



RE '04 Tutorial Program

<http://www.re04.org>

Tutorial Timetable

Date	Time	English	English	Japanese
Sep. 6 (Mon)	AM	T-1: Requirements-Driven Product Line Development	T-2 Requirements: the Bridge between Business and Development	
	PM	T-3 Requirements-Based Product Line Engineering	T-4 Requirements and Creativity	J-1 Introduction to Scenario Analysis
Sep. 7 (Tue)	AM	T-5 Developing Practical Scenarios		J-2 Requirements Traceability Methodology and Its Application to UML
	PM	T-6 Software Traceability	T-7 Financially Informed Requirements Prioritization	

Introduction to Tutorials

T-1 Requirements-Driven Product Line Development: Scoping for Reusability, Modeling for Implementability

Time: Sep. 6(Mon) Morning

Lecturer: Klaus Schmid (Fraunhofer IESE, Germany)

Klaus Schmid is department head for RUE (requirements and usability engineering) at Fraunhofer IESE. He has been involved in several projects that have successfully transferred the product line development paradigm and other software engineering technologies into industrial environments. At Fraunhofer IESE he contributed to the initial definition and the continuous evolution of the PuLSE method for Product Line Software Engineering. His main research interests are software evolution, requirements engineering, and institutionalizing product line development in industry. He authored numerous papers on this and related topics. He organized several workshops on topics like software product lines and their evolutionary introduction and is currently workshop chair of the Software Product Line Conference (SPLC'3) and the International Conference on Software Reuse (ICSR'8). He received an M.S. and a Ph.D. in computer science from the University of Kaiserslautern.



Abstract: Product line development has proven a successful approach to achieve unprecedented levels of reuse and hence time-to-market and effort reductions. A key challenge within product line engineering is to identify (and describe) the right functionality for reusable implementation. Three activities need to be handled successfully in order to address this challenge:

- A clear vision (i.e., the key requirements) of the future products must be identified,
- The key requirements that drive the product line - and in particular product differentiation - must be identified,

- A detailed analysis requirements analysis of the product line must be performed, where in particular common and variable requirements are systematically identified and described. Thus elicitation, management, and modeling of requirements are strongly interwoven in the software product line context.

In this tutorial we will discuss these methodological challenges and will discuss concepts and techniques to overcome them. In particular, we will describe how to construct a product portfolio using simple feature analysis techniques. This will provide the basis for a systematic ROI analysis of reuse opportunities (reuse scoping). For capturing variabilities in product line description various techniques exists (e.g., feature trees). We will discuss them and provide an approach that can be applied in order to extend existing RE notations in a systematic manner for product line usage in order to support a seamless transition to product line engineering.

T-2 -2 Requirements: the Bridge between Business and Development

Time: Sep. 6(Mon) Morning

Lecturer: Suzanne Robertson (The Atlantic Systems Guild UK)

Suzanne Robertson is co-author of Mastering the Requirements Process (Addison-Wesley 1999) a book that provides guidance on finding requirements and writing them so that all the stakeholders can understand them. She is also co-author of the Volere <http://www.volere.co.uk> approach to requirements engineering.



She has more than 30 years experience in systems specification and building. Her courses on requirements, systems analysis, design and problem solving are well known for their innovative workshops and practical applicability. Current work includes research and consulting on finding and involving the right stakeholders, the building of requirements knowledge models and running audits for assessing requirements specifications.

Her forthcoming book, Requirements-Led Project Management: Discovering David's Slingshot, addresses the question of how project managers can take advantage of requirements to plan and steer the project. She is a principal and founder of The Atlantic Systems Guild <http://www.systemsguild.com> and is editor of the Requirements column in IEEE Software magazine.

Abstract: There are a myriad of ways of talking about requirements. If you work in marketing you are concerned with features, a business person is more interested in business functions, an analyst talks about use cases, a systems architect is concerned with architecture components, a programmer thinks about code. This diversity of views is necessary because each of a project's stakeholders needs to view the requirements according to how they affect his own concerns. However, it is impossible to communicate, manage and trace requirements without an agreed bridge between these different views. The requirements knowledge model provides this bridge.

This tutorial draws on the experience of building requirements knowledge models with many different organisations. The starting point is a generic knowledge model along with guidelines for how to tailor it to the needs of each individual project. A consistent way of communicating requirements makes it quicker to accurately transfer knowledge across organisational and geographical boundaries and minimises misunderstandings thus leading to faster planning and development. The knowledge model provides a common linguistic framework for involving the necessary mixture of strategic, marketing, operational and technical skills at the appropriate time.

This practical tutorial is about the different meanings that requirements have for different people in an

organisation and how to build a bridge that connects those meanings.

The 5 sections of the tutorial are:

1. Surveying the Terrain: Identifying and involving the appropriate stakeholders.
2. A Generic Starting Point: A model that packages the atomic requirements for different viewpoints.
3. Tailoring to Suit Your Project: How to tailor the model for a specific project.
4. Refining to Capture Experience: Techniques for iteration and involvement. Monitoring the stability of the model.
5. Using the Model: Input to designing the project plan. Responsibility for knowledge, level of detail, quality criteria, packaging of knowledge, agreed traceability.

T-3 Requirements-Based Product Line Engineering

Time: Sep. 6(Mon) Afternoon

Lecturers: Mike Mannion (Glasgow Caledonian University, UK) and Hermann Kaindl (Vienna University of Technology, Austria)

Prof Michael Mannion is Dean of School of Computing and Mathematical Sciences, Glasgow Caledonian University, Glasgow, Scotland, UK. He has a BSc in Computer Science from Brunel University and a PhD in Artificial



Intelligence from Bristol University. He has several years' software engineering industrial experience, is a former Chairman of the British Computer Society Special Interest Group in Software Reuse and has served as a member of the British National Space Centre's Software Steering Group. He lectured for 8 years at Napier University, Edinburgh and then in 2000 took up a Professorship at Glasgow Caledonian University. He has published more than 40 papers.



Prof Hermann Kaindl About a year ago, he joined the Institute of Computer Technology at the Vienna University of Technology in Vienna, Austria. Prior to moving to academia, he was a senior consultant with the division of program and systems engineering at Siemens AG Austria. There he gained more than 24 years of industrial experience in software development. His current research interests include software engineering with a focus on requirements engineering, and human-computer interaction as it relates to scenario-based design. He has published three books and more than seventy papers in refereed journals, books and conference proceedings. He is a senior member of the IEEE, a member of the ACM and INCOSE, and is on the executive board of the Austrian Society for Artificial Intelligence.

Abstract: Reuse and requirements are very important for efficient and successful systems development. However there are many open issues about performing them well, in particular the reuse of requirements. This tutorial presents the experiences of requirements reuse using a Method for Requirements Authoring and Management (MRAM).

For modern, highly complex, high reliability systems, the need for properly structured, carefully controlled requirements specifications, which are understandable, complete and consistent is essential in order for the resultant computer-based system to be delivered on time, within budget and to the desired high level of quality. One approach to managing these problems is to establish a pool of reusable requirements and to construct the requirements for a new system by making a selection from the pool. A concern of this approach is the efficient and clean selection of a consistent combination of requirements. A consistent combination is one in which the requirements selected satisfy any constraints imposed by the pool of reusable requirements.

MRAM is a method for establishing and selecting from product line requirements that addresses this concern. Using MRAM means the management of the requirements definition process is more effective and efficient, producing more accurate and complete requirements documents. TRAM (Tool for Requirements Authoring and Management) is a software tool to support MRAM that utilises current proven office technology (MS-Word, MS-Access). The tutorial presents the results of MRAM/TRAM as it has been applied to a real-world application.

T-4 Requirements and Creativity: How To Integrate Invention into Requirements Engineering

Time: Sep. 6(Mon) Afternoon

Lecturers: Suzanne Robertson and James Robertson (The Atlantic Systems Guild, UK)

For Suzanne Robertson, please see T2.

James Robertson is a consultant, teacher, author, project leader whose area of concern is the requirements for products, and the contribution that good requirements make to successful projects. James is a leading proponent of the principle of introducing creativity into the requirements process. His controversial article “Eureka: Why Analysts Should Invent Requirements” in IEEE Software, July 2002, has been widely quoted and discussed.



Before becoming a systems engineer, James trained as an architect and his experience in that profession provides inspiration for his work on innovation and creativity. He is co-author of Mastering the Requirements Process (Addison-Wesley 1999), the Volere <http://www.volere.co.uk> approach to requirements engineering, and the forthcoming book, Requirements-Led Project Management: Discovering David's Slingshot, that addresses the question of how project managers can take advantage of requirements to plan and steer the project.

He is also a principal and founder of The Atlantic Systems Guild <http://www.systemsguild.com> a think tank known for its research into new systems engineering techniques and its practice in making good ideas more accessible.

Abstract: The traditional job of a requirements analyst is to ask people what they want and need, and specify those requirements. But people often do not know what they want, neither do they know what is possible nor can imagine it. Naturally they ask for the obvious requirements – those that fit within their experience. However, the most useful products have come about not by small, incremental improvements, but by a quantum leap forward – by an invention.

Instead of just accepting the status quo, requirements engineers must be able to invent something better, something to improve their client's work. This tutorial is a practical guide for how to use creative techniques to invent requirements that result in more useful, usable and competitive products.

The contents of this tutorial are based on:

- Series of creative workshops for Eurocontrol air traffic management systems run by Neil Maiden, The Centre for HCI Design, City University, London and Suzanne Robertson, The Atlantic Systems Guild, London.
- Research and project work of James Robertson, the Atlantic Systems Guild, London into the link between requirements and innovation. Including discoveries on why requirements analysts should be inventors and how to use inventive requirements triggers.

Sections of the tutorial are:

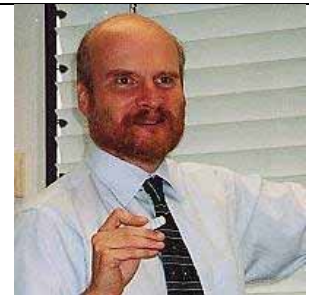
1. Creativity – what is it and why do we need it?: Principles from the field of creative thinking.

2. Inventive Requirements Triggers: How to use inventive requirements triggers: Service, Ideas, Speed, Information, Technology, Choices, Participation, Design, Origin, Senses, to lead to competitive and innovative products.
3. The story of the air traffic control workshops: How we ran the workshops. Quantification of the results.
4. Experts and Analogies: How to use experts from other fields and analogical reasoning to generate creative ideas.
5. Integrating Creativity: How to integrate creativity and invention into your existing requirements engineering process. Planning and running creative workshops.
6. Creative Action: Key ideas, lessons learned and action plan.

T-5 Developing Practical Scenarios

Time: Sep. 7 (Tue), Morning

Lecturers: Ian Alexander (Consultant, UK) and Alistair Mavin (Praxis, UK)
 He is the author of the JBA 3-Day Requirements Engineering Workshop, and is co-author of JBA's 3-Day Systems Engineering Course. He is accredited as an instructor for Telelogic's 2-Day Applying DOORS, DXL, and Requirements Methodology courses, and for the Atlantic Systems Guild's 3-Day Mastering the Requirements Process course. He aims to improve the requirements engineering process using scenarios, and is currently working alongside the DaimlerChrysler Research & Technology Centre, Ulm, on reusing requirements between models of car. He created the Scenario Plus for Use Cases toolkit, available from <http://www.scenarioplus.org.uk>. His book *Writing Better Requirements* (Addison-Wesley 2002) emphasises that requirements come from people, and gives practical advice on how to discover and structure them. He has written many papers on the people-facing side of requirements engineering for both popular and refereed journals. He helps to run the BCS Requirements Engineering Specialist Group and the IEE Professional Network for Systems Engineers. He is a Chartered Engineer.



Alistair Mavin is a Systems Engineer with Praxis Critical Systems Limited, a UK based company specialising in requirements, systems, software and safety engineering (<http://www.praxis-cs.co.uk/>). He has worked on requirements engineering projects in a range of industries including aerospace, defence, air traffic management, rail, automotive and local government. Alistair regularly uses scenario analysis in his client project work with the Praxis REVEAL® requirements engineering method. He has been involved in a number of capability enhancement projects using REVEAL, which have the aim of improving the engineering capability within the client organisation. He is a qualified trainer for the REVEAL 4-day practitioners training course and is a member of the British Computer Society Requirements Engineering Specialist Group.



Many large engineering projects fail to provide all the necessary functionality, run over budget, or are delivered late. One way to address these problems is to define what the system must do earlier and more carefully. Engineers therefore need to provide a more effective means of communicating stakeholder needs. One of the most powerful mechanisms for achieving this is the Scenario. Scenarios are simple, human stories, usually structured as lists of actions.

In contrast to Alistair Sutcliffe's admirable tour of Scenario approaches at RE'03, this tutorial focuses on

disseminating the practical skills needed to develop scenarios on industrial projects, such as for embedded systems. This hands-on tutorial gives participants the theory and allows them to experience the practice of scenario analysis. We offer a simple, well-defined method based on an iterative, human-centred process. In our process, we begin by defining the mission and objectives. Next, we identify stakeholders and their viewpoints. Then we turn our attention to scenarios, and successively tell the primary stories as scenarios; explore exception scenarios; and validate the discovered scenarios. A single example of an embedded (hardware/software) system is used in all the exercises. This is a multi-purpose portable device for outdoor sportspeople such as climbers, walkers, campers and fishermen, to help them with navigating, receiving weather forecasts, obtaining help if in distress, and other functions thought of by participants in the exercises. Both instructors will supply background papers on scenarios to tutorial participants. We intend to provide a Japanese translation of the tutorial to encourage local participation.

T-6 Software Traceability

Time: Sep. 7 (Tue), Afternoon

Lecturers: George Spanoudakis and Andrea Zisman (City University London, UK)

Dr. George Spanoudakis is a senior lecturer in the Department of Computing of The City University in London, deputy head of this department and senior tutor for research in it. He holds BSc, MSc and PhD degrees in Computer Science and has been a visiting associate professor in the Department of Computer Science of the University of Crete (2000) and visiting lecturer in the Department of Information Systems of the London School of Economics (2000). George has served in the program committees of numerous international conferences and workshops in the field of software engineering and has acted as a reviewer for international scientific journals including the IEEE Transactions on Software Engineering and ACM Transactions on Software Engineering and Methodology. Recently, George has organised two international workshops in the area of traceability (in conjunction with ASE '02 and ASE '03) and he is currently editing a special issue of the International Journal in Software Engineering and Knowledge Engineering in software traceability. His research in the area of software engineering is related to multi-perspective software modelling, managing inconsistencies in software models, software traceability and measurement, and software reuse. George has over 45 publications in these areas. For more information see: <http://www soi.city.ac.uk/~gespan>.



Dr. Andrea Zisman is a senior lecturer in the Department of Computing of City University, London. She holds a PhD degree in Computer Science, a MSc degree in Applied Mathematics to Computer Science, and a B. Sc degree in Computer Science. Prior to this position she was a research fellow at the Department of Computer Science at University College London, UK and has worked as a software system consultant, developer and analyst. Andrea was a visiting researcher at AT&T Labs Research, USA (2003). Andrea has been research-active in the areas of automated software engineering and management of distributed data where she has published extensively. Her research has been concerned with traceability and consistency management of software artefacts, validation of distributed systems, interoperability of distributed data sources, and XML and web services



applications. Andrea has served in the organising and program committees of various international conferences and workshops, has acted as a reviewer for international journals, and has given tutorials in various international conferences. She has organised two international workshops in the area of traceability (TEFSE'2002 and TEFSE'2003) and is co-editing a special issue of the International Journal in Software Engineering and Knowledge Engineering in software traceability. For more information see: <http://www soi.city.ac.uk/~zisman>.

Software traceability – that is the ability to relate artefacts which are created during the development of a software system (e.g. requirements, design and code artefacts) with each other, the stakeholders that created them, and/or the rationale underpinning their exact form – has been recognised as a significant capability in the software development and maintenance process, and as an important factor for the quality of the final product.

Traceability information can be used to support the analysis of the implications and integration of changes requested in the system development process; the maintenance and evolution of software systems and documentation; the reuse of software systems and their components; and the inspection and testing of software systems.

The software and system engineering communities have long standing interests in the area of traceability and have developed numerous approaches and techniques to establish and maintain traceability. More specifically, research in this area has been mainly concerned with the study and definition of different types of traceability relations; the generation of these relations; the development of architectures, tools, and environments for managing traceability information; and the empirical investigation of organisational practices regarding the establishment and deployment of traceability information in the software development life cycle.

In this tutorial, we present an overview of the state of the art and practice in software traceability, discuss the main scientific and technological advances in this area, present the available technologies and their contributions, and identify issues that require further research in the field. The tutorial is intended for practitioners, managers, researchers, and teachers with interests in software development and the management of software systems and development processes. The tutorial may also be attended by students that follow a university programme in computing at all levels.

T-7 Financially Informed Requirements Prioritization

Time: Sep. 7 (Tue), Afternoon

Lecturer: Jane Cleland-Huang (DePaul University, USA)

Dr. Jane Cleland-Huang is an Assistant Professor at DePaul University's School of Computer Science, Telecommunications, and Information Systems. Her research interests include Value-based Software Engineering, Requirements Traceability, and Process Models. She is Associate Director of the Institute of Software Engineering in Chicago, and director of the DePaul Center for Applied Requirements. She previously worked in industry as a developer, systems analyst, and consultant. Dr. Cleland-Huang recently co-authored the book “Software by Numbers, low-risk, high-return development”, which describes a financially informed approach to software development, and has been highly endorsed by both agile and planned development communities. She is a member of the IEEE Computer Society.



This tutorial introduces a financially responsible approach to requirements prioritization known as

the Incremental Funding Method (IFM) and described in the book “Software by Numbers: Low-risk, High-Return Development”. Attendees will learn how to group requirements into “chunks” of revenue-generating functionality known as Minimal Marketable Features (MMFs), and how to carefully sequence those MMFs in order to maximize the overall value of the project, reduce initial funding investments, and manipulate other project metrics such as the time needed for a project to reach break-even status. A gentle introduction to financial analysis will also equip participants to analyze and understand the impact of other requirements prioritization decisions upon the financial returns of a project. This process is applicable within any iterative development approach such as the Rational Unified Process or within a more agile environment.

In addition to introducing the concepts and techniques needed to implement IFM within a real project, the tutorial will include hands-on activities in which participants will apply the principles they have learned. All attendees will receive tutorial notes, as well as proprietary software tools to take home with them.

The tutorial is targeted to developers, project managers, process specialists, CIOs, CTOs, and CFOs, or anyone interested in learning more about the role of requirements prioritization as a financial value-creating activity.

For additional information about this tutorial please visit <http://facweb.cs.depaul.edu/jhuang/RE04/Tutorial.htm>

J-1 Introduction to Scenario Analysis [In Japanese]

Time: Sep. 6(Mon) Afternoon

Lecturer: Kentaro Go (Yamanashi University, Japan)

J-2 Requirements Traceability Methodology and Its Application to UML [In Japanese]

Time: Sep. 7 (Tue), Morning

Lecturers: Shuichiro Yamamoto Tomoyuki Azuma (NTT Data, Japan)

Shuichiro Yamamoto:

He joined NTT in 1979. Dr.Yamamoto received a B.S. in information engineering from Nagoya Institute of Technology in 1977, and an M.S. and D.R. in information engineering from Nagoya University in 1979 and 2000. He moved NTT DATA Corporation in 2002. He is a member of IEICE, IPSJ, JSAI. He was formerly a chairman of SIG Knowledge Based Software Engineering of IEICE.



Tomoyuki Azuma:

He joined NTT DATA in 2001. Mr. Azuma received a B.S. and M.S. in administration engineering from Keio University in 1999 and 2001.

