

AWRE'04 – List of Abstracts

SESSION 1 : METHODS AND TECHNIQUES

Evaluation Framework for Tools that Manage Requirements Inconsistency

Jyothi Iyer and Debbie Richards

Department of Computing, Division of ICS, Macquarie University, Sydney, Australia , 2109
{jiyer,richards}@ics.mq.edu.au

Management of inconsistency in requirements is an important but effort intensive task. Automated tools promise greater reliability and less effort. However, after review of a number of surveys of current commercial requirements engineering tools we found that while some forms of inconsistency are considered in the evaluations, such as version control and spell checking, the surveys themselves do not address the deeper concerns that inconsistency raises. Hence there is a need for a framework to facilitate comparative evaluation of existing tools that could also serve as a requirements specification for a new or enhanced tool. Our framework and underlying reasoning are presented.

Connecting Role Activity Diagrams to the Problem Frames Approach

Karl Cox¹, Keith Phalp², Aybüke Aurum⁴, Steven J. Bleistein^{1,3}, June Verner¹

¹National ICT Australia, Sydney, karl.cox, steven.bleistein, june.verner@nicta.com.au

²ESERG, Bournemouth University, UK, kphalp@bmath.at.uk

³School of Computer Science and Engineering

⁴School of Information, Systems and Management, aybuke@unsw.edu.au

University of New South Wales, Sydney, Australia

Capturing process information in requirements is not a simple task. To alleviate this difficulty, we propose a Problem Frames approach that takes elements of process models and maps them to Problem Frames. To show where requirements have an effect on particular contextual entities, we use an extended problem diagram. From here we attempt to derive appropriate Problem Frames. The paper reports on an industrial study for a major online brokerage and financial system that describes the approach.

Capturing Informal Requirements as Formal Models

Rubens Fernandes

University of South Australia
Rubens.Fernandes@postgrads.unisa.edu.au

Alex J. Cowie

University of South Australia
Alex.Cowie@unisa.edu.au

We present a requirements engineering tool and associated methodology that converts natural language and graphical requirements to models expressed in a process algebra formalism. Natural language requirements are automatically converted into the concrete syntax of the process algebra using recon_gurable parsing and transformations. Graphical requirements in the form of _nite state diagrams are changed into textual form before parsing and generation of the formal models. Analysis of the formal models by a requirements engineer can lead to iterative re_nement of the natural language requirements and the tool con_guration. The motivation for this work comes from the possible reduction in reworking when formal methods are used early in the development life cycle. Our framework allows the coexistence of an user-friendly document that is suitable for non-technical users and a formal model suitable for veri_cation and reasoning.

A Context-free Requirements Grammar to Facilitate Automatic Assessment

William Scott

Stephen Cook

Systems Engineering and Evaluation Centre,
University of South Australia
William.Scott@unisa.edu.au

Poor requirements are often cited as a leading cause of project failures. Preparing a good set of requirements presents challenges at many levels. The approach of the authors is to work from the bottom up by formulating a stylized English grammar that can be used to describe requirements in a more precise way. The grammar was derived from an examination for the literary practices of requirements engineers, scholarship of English Grammar, and a study of mathematical linguistics. The grammar can be used in a variety of ways to assist the engineer to improve the overall quality of a set of requirements, the most significant of which is to support automatic reasoning.

SESSION 2 : SOCIO-TECHNICAL ISSUES

Defining User Requirements for a Large Corporate Data Warehouse: An Experiential Case Study

Mark Dale

Department of Information Systems
Melbourne University
markd@staff.dis.unimelb.edu.au

A large corporate Data Warehouse supporting the operational and strategic information requirements of an international financial services organisation is in need of significant improvement. The current technical infrastructure, data model, and system processes and controls are limiting access to the system and relevant customer, product and channel data, the quality and detail of data, the timeliness and delivery of that data, and other operational and strategic information requirements. This paper includes a discussion of the social, organisational and political forces impacting the requirements gathering and prioritisation for a large strategic information system. The article is written as a experiential case study and will be very useful to a number of different audiences including software and requirements engineering students, professional system analysts and software engineers as well as business management students and professionals.

Influence of Stakeholder Communication on Consensus Making in Requirements Negotiation

Jenny Price

jennylou@bigpond.net.au

Jacob L. Cybulski

jlcybuls@deakin.edu.au

+61 3 9244 6847

School of Information Systems

Deakin University

221 Burwood Highway

Burwood, Victoria 3125, Australia

Consensus building is essential in aligning multiple stakeholder viewpoints which are commonly emerging during the process of requirements elicitation, analysis and validation [15]. In fact, agreement among iverse groups of stakeholders is deemed prerequisite to establishing project cooperation and collaboration. While individuals frequently rank the importance of their personal goals over the goals of an organisation, having an appropriate strategy for finding common ground among individuals is crucial to any project's success. This paper therefore explores factors that influence the consensus making process and develops a model that helps with understanding the role of consensus specifically during the elicitation phase of the Requirements Engineering process.

Issues in IT Service-oriented Requirements Engineering

Sharman Lichtenstein

sharman.lichtenstein@deakin.edu.au

Lemai Nguyen

lemai.nguyen@deakin.edu.au

Alexia Hunter

alexia_h@bigpond.net.au

School of Information Systems, Faculty of Business and Law, Deakin University Australia

Corporate information technology (IT) management is increasingly service-oriented, offering continuous evaluation and improvement of application, communication, delivery and support services to internal and external customers. Service-oriented requirements engineering (SoRE) plays a significant role in identifying and specifying service requirements, formally defined through service-level agreements (SLAs). However, the new frameworks and approaches emerging to guide these developments have not yet addressed how requirements for such services can be effectively developed, nor identified the diverse issues involved. We report a case study of a web services team development of requirements for an internal Service Desk service. The study revealed five main issues of concern when developing service provider requirements: service roles, responsibilities and accountability; service performance metrics; resolution of conflicting stakeholder service requirements; customer acceptance of service change; and service provider team structure. This study suggests that in the new IT services era, new techniques and approaches are needed for eliciting and determining provider and customer requirements that involve key stakeholder groups equitably and more closely negotiate the sometimes-conflicting provider and customer needs.

Towards Interface Specification and Design Guidelines to raise User Awareness of Application Security

Rodney MacDonald

School of Information Systems

Deakin University, Victoria, AUSTRALIA

rodneym@deakin.edu.au

Ross Smith

School of Information Systems

Deakin University, Victoria, AUSTRALIA

ross.smith@deakin.edu.au

While the discipline of interface specification and design has received substantial attention in the existing body of academic and professional literature, there has been little research conducted into the specification and design of interfaces that allow users to interact with systems in a secure manner. As the awareness of security issues increases amongst the general user community, together with an apparently ever increasing growth in the number of information systems security breaches, the specification and design of "secure interfaces" will

become a critical development consideration. This research aims to determine how the specification and design principles identified in the extant body of literature, might be fused with the development practices currently employed within two focus organisations (case study organisations), to synthesise and partially validate a set of interface specification/design guidelines for use when developing interfaces with the ability to raise the security awareness of users.

SESSION 3 – TOWARDS BETTER PRACTICE & EDUCATION

Requirements Engineering: a Close Look at Industry Needs and Model Curricula

Oliver Minor

University of Koblenz-Landau, Germany

minor@uni-koblenz.de

Jocelyn Armarego

Murdoch University, Western Australia

jocelyn@eng.murdoch.edu.au

Research endeavours in software development have found that failures and deficiencies of software systems are often rooted in the requirements activities undertaken. One possible cause for poor requirements activities is the appropriateness of the education of those engaged with the requirements component of software development. This education is largely based on model curricula used as guidelines. This paper summarises the results of a study of the requirements component of model curricula in the disciplines of computer science, information systems and software engineering. The requirements components are compared to the opinions of a small but representative group of practitioners, assembled through personal interviews. The results reveal that the model curricula address to a high degree the expectations practitioners have of the formal education in preparing for requirements activities. However, the results also show that practitioners see shortcomings in formal education, particularly with respect to more generic skills, such as communication and team skills.

Requirements Engineering and Software Project Success: An Industrial Survey in Australia and the U.S.

June Verner

Empirical Software Engineering

National ICT Australia

AustralianTechnology Park

june.verner@nicta.com.au

Karl Cox

Empirical Software Engineering

National ICT Australia

AustralianTechnology Park

karl.cox@nicta.com.au

Steven Bleistein

Computer Science and Engineering,

University of New South Wales &

National ICT Australia

steven.bleistein@nicta.com.au

Narcisco Cerpa

Departamento de Ingeniería de Sistemas, Universidad de Talca,

Camino Los Niches Km. 1, Curicó, Chile

ncerpa@utalca.cl

We surveyed a number of software practitioners regarding their software development practices. Because requirements engineering is recognized as critical to successful software project part of our survey explored requirements engineering activities during recent software projects. The relationship between requirements practices and software project outcomes enables us to better understand requirements issues and their relationship with project success. Our respondents were almost exclusively involved with in-house software for business organizations in the US and Australia We asked three sets of questions directly related to requirements: 1) the requirements, 2) the customer/users, and 3) the project management. The most significant factor from each was 1) the requirements were good, 2) there was a high level of Customer/User involvement, and 3) the requirements were managed effectively. The best predictor of successful project outcomes (93% correctly predicted) was that the requirements were good together with the requirements were managed effectively. We also explore project scope and use of an appropriate software development methodology. Our survey shows that effective project management is fundamental to effective requirements engineering.

Learning Requirements Engineering within an Engineering Ethos

Jocelyn Armarego

School of Engineering Science, Murdoch University

Western Australia

jocelyn@eng.murdoch.edu.au

An interest in educating software developers within an engineering ethos may not align well with the characteristics of the discipline, nor address the underlying concerns of software practitioners. Education for software development needs to focus on creativity, adaptability and the ability to transfer knowledge. A change in the way learning is undertaken in a core Software Engineering unit within a university's engineering program demonstrates one attempt to provide students with a solid foundation in subject matter while at the same time exposing them to these real-world characteristics. It provides students with a process to deal with problems within a metacognitive-rich framework that makes complexity apparent and lets students deal with it adaptively. The results indicate that, while the approach is appropriate, student-learning characteristics need to be investigated further, so that the two aspects of learning software (academic and practitioner) may be aligned more closely.

Global Software Development: The Next RE Frontier?

Merete Crofts

School of Information Systems

Deakin University, Victoria, Australia

merete.crofts@deakin.edu.au

Ross Smith

School of Information Systems

Deakin University, Victoria, Australia

ross.smith@deakin.edu.au

Bardo Fraunholz

School of Information Systems

Deakin University, Victoria, Australia

bardo.fraunholz@deakin.edu.au

Requirements Engineers face an emerging set of challenges, which compound the traditional Requirements Engineering (RE) challenges (stakeholder identification, domain expertise, communication, analytic skills, problem solving, ...) that have arguably still not been fully addressed. This is the challenge of RE in the world of global software development, with requirements teams working in virtual mode (possibly on different continents), with the software having to operate in multiple contexts, addressing the needs of different cultures and legal jurisdictions, and having to build sales in different marketplaces. Further the need arises to specify software that is progressively enhanced through regular releases, rather than the "green field" specification of products.

This theoretical paper introduces these challenges, and presents an initial selection of theoretical models, drawn from many and varied source disciplines, which might be employed to gain insight into various features of RE in support of global software development. To illustrate the potential relevance of this selection of models, a longitudinal case study with a recently identified software developer, to follow the specification and subsequent roll-out of a future release of a software product for sale globally, is introduced. Features of the situation faced by that organisation are highlighted, to illustrate the potential relevance of the diverse models that have been identified.

SESSION 4 : RE PROCESS

Measuring and Influencing Requirements Engineering Process Quality in Organizations

Bhavani Palyagar

Information and Communication Sciences Macquarie

University NSW

bpalyaga@ics.mq.edu.au

Requirements Engineering (RE) is a process for determining stakeholder needs during the development of a software intensive system. Research indicates that RE process quality critically influences the success of the Software Development Life Cycle (SDLC). This places significant importance on RE process measurement. Unfortunately, however, RE process quality measurement is neglected, possibly because it is difficult to measure the quality of RE processes, and therefore, to determine what constitutes a quality RE process across different types of organizations and software development projects. This makes it difficult to formalize RE processes from a quality perspective. I attempt to address this issue by proposing a definition of RE process quality that is applicable across different software development projects. I then propose some high and low-level measures of RE process quality and present a framework that identifies factors that influence RE process quality, followed by a snapshot of a case study. I conclude with a discussion of the implications of this framework both for the quality of the requirements, and for software in general.

Deriving Process Specifications from Augmented Data Models

Götz Botterweck

Institute for IS Research

University of Koblenz-Landau

Universitaetsstr. 1

56070 Koblenz, Germany

E-mail botterwe@uni-koblenz.de

Phone +49 261 2872531

Fax +49 261 2871002531

Carlo Simon

Institute for Management

University of Koblenz-Landau

Universitaetsstr. 1

56070 Koblenz, Germany

E-mail simon@uni-koblenz.de

Phone +49 261 2872861

Fax +49 261 2872851

This paper presents an approach which enables a system analyst to derive process models from data models. To accomplish this, existing data models are augmented with additional information regarding the intended process goal and the dependencies which have to be fulfilled.

After giving an introductory example and proposing a notation for the augmentations, a simple algorithm is sketched. This derives a process model consisting of all steps necessary to fulfil the specified goal and the related dependencies.

Overview of a Gap-driven Evolution Process

Anne Etien, Colette Rolland, Camille Salinesi

CRI – University Paris 1, 90 rue de Tolbiac, 75013

Paris, France

Phone: + 33 1 44 07 86 34, Fax: + 33 1 44 07 89 54

{aetien, rolland, camille}@univ-paris1.fr

The construction of an Information System takes several years. Part of the solution is in the improvement of methods technique and tools to adapt systems.

In this context, we propose an evolution process at the requirement level based on the concept of gap. This approach allows, on the one hand, thanks to the abstraction mechanism to focus only on goals and ways to achieve them avoiding the details. On the other hand, gap modelling allows to only focus on what makes the future system different from the legacy and avoid re-specifying things that remain stable. We describe in details the gap driven evolution process and illustrate it using a hotel room booking case study.

Requirements Engineering for Generic Process Aware Information Systems

Roger Tagg

*School of Computer & Information Science,
University of South Australia, Mawson Lakes
Roger.Tagg@unisa.edu.au*

This is a position paper to initiate discussion on how – and whether – traditional requirements engineering (RE) can be applied to modern information systems (IS) where only part of the business process is implemented “up front”, and where the IS has to cater for the continual addition and amendment of processing logic. A categorization of modern information systems is proposed, and a number of variations on requirements engineering are suggested to cover the proposed categories.