studies	Aug 28, 1998
Tutorials	Aug 28, 1998
Doctoral symposium	Jan 4, 1999
Research demos & posters	Jan 4, 1999

General Chair

Barry Boehm,
USC, USA, boehm@sunset.usc.edu

Program Chairs

David Garlan,
CMU, USA, icse99@cs.cmu.edu

Jeff Kramer,
Imperial College, UK, jk@doc.ic.ac.uk

For more information

Please see the ICSE 99 web site:
http://sunset.usc.edu/icse99

$\mathcal{RE}$-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: http://research.ivv.nasa.gov/~steve/resg/

Web Pages

The BCS RESG home page can be found at:
http://porta.cs.york.ac.uk/bcs/resg/

Back issues of Requirenautics Quarterly:
http://research.ivv.nasa.gov/~steve/resg/

Compendium of Software Engineering Tools:
http://www.methods-tools.com

Books

Biren Prasad "Concurrent Engineering Fundamentals. Volume II: Integrated Product Development". Prentice Hall, 1997. ISBN # 0-13-396496-0

Mailing lists

Software Requirements Engineering Mailing List

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

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

Forum for Advancing Software engineering Education

FASE (Forum for Advancing Software engineering Education) was started in 1991 by members of the software engineering education community in order to have a electronic forum for the dissemination and discussion of events related to software engineering education. The original acroynm for FASE was Forum for Academic Software Engineering, but was subquently changed so that it was more inclusive to industrial and government training issues (which led to a co-editor for that area).

FASE has published 97 issues to date, and is presently on a monthly schedule, being published on the 15th of each month. FASE currently has about 670 subscribers in over 40 countries.

Previous issues of FASE can be found at http://www.cs.ttu.edu/fase.

To join the FASE mailing list, write to <listserv@cpm211-1.cs.ttu.edu> and, in the text of your message (not the subject line), write: subscribe fase.

New Software Risk Mailing List

A new mailing list for discussion of software risk topics has been created called software-risk. the list is intended for academics, software engineers, project managers, or anyone else interested in software risk. The list offers a forum for discussion on software risk management. The list may be used to ask a question relating to risk, answer someone else's question about risk state an opinion on a risk related topic, disagree with an opinion (respectfully of course), promote relevant events, conferences, courses etc book reviews, book recommendations bouncing ideas off others etc etc so long as it is broadly relevant. At a later stage, files containing information relating to software risk will be added to the mailbase web site at:
 http://www.mailbase.ac.uk/lists/software-risk/

The list owner can be contacted at
software-risk-request@mailbase.ac.uk

To join software-risk send the following command, (typing your own personal names instead of firstname(s) and lastname)
     join software-risk firstname(s) lastname
as the only text in the body of a message addressed to:
 mailbase@mailbase.ac.uk

After joining you are a member of the mailing list and you may send messages to the list by sending an email to software-risk@mailbase.ac.uk. Everyone on the list membership will receive the message. Non-members cannot send messages to the list or review the list members. You may of course subsequently leave software-risk by sending this command:
  leave software-risk
as the only text in the body of a message addressed to:
 mailbase@mailbase.ac.uk

For any queries relating to the list, contact the owner at: software-risk-request@mailbase.ac.uk

Tools

Integral System Simulation (ISS) is a modelling, simulation and analysis package with graphics and animation facilities, developed at the Philips Research Laboratories in co-operation with Philips Medical Systems. It accompanies the existing system development life cycle from specification to design. The main characteristic of ISS is the multi-disciplinary and integral system approach. In ISS a system, in general, is viewed as en integration of sub systems with different kinds of physics and technologies, with software as well as hardware and with a user interface. The ISS/Structured Analysis tool is part of this concept, aimed at helping system designers in specifying their system. The ISS/Architecture Modelling tool is aimed at helping system designers in designing their system.

The ISS tool is based upon the Structured Analysis/Structured Design (SA/SD) method of Hatley and Pirbhai. It is described in their book "Strategies for Real Time System Specification". Since the chapter on architecture was quite brief, the method had to be enhanced and/or adapted. The enhancement includes the concept of architecture modules for specific technologies. Included are modules for processing technologies and interconnect technologies.

Availability: The tool is not available from Philips. Currently, it is being introduced for commercial use by CACI under the name Verispec. Information on this product can be found at:

http://www.caciasl.com/designval.html

RE-Actors

The committee of RESG

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

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

Secretary: Wolfgang Emmerich, City University, Department of Computer Science, Northampton Square, London EC1V OHB. W.Emmerich@cs.ucl.ac.uk, Tel: +44 171 504 4413, Fax: +44 171 387 1397

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

Industrial Liaison Officer: Dr. Orlena Gotel, Systems and Software Engineering Centre, Defence and Evaluation Research Agency, St. Andrews Road, Malvern, Worcestershire, WR14 3PS. olly@hydra.dra.hmg.gb, Tel: 01684 894674

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

Events Officer: Carol Britton, Department of Computer Science, University of Hertfordshire, College Lane, Hatfield, UK. AL10 9AB, c.britton@herts.ac.uk, Tel: 01707 284354, Fax: 01707 284303

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

Newsletter Associate Editor (Features and Book Reviews): Dr George Spanoudakis, City University, Department of Computer Science, Northampton Square, London EC1V OHB. gespan@cs.city.ac.uk, Tel: 0171 477 8000 ext. 3701, Fax: 0171 477 8587.

Newsletter Associate Editor (Features and Event Reviews): Ian Alexander, 17A Rothschild Road, Chiswick, London W4 5HS. iany@easynet.co.uk. Tel: 0181-995 3057

RE-Creations

How to contribute to RQ

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

http://research.ivv.nasa.gov/~steve/resg/rq5/ReCreations5.html

	Copy deadline
Issue 15 (July)	26th June 1998
Issue 16 (October)	25th September 1998
Issue 17 (January)	18th December 1998