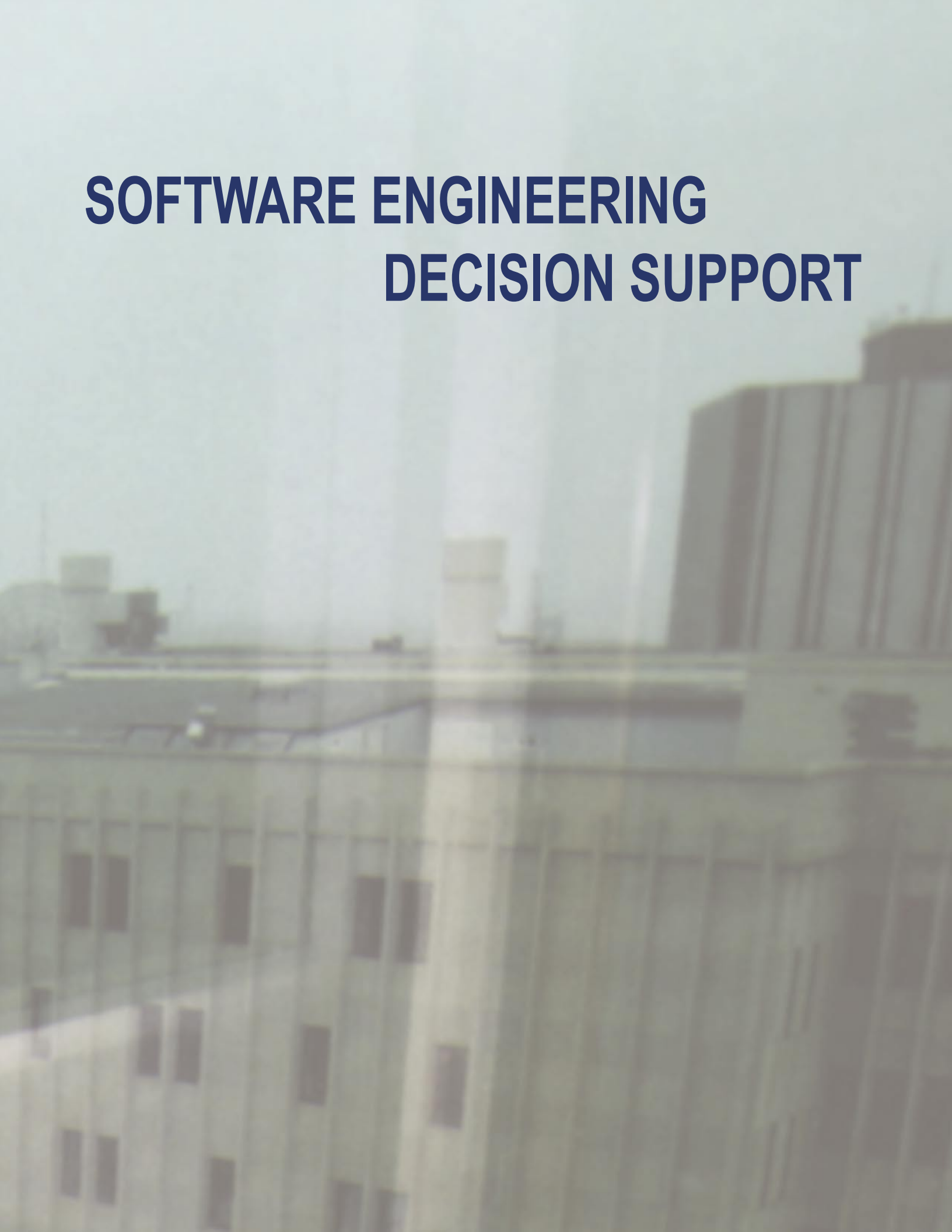
SOFTWARE ENGINEERING
DECISION SUPPORT

GUENTHER RUHE

iCORE Professor
Department of Computer Science, Department of Electrical and Computer Engineering and Industrial Research Chair Software Engineering, University of Calgary
http://sern.ucalgary.ca/~ruhe

Software is quickly becoming a dominant factor for business success in an increasing number of products and services in the telecommunication, health care, automotive, banking, insurance, and manufacturing industries. A longstanding goal in software development is to construct systems that are easily modified and extended. The proliferation of software in everyday life (for example, embedded systems found in automobiles, mobile phones, and television sets) has increased the conformity and invisibility of software. The capability to react better and faster to changes in requirements, technologies and policies in all application domains will significantly improve the competitiveness of software dependent industries.

EXECUTIVE SUMMARY

Intelligent Decision Support Systems help decision makers in using communications technologies, data, documents, domain knowledge, and models to identify and solve problems. Main achievements of the Laboratory of Software Engineering Decision Support over the last year have been breakthrough results in the area of intelligent support, in particular for design and analysis of releases in incremental software development, analysis of software requirements, and selection of components-of-the shelf (COTS) software products. We are following a multidisciplinary research approach and have integrated concepts and approaches from software engineering, decision science, artificial intelligence, knowledge management, and computational intelligence to achieve novel results. Research in the reporting year has resulted in six journal publications, fifteen publications at conferences or workshops, six book chapters (partially in progress). A book titled *Software Engineering Decision Support – Methodology, Tools and Applications* is in preparation to be published by CRC Press.

A new generation of intelligent decision support has been designed, implemented and initially evaluated. ReleasePlanner™ (see www.releaseplanner.com) is a tool suite that provides a flexible and web-based tool support for assigning requirements or features to releases such that most important risk, resource, and budget constraints are fulfilled. ReleasePlanner™ is based on a solution approach called EVOLVE* that combines the computational strength of optimization algorithms algorithms with the flexibility of an iterative solution method. As a web-based tool, it

allows input and interaction even with remote stakeholders (sales representative, shareholder, developer, different classes of user). This results in a much better chance to actually achieve high customer satisfaction with the products to be developed. The technology was approved by the Inno-Centre Alberta Advisory Board to enter the business mentorship program. This is a highly competitive entrepreneurship program. So far, 11 out of more than 300 evaluated technologies have been selected. Both US and Canada patent protection for release planning technology is pending.

During the reporting period, further progress has been achieved in creating a core team of researchers and in establishing and enhancing national and international collaboration. Industrial collaboration projects have been conducted with DaimlerChrysler, Corel, Nortel Networks and Brycol Consulting. A NSERC CRD grant was conditionally approved. A NSERC Strategic Research Grant proposal and a proposal for the new program "Idea to Innovation" were submitted. Finally, the Software Engineering Consulting Consortium (SECCO) was formed. This is an initiative with the objective of fostering and encouraging links between the industry and graduate students in the software engineering discipline engaged in research activities.

RESEARCH PROGRAM OVERVIEW

Objectives and Results

The future vision of life is a society of citizens whose communal interactions and access to personal knowledge and services are enabled, supported and strengthened by computing that is available in practically every environment: in the home, at the workplace, in public spaces such as libraries, airports, town squares, in public and private vehicles, and through developments of mobile phones carried on the person. The quality of life in our 21st century society depends heavily on products and services that are enabled, supported, and strengthened by software engineering technologies and distributed systems. The next step of the Internet and the mobile revolution requires software solutions with extreme demands on their quality and predictability.

There are very good reasons for offering support for making decisions at the various stages of software development and evolution. The current maturity of software engineering decision-making is low. Decisions are made ad hoc, not relying on validated models and sound methodology. Decisions are hard to understand and far from being optimal in terms of their quality. What can be expected from decision support in the area of software engineering is higher decision quality, improved communication between all involved parties, increased productivity, time savings and improved customer satisfaction.

Our research goals can be structured into four areas:

- Advancing SEDS methodology by multidisciplinary research integrating software engineering with decision support, computational intelligence, empirical research and knowledge management;
- Developing intelligence-based methods and techniques for specific software engineering decision problems;
- Design and implementation of tools for intelligent decision support for release and design decisions and;
- Evaluation of the impact of intelligent decision support and the related tool support.

Main Achievements

The core team is growing quantitatively and qualitatively. It now consists of one post-doctoral fellow, three PhD students (two more have accepted an offer), seven graduate students (two more already accepted an offer), four undergraduate students and two support staff.

Technically, we have achieved the following main results:

Hybrid Intelligence in Release Planning

The advantage of the human intelligence based approach is that it is able to better handle soft and implicit objectives and constraints. The advantage of computer-based approach is exactly where the human based approach fails: to cover a large portion of the solutions space. A hybrid approach is described to combine the strength of both human and computational intelligence.

The overall architecture of EVOLVE* is designed as an iterative and evolutionary procedure mediating between the real world problem of software release planning, the available tools of computational intelligence for handling explicit knowledge and crisp data, and the involvement of human intelligence for tackling tacit knowledge and fuzzy data. At all iterations, three phases are passed:

- Phase 1 – Modeling: Formal description of the (changing) real world to make it suitable for computational intelligence based solution techniques. This includes the definition of all decision variables, as well as their dependencies and constraints, and the description of what is, or contributes to, the "goodness" of a solution. Other data, such as stakeholder evaluation of all requirements, are also part of modeling.
- Phase 2 – Exploration: Application of computational techniques to explore the solution space, to generate and evaluate solution alternatives. Exploration phase is mainly based on evolutionary computing and optimization algorithms.
- Phase 3 – Consolidation: Human decision maker is invited to investigate current solution alternatives. This contributes to the understanding of the problem and results in modifying parts of the underlying model or in some local decisions (for example, pre-assigning some requirements to a release). Typically, these decisions reduce the size and complexity of the problem for the next iteration.

EVOLVE* was extended to problems with fuzzy objectives and constraints and to problems comprising resource, financial and risk constraints.

Tool Suite ReleasePlanner™

ReleasePlanner™ (see www.releaseplanner.com) is a tool suite that provides a flexible and web-based tool support for assigning requirements or features to releases such that the most important risk, resource, and budget constraints are fulfilled. The tool was developed at the Laboratory for Software Engineering Decision Support http://www.seng-decisionsupport.ucalgary.ca at the University of Calgary. It can be used in different user modes and aims in providing intelligent decision support for any kind of iterative development. It addresses the wicked character of the problem by an approach integrating computational and human intelligence.

At all iterations, optimization algorithms are applied to determine the most promising solutions of constrained release planning.

Main features of the tool are:

- As a web-based tool, it allows input and interaction even with remote stakeholders (sales representatives, shareholder, developer, different classes of user). This results in a much better chance to actually achieve high customer satisfaction with the products to be developed.
- Ease of customization for the different types of users and the different usage scenarios. After defining the usage mode, only respective functionality is activated. This increases user-friendliness.
- Compatibility with commercial tools addressing features management (as a prerequisite for using release planning) and project management (for actually planning and controlling the performance of the plan).
- Computational strength of its core algorithms that are based on innovative ideas (optimization

Guenther Ruhe and some research team members at the 2004 Banff Informatics Summit

(From left to right)
Wei Shen
Gengsheng Du
Joseph Momoh
Jingzhou Li
Omalade Saliu
Kornelia Streb
Guenther Ruhe
Abdallah Mohamed
Sebastian Maurice
An Ngo-The
Zhizhong Li

algorithm) and their novel implementation. This results in very effective procedures to generate near-optimal solutions for large-scale problems of high complexity.

- Applicability to a broad range of usage scenarios in different application domains (IT, logistics, automotive, telecommunication, pharmacy).
- Modeling support is offered. As validated models are a mandatory prerequisite for meaningful results, different objectives different schema of evaluation, and different types of constraints (effort, resource, precedence, finance, risk) can be included.
- Process-driven usage guidelines as well as experience and knowledge support: There is a process framework for how to conduct planning. Each task is the object of adding or retrieving related knowledge and experience (Knowledge management component).

RESEARCH PROJECTS

Intelligent Support for Software Release Planning

One of the most prominent issues involved in incremental software development is to decide upon the most appropriate software release plans taking into account all explicit and implicit objectives and constraints. Such decisions have become even more complicated in the presence of large number of stakeholders such as different groups of users, managers, or developers. However, early involvement of customers and understanding of their real needs is one of the core success factors of software business.

Guenther Ruhe

A six-step process model for release planning was designed and evaluated. Release planning is a learning and improvement process inspired by the Quality Improvement Paradigm. Implementing this process is the emphasis of this tool support. The use of the intelligent decision support tool ReleasePlanner™ is presented by comparing a baseline scenario reflecting current state-of-the practice of release planning with a supposed improvement scenario obtained after usage of the tool. Initial experience from a real-world environment at iGrafx Corel Inc. is used to validate the improvement scenario.

Hybrid Intelligence

Neither human nor computational intelligence alone are able to provide appropriate decision support for the problem under consideration. The advantage of the human intelligence based approach is that it is able to better handle soft and implicit objectives and constraints. The advantage of computer-based approach is exactly where the human based approach fails: to cover a large portion of the solutions space. The computational complexity of the problem makes it impossible for the decision maker to have a reasonable perception of the set of possible solutions and to evaluate and prioritize different solution alternatives. To solve this inherent dilemma, we propose a hybrid approach integrating the advantages of both principal approaches.

Integration of human and computational intelligence can be understood in two directions. Firstly, from the integration of powerful solution approaches as offered by computational intelligence, we can expect solutions of formally defined problems of large size and complexity. Secondly, decision support needs the inclusion of human intelligence to include tacit and subjective components into the process of selecting the most promising solutions. Typically, from this involvement, new questions are raised. Computational and human intelligence are constantly interacting with each other.

Quantitative Studies in Software Release Planning under Risk and Resource Constraints

Delivering software in an incremental fashion implicitly reduces many of the risks associated with delivering large software projects. However, adopting a process, where requirements are delivered in releases means decisions have to be made on which requirements should be delivered in which release.

A method called EVOLVE+ was developed. It is based on genetic algorithms and aimed at the evolutionary planning of incremental software development. The method is initially evaluated using a sample project. The evaluation involves an investigation of the trade-off relationship between risk and the overall benefit. The link to empirical research is two-fold. Firstly, our model is based on interaction with industry and randomly generated data for effort and risk of requirements. The results achieved this way are the first step for a more comprehensive evaluation using real-world data. Secondly, we try to approach uncertainty of data by additional computational effort providing more insight into the problem solutions: (i) Effort estimates are considered to be stochastic variables following a given probability function; (ii) Instead of offering just one solution, the L-best (L>1) solutions are determined. This provides support in finding the most appropriate solution, reflecting implicit preferences and constraints of the actual decision-maker. Stability intervals are given to indicate the validity of solutions and to allow the problem parameters to be changed without adversely affecting the optimality of the solution.

Release Planning under Fuzzy Constraints

Release planning is extended to encompass fuzzy constraints and objectives. Two fundamental paradigms of problem solving under uncertainty are integrated: soft computing using fuzzy sets and intelligent decision support. The paradigm of soft computing integrates human and computational intelligence. It combines computational power with the human capability to store and process information that is imprecise, uncertain or may be only partially true. We have developed an approach improving existing methods for release planning by handling the uncertainty of data using fuzzy logic. Concretely, we use triangular fuzzy numbers to better represent the estimation of effort. The satisfaction of traditional constraints on effort is performed using a fuzzy system to obtain an overall satisfaction level of a solution. The complete approach is illustrated by a case study example.

Feature-Based Release Planning

There is a growing recognition that features act as an important organizing concept within the problem domain and as a communication mechanism between users and developers. Features are an essential abstraction that both customers and developers understand. They provide an efficient way to manage the complexity and size of requirements. The term "Feature Engineering" describes all the activities to promote features as "first-class objects" during the whole life-cycle. These activities include identifying features in requirements specifications, evaluating designs based on their ability to incorporate new and modified features, understanding the relationship between a software architecture and feature implementation mechanisms, uncovering feature constraints and interactions, and configuring systems based on desired feature sets.

The concept of a feature is applicable and important for any software development paradigm. However, it is especially important for any type of incremental product development. Features are the "selling units" provided to the customer. Incremental development has many advantages over the traditional waterfall approach. First, prioritization of features ensures that the most important features are delivered first. This implies that benefits of the new system are realized earlier. Consequently, less important features are left until later and so, if the time or budget is not sufficient, the least important features are the ones most likely to be omitted. Second, customers receive an early version of the system and so are more likely to support the system and to provide feedback on it. Third, the schedule and cost for each delivery stage are easier to estimate due to smaller system size. This facilitates project management and control. Fourth, user feedback can be obtained at each stage and plans can be adjusted accordingly. Fifth, an incremental approach is sensitive to changes or additions to features.

Determining Maximally Diversified Solutions for Constrained Multi-Objective Software Release Planning

We have invented an approach to support decision making by providing a portfolio of different solutions instead of proposing just one "best solution". Suggesting alternative solutions is important when both the model and the data in use are just a rough approximation of reality. Providing a qualified set of solutions being maximally diversified supports decision-making in case of incompleteness and uncertainty of information. This increases the understanding of the possible range of solutions and improves the chance of acceptance of a final solution by the decision-maker.

The problem under consideration is a generalized knapsack problem and is formulated and exactly solved using multi-objective integer programming. The model includes technological and resource constraints. Numerical results from a real-world project with more than 500 requirements and seven stakeholders involved are given. We have introduced a procedure to stepwise reduce the set of solution alternatives. In the first phase, we determine qualified solutions from solving a sequence of problems with changed parameters for the weighted objectives and the weighted increments. In the second phase, a set of maximally diversified solution is determined out of the qualified solutions.

Web-Based Decision Support for Software Release Planning

Web technology and web services represent a great opportunity for improving knowledge and experience exchange. The focus is on intelligent decision support for software release planning. We first characterize the problem of software release planning, and then review web technology and web-based decision support systems. By deriving some major requirements on web-based decision support for release planning, we then put forward a suggestion for an architectural design. We discuss the first steps of its realization and future directions of its real-world application.

Soft Requirements Negotiations

We have proposed a new and hybrid method called SRN (Soft Requirements Negotiator) to support decision-making during requirements negotiation. More precisely, given a set of requirements, we assist the DM in the process to gradually reduce, evaluate and prioritize the candidate sets of requirements. As a final result, we generate a set of most appropriate solutions. The proposed solutions are trade-offs between the supposed value (or priority) of those requirements, and the estimated effort to realize them. Our approach is soft in the sense that it only uses information in the degree and size as actually available at the different stages of the negotiation process. It was inspired from the paradigm of multi-criteria decision aid (MCDA), particularly the concordance/non-discordance principle.

Trade-off Analysis for Requirements Selection

The proposed method called Quantitative WinWin (Best paper award SEKE'2003) uses an evolutionary approach to address support for requirements negotiations. As a main result, Quantitative WinWin proposes a small number of alternatives for selecting possible sets of requirements from which the actual decision-maker finally can select the most appropriate one(s).

Management of Tabular-Based Requirements Using Rough Sets

Requirements management is an essential process to better understand, identify, derive, control and improve system requirements. Rough Set Analysis (RSA) is a promising technique of granular computing. We compare RSA with other known technique for handling data given in tabular form: Table inversion algorithms and Software Cost Reduction (SCR). One of the main strengths of RSA is its capability to handle inconsistency and to differentiate in terms of importance of the chosen attributes. These advantages are used to provide support during our new process of elicitation of requirements that are represented in tabular form. The emphasis of my research is on applying RSA for the process of requirements elicitation and on building new hybrid approach called MARTARS that improves existing methods for inconsistency management in Requirements Engineering. We use a modification of the well-known A-7E Aircraft Requirements Document to illustrate main concepts and ideas of our approach.

Strategic Planning for Enterprise Application Integration

Integration technology enables organizations to improve their business processes while retaining more value from their existing investments, thereby increasing their business agility and improving efficiency. We have proposed a novel and innovative approach for release planning as a means to perform strategic planning of enterprise application integration (EAI). It is intended to serve as an early guidance of how to conduct EAI which has to be supplemented later by more detailed operational planning. The release planning technology is based on an evolutionary approach for assigning features to releases. For varying degree of stakeholder importance, well balanced plans are generated satisfying most relevant technological, resource and risk constraints.

Selecting COTS Components for Product Lines

There is an increasing trend to build software systems from reusable components. Typical examples include systems that have product line architecture. Software product lines have a lot of advantages for both the developer and the customer, such as decreased developmental time and cost, and increased product quality. Using COTS components within product lines provides further improvement for the developmental time, cost and quality. Nevertheless, using COTS components involves several risks such as selecting a low-quality or otherwise improper component. We propose a systematic approach for performing the selection process based on the target system domain. The paper focuses on evaluating the quality of COTS products and provides techniques to deal with uncertainty as well as interactions between quality requirements.

Evaluation of COTS Products Using Knowledge Bases

Selection of commercial-off-the-shelf (COTS) software products is a knowledge-intensive process. In this paper, we show how knowledge bases can be used to facilitate the COTS selection process. We propose a conceptual model to support decision makers during the evaluation procedures. We then describe how this model is implemented using agent technologies supported by two knowledge bases (KB): the COTS KB and the methods KB. The model relies

on group decision-making and facilitates stakeholder negotiations during the selection process. It employs hybrid techniques, such as Bayesian belief networks and game theory, to address different challenges throughout the process. In addition, the paper also describes how the COTS knowledge base can be used at three levels of usage: global (over the internet), limited (between limited number of organizations) and local (within a single organization).

Software Engineering Decision Support and Empirical Investigations

Empirical software engineering is relying on or derived from observations or experiments. It is oriented towards making decisions about software engineering technologies. The paradigm of software engineering decision support (SEDS) goes beyond the concept of just reusing models, knowledge or experience. For a more focused problem domain, emphasis is on providing a methodology for pro-active generation, evaluation, prioritization and selection of solution alternatives. However, the results of this process can only be as good as the underlying models and experience. This is exactly the main purpose of empirical software engineering: to incrementally establish a body of empirically validated knowledge about existing or new phenomena.

We have studied the potential synergy between SEDS and empirical software engineering. The need for decision-making defines the objects and attributes of empirical investigations. This avoids esoteric experiments of high statistical significance that have no impact on reality. Vice versa, validated models and experience as concluded from sound empirical studies are enabler for making good decisions. This synergy is studied in more detail for decision support in software release planning.

Goal-Oriented Measurement plus System Dynamics – A Hybrid and Evolutionary Approach

Goal-oriented measurement following the GQM paradigm is focusing on a top-down approach for definition of appropriate metrics and a bottom-up approach to analyze and interpret results. The GQM approach leads to the specification and implementation of a measurement program for a particular set of issues and will form the basis for the interpretation of the measurement data in the context of the precisely defined goal.

Simulation in general has the ability to study complex systems in greater detail. The continuous simulation method SD is a very powerful approach with a broad range of applications in complex social, managerial, economic or engineering systems. As soon as an SD simulation model exists that reproduces current or past behavior of reality, systematic variation of model parameters (that is, sensitivity analysis or inclusion and exclusion of model structures) can help in understanding, controlling, and improving system behavior. SD modeling is supported by comprehensive tool support. The underlying paradigm of SD is systems thinking. The essential step toward systems thinking is to recognize the presence of feedback mechanisms in the observed system. In systems thinking, the behavior of a system is considered as primarily being generated by the interaction of all the feedback loops over time.

OBJECTIVES FOR NEXT YEAR

Intelligent Support for Release and Design Decisions of Evolvable Software-Intensive Systems

Hybrid Intelligent Systems (HIS) is a promising area of research due to the applicability of HIS to many complex and real world problems. HIS combine and integrate the strengths of multiple automated and semi-automated approaches to collect and analyze both qualitative and quantitative data. Such integration is necessary as neither human nor computational intelligence alone are able to provide sufficient release planning and software design decision support to achieve cost-effective evolvable systems. Humans are better able to handle soft and implicit objectives and constraints; computational intelligence is better able to handle a wide search through a large solution space. We will create and apply HIS in two innovative directions: (i) as the foundation of a methodology to support decision-making for release and design decisions; and (ii) as a new paradigm to generate, activate, and exploit knowledge for generating, evaluating, and explaining alternatives for release and design decisions.

We are working on a hybrid methodology for generating qualified solution alternatives for release and design problems under risk and resource constraints. The methodology will be based on the hybrid and customized use of modelling and simulation, genetic algorithms, and integer programming. It will overcome the limitations of current solution approaches in terms of its scope and its proactive decision support capability. No comparative results for this research question are currently available.

Dynamic Release Planning and Software Project Monitoring

Current planning is static in the sense that is does not consider any aspect of execution of the plans other than estimates of the resources required. However,

software development and evolution is a dynamic process with a large number of impacting factors. To better accommodate this dynamic character including feedback loops within the process, we will investigate software process simulation for modeling and executing the individual tasks related to the features to be assigned to releases. This will result in a better validity of the proposed plans. If done for the different types of resources involved, this will further allow project monitoring by comparing the planned tasks and their comparison with actual performance.

Open Scope Release Planning

Releases may also be arranged according to open scope release planning, where release times are not predefined. If this approach is used, the definitions of the release times and the requirements or features assigned to the respective releases are obtained as results. In this case, a solution may be sought that minimizes the time between releases, since the earlier a release is issued, the earlier it generates value, such as money. Another point to consider is whether there is any sub-, or otherwise related, products. In this case, each may have its own release planning goals, however, for the overall products the different cycles have to be synchronized. This can be modeled and solved using integer programming. The notion of "open scope planning" can be extended to address synchronization of releases as requested in any kind of embedded product development where you have to address planning of different parts (hardware, software, middleware). Each of these components is an open scope problem. However, for the final product, planning for all the components has to be synchronized because of the mutual dependency between the components. The solution is approached through a formal description of the problem that uses binary variables and applying genetic and integer programming optimization algorithms.

Fast Heuristics and Integer Programming Solution Algorithms

Current solution algorithms for release planning are based on genetic algorithms. These algorithms cannot guarantee optimality. We are working on fast (knapsack-type) heuristics able to generate good solutions for large-scale and complex problems in very short time. We will further analyze and fine-tune parameters for genetic algorithms and comparative analysis between heuristic, exact (integer programming) and evolutionary optimization algorithms. The results of that will be used to extend the functionality of Release Planner by providing features for handling uncertainty of data, fuzzy effort, and risk and dependency constraints. This includes the development of on-line user support to optimally customize the algorithms in dependence of the problem parameters.

Explanation Component for Release Planning Decision Support

Explanation is intended to increase acceptance of the tool, to improve understanding of the results, and to increase applicability and acceptance of suggested solutions. The release planner explanation scenario involves three types of agents (participants): the system that provides the solution to the problem. In our context a software agent; the user who obtains the solution for further treatment, in our context a human agent; and the explainer who explains the system's solution to the user, in our context a software agent.

We are mainly interested in the explainer agent. The explanation agent needs knowledge about the other agents. More precisely, the explainer needs an understanding of how the system obtains the solution, and a model of the user. A user model in general describes (i) What the user knows and (ii) What the user wants or needs to know. In some way, one can think of an explanation as an answer to (yet) unexpressed questions of the customer; therefore somehow the concept of a dialog enters the scenario. In particular, two types of explanations must be distinguished: a) One-step explanations, provided only once, and b) Dialog-type explanations that proceed in several steps. Both types contain a communication aspect, for a) it is degenerated.

Decision Support for Value-Based Software Technologies

A value-based approach to software engineering involves a different focus and approach to developing software. Meeting the needs of customers is as important as justifying the development efforts needed to meet those needs. In today's world with rapidly changing consumer demands, informational technology, and marketplaces, the requirements are changing rapidly requiring quicker adaptability by market participants. The critical success factor in this world for software developers is responding to changing requirements quickly while maintaining a focus on their value proposition which may be a quicker return on investments or an improvement in a public service like health, education, and defense.

Modeling and Implementation of Software Agents Decision Making

Software agents are knowledgeable, autonomous, situated and interactive software entities. Agents' interactions are of special importance when a group of agents interact with each other to solve a problem that is beyond the capability and knowledge of each individual. Efficiency, performance and overall quality of the multi-agent applications depend mainly on how the agents interact with each other effectively. In this chapter, we suggest an agent model by which we

can clearly distinguish different agent's interaction scenarios. The model has five attributes: goal, control, interface, identity and knowledge base. Using the model, we analyze and describe possible scenarios; devise the appropriate reasoning and decision making techniques for each scenario; and build a library of reasoning and decision making modules that can be used readily in the design and implementation of multi-agent systems.

RESEARCH TEAM MEMBERS AND CONTRIBUTIONS

Faculty Team Members

About ten researchers with a broad range of expertise represent the area of Software Engineering at University of Calgary. Emphasis is to integrate those ones having a close relationship to Software Engineering Decision Support. Simultaneously, iCORE support is used to attract further researchers from all over the world to join the group. Currently, most intensive collaborations are with:

NAME	ROLE/TOPIC
Dr Denzinger	Associate Professor, Department of Computer Science
Dr Eberlein	Associate Professor, Department of Electrical and Computer Enginee-ring
Dr Far	Associate Professor, Department Electrical and Computer Engineering
Dr Maurer	Full Professor, Department of Computer Science
Dr Walker	Assistant Professor, Department of Computer Science
Dr Wang	Full Professor, Department Electrical and Computer Engineering

Postdoctoral Fellows

NAME	ROLE/TOPIC
Dr An Ngo-The	Soft Computing in Software Engineering Decision Support
Dr Dietmar Pfahl (Fraunhofer Institute for Experimental Software Engineering)	Simulation-based Decision Support

PhD Students

NAME	ROLE/TOPIC
Jingzhou Li	Effort Prediction for Release Planning Using Collaborative Filtering
Abdallah Mohamed	COTS Software Product Selection
Michael Ochs (Fraunhofer IESE, Co-Supervisor)	Efficient and Effective Management of COTS Assessment and Selection
Omalde Saliu	iCORE International Graduate Student Scholarship Award
Tom Watanaya (Co-Supervisor)	Agent-based COTS Product Selection Method

MSc Students

NAME	ROLE/TOPIC
Gengsheng Du	Design and Analysis of an Explanation Component for ReleasePlanner (Department of Computer Science award)
Zhizhong Li	Management of Tabular-based Requirements Using Rough Sets
Sebastian Maurice	Decision Support for Value-Based Software Engineering Release Planning
Joseph Momoh	Impact Analysis of Release Planning Using ReleasePlanner
Wei Shen	Release Planning Under Fuzzy Effort Constraints
Yuhang Wang	Machine Learning for Improving Performance of Software Inspections (Department of Computer Science award and Alberta grad student scholarship from Alberta Learning)
Qun Zhou	Simulation Based Effort Estimation for COTS Based Software Development

Support and technical staff

NAME	ROLE/TOPIC
Amandeep	Research Associate
Kornelia Streb	Assistant

COLLABORATIONS

RESEARCH COLLABORATIONS
FRAUNHODER IESE AND FRAUNHOFER-CENTER MARYLAND
In accordance to the Academic Cooperation Research Exchange between the University of Calgary and the Fraunhofer Institute for Experimental Software Engineering ("Fh IESE"), the Laboratory for Software Engineering Decision Support and Fh IESE agree to a collaborative research and personnel exchange. On this basis, Dr Dietmar Pfahl has visited the laboratory for three months.
UNIVERSITY OF NEW SOUTH WALES
A similar agreement as signed with Fh IESE is in preparation to be signed with the research group of Dr Ross Jeffrey at University of New South Wales. Based on this agreement, the intention is to conduct joint research and exchange PhD students.
INFORMAL COLLABORATIONS
Informal collaborations were launched especially with the groups of Dr Lionel Briand (Carleton University, Canada), Dr Jens Jahnke (University of Victoria, Canada), Dr David Raffo (University of Portland, USA), and Dr Gerardo Canfora (University of Sannio, Italy).

INDUSTRY
DaimlerChrysler
A study on "Feature-Base Release Planning" was conducted for DaimlerChrysler AG, Research and Technology in Ulm (Germany). This study reflects the state-of-the art in software release planning. Therein, features are considered to be "a logical unit of behaviour that is specified by a set of functional and quality requirements." In other words, features are an abstraction from requirements that both customers and developers understand. Most of the topics discussed in the study are applicable to both the original requirements as well as to their aggregation into features.
COREL
Collaboration with Corel is devoted to Release Planning. Their main interest is to provide plans that fit to resource and budget constraints. No commercial product for that purpose is available on the market. In accordance to discussions conducted with Inno-Center Alberta and University Technologies Inc. (UTI), the strategy is to have Corel as a reference customer for a later product development. For access to business relevant real-world data, a Non-disclosure agreement was signed.
NORTEL NETWORKS AND TREMA TECHNOLOGIES
Two non-academic organizations have joined the proposed NSERC Strategic Project Grant: Trema Laboratories Inc. (Calgary) and Nortel Networks (CDMA Base-station Development, Calgary). The CDMA wireless group of Nortel Networks is presently supporting three releases in the application field, plus three releases in development and one release in the planning phase. The base transceiver station is primarily driven by software. Each release, the software team adds more functionality to its application code. Release planning and requirements negotiation is of crucial importance for business success. Trema Laboratories Inc. provides and integrates a full suite of internet-based global e-finance solutions allowing customers to maximize their visibility to liquidity, improve productivity, reduce costs, and increase shareholder value. Evolutionary development processes are an essential means to better react to changing markets and policies.
MULTIDISCIPLINE OR MULTI-INSTITUTIONAL PARTNERSHIPS
CSER
An NSERC CRD proposal titled "Simulation-Based Decision Support Software Quality Assurance" was conditionally approved. The project is part of the (Canadian) Consortium for Software Engineering Research. Created in 1996, CSER is a multi-party, industry-led research program, geared toward solving selected industrial problems in software engineering. The project called SimQuali aims to benefit the collaborators, their students and the Canadian economy in various ways. As a small company, Brycol Consulting cannot afford to support a research department. This project provides opportunity for Brycol to benefit from the collaborative research results embedded in an interaction/argumentation device when discussing trade-off in software quality improvements and outcomes. The intelligent decision support tool will provide the capability to evaluate the outcomes of feasibility alternatives based on standard variables for verification and validation techniques.
SECCO
The Software Engineering Consulting Consortium (SECCO) was formed with the objective of fostering and encouraging links between the industry and graduate students in the Software Engineering discipline engaged in research activities. SECCO is an organization operated by Software Engineering graduate students under the direction of the Software Engineering Decision Support (SEDS) lab (http://www.seng-decisionsupport.ucalgary.ca/). The objectives of SECCO are to be run not for profit, grow at a challenging and manageable rate, and to provide avenue for interaction between software engineering graduate students and the industry. The mission of SECCO is to provide industry access to cutting edge technologies in the area of SEDS and promote applied research and empirical validation of new technologies as part of the graduate education.
INTERNATIONAL SOFTWARE ENGINEERING RESEARCH NETWORK (ISERN)
The Software Engineering research group at the University of Calgary successfully applied to become a member of the International Software Engineering Research Network ISERN. This gives us excellent opportunities to further extend collaboration with leading researchers and research institutions all over the world. For a list of the 33 member organizations, see http://www.iese.fhg.de/network/ISERN/pub/isern.list_of_members.html.

INTELLECTUAL PROPERTY

ReleasePlanner™
While the release planner technology builds on a web-based approach, it has also been designed and developed to be easily customizable for the different types of users and different application scenarios. University Technology Inc. (UTI) has evaluated the technology for patent protection and determined that the objective function and consolidation process is both novel and patentable and has initiated the patent application process. UTI will be actively involved in managing this process. Due to the fact that this technology can service a broad horizontal market, the potential for securing multiple follow-up patents is also a positive opportunity. US and Canada patent protection is pending.

The research will also produce additional, valuable intellectual property such as the know-how generated through broad-ranging applications of the technology.

NSERC Project SimQuali
We have started to discuss general strategy and details of Intellectual Property. This is based on the existing IP policies of the involved parties (U of C, Brycol, and CSER).

FUNDING

In addition to the iCORE funding, Guenther Ruhe currently receives funding from NSERC (~$20K/year) and DaimlerChrysler (~$30K) to support his research. He is pursuing other sources of industry funding and participating in a potential commercial spinoff of his release planner software.

PUBLICATIONS

REFEREED JOURNAL PUBLICATIONS

G. Ruhe, A. Ngo-The: "Hybrid Intelligence in Software Release Planning", *International Journal of Hybrid Intelligent Systems*, Vol 1(2004), pp 99-110.

D. Greer, G. Ruhe: "Software Release Planning: An Evolutionary and Iterative Approach", *Information and Software Technology*, Vol 46 (2004), pp 243-253.

G. Ruhe, A. Eberlein, D. Pfahl: "Trade-off Analysis for Requirements Selection". *International Journal on Software Engineering and Knowledge Engineering*, Vol. 13 (2003), pp 345-366.

D. Pfahl, O. Laitenberger, Ruhe, J. Dorsch: "An Externally Replicated Experiment for Evaluating the Learning Effectiveness of Using Simulations in Software Project Management Education", *International Journal on Empirical Software Engineering*, Vol 8 (2003), pp 367-395.

D. Pfahl, G. Ruhe: IMMoS: "A Methodology for Integrated Measurement, Modeling, and Simulation", *International Journal of Software Process Improvement and Practice*, Wiley, Vol. 7 (2003), pp 189-210.

D. Pfahl, O. Laitenberger, G. Ruhe, J. Dorsch, T. Krivobokova: "Evaluating the Learning Effectiveness of Using Simulations in Software Project Management Education: Results from a Two Times Replicated Experiment", *Information and Software Technology*, Vol 46 (2004), pp 81-147.

CONFERENCES AND WORKSHOPS

Z. Li, G. Ruhe, "Management of Tabular-based Requirements Using Rough Sets," Proc. 4th ASERC Workshop on Quantitative and Soft Computing based Software Engineering. (QSSE2004), ASERC, Banff, Alberta, Feb. 2004, pp. 29- 34.

A. Mohamed, T. Watanaya, G. Ruhe, A. Eberlein, B. Far: "COTS Evaluation Supported by Knowledge Bases", International Workshop on Learning Software Organizations, Banff, June 21, 2004 (accepted).

G. Ruhe, G. Du: "Strategic Planning of Enterprise Application Integration". *Proceedings Banff Summit on Enterprise Application Integration,* Banff, 2004. pp 193-203.

Y. Wang, D. Liu, G. Ruhe: "Formal Description of the Cognitive Process of Decision Making", *Proceedings 3rd IEEE International Conference on Cognitive Informatics ICCI'04*, August 2004, pp 124-130.

An Ngo-The, G. Ruhe, W. Shen: "Release Planning under Fuzzy Effort Constraint", *Proceedings 3rd IEEE International Conference on Cognitive Informatics ICCI'04*, August 2004, pp 168-175.

S. Maurice, G. Ruhe, A. Mohamed, A. Amandeep, J. Momoh: "How to create a Win-Win Relationship between Industry and Software Engineering Graduate Education?", *Proceedings Canadian Conference on Computer and Software Engineering Education*, Calgary, March 2004.

A. Amandeep, G. Ruhe, M. Stanford: "Intelligent Support for Software Release Planning", 5th International Conference on Product Focused Software Process Improvement (PROFES'2004), April 5 - 8, Kansai Science City, Japan, LNCS Vol. 3009, pp 248-262.

G. Ruhe: "Software Engineering Decision Support and Empirical Investigations - A Proposed Marriage", *Proceedings Workshop on Empirical Studies in Software Engineering*, Rome 2003, 10 pages.

J. Li, G. Ruhe: "Web-Based Decision Support for Software Release Planning", *Proceedings of WI/IAT 2003 Workshop on Applications, Products and Services of Web-based Support Systems*, Halifax, 2003, pp 13-20.

An Ngo-The, G. Ruhe: Requirements Negotiation under Incompleteness and Uncertainty. *Proceedings SEKE'03*, June 2003, San Francisco Bay, pp 586-593.

G. Ruhe: "Intelligent Support for Selection of COTS Products," In: *Web, Web-Services, and Database Systems*, Lecture Notes in Computer Science, Vol 2593, Springer 2003, pp 34-45.

G. Ruhe: Software Engineering Decision Support – A New Paradigm for Learning Software Organizations. *Advances in Learning Software Organization. Lecture Notes In Computer Science* Vol. 2640, Springer 2003, pp. 104-115.

D. Pfahl, G. Ruhe: Goal-Oriented Measurement plus System Dynamics - A Hybrid and Evolutionary Approach. ProSim 2003, May 3-4, 2003 Portland, 9 pages.

D. Pfahl, G. Ruhe: IMMoS: A Methodology for Integrated Measurement, Modeling, and Simulation, *Proceedings of the ProSim'03 Workshop*, May 3-4, 2003, Portland.

BOOKS, BOOK CHAPTERS AND SPECIAL ISSUE EDITOR

A. Ngo-The, G. Ruhe: "Decision Support in Requirements Engineering", Chapter of the book: *Engineering and Managing Software Requirements* (Ed. by A. Aurum and C. Wohlin) (in preparation).

S. Maurice, G. Ruhe: "Decision Support for Value Based Software Engineering Release Planning", Chapter of the book: *Value-Based Software Engineering Management* (Ed. by S. Biffl, A. Aurum, P. Grÿnbacher, M. Halling) (in preparation).

D. Pfahl, G. Ruhe, K. Lebsanft, M. Stubberich: "Software Process Simulation with System Dynamics – A Tool for Learning and Decision Support", Chapter of the book: *Software Process Modeling* (Ed. By S.T. Acuna and N. Juristo), Kluwer (submitted).

D. Pfahl, G. Ruhe: "System Dynamics and Goal-Oriented Measurement: A Hybrid Approach". Accepted for: *Handbook of Software Engineering and Knowledge Engineering*, Vol. 3

G. Ruhe: "Software Release Planning". Accepted for: *Handbook of Software Engineering and Knowledge Engineering, Vol. 3*

G. Ruhe: Software Engineering Decision Support - Methodology and Applications. In: "Innovations in Decision Support Systems" (Ed. by Tonfoni and Jain). *International Series on Advanced Intelligence*, Volume 3, 2003, pp 143-174.

G. Ruhe (Guest Editor): Software Engineering Decision Support, *Journal of Software Engineering and Knowledge Engineering,* Vol. 13, No. 5 (Oct 2003),

PRESENTATIONS (SELECTION)

Goal-Oriented Measurement plus System Dynamics - A Hybrid an Evolutionary Approach, ProSim'03 Workshop, Portland, May 2003.

Simulation-Based Decision Support in Software Quality Management (SimQuali), CSER Proposal, Victoria, May 2003.

Approaching Uncertainty in Requirements Negotiations and Software Release Planning, University of Victoria Victoria, BC, June 2003.

Approaching Uncertainty in Requirements Negotiations and Software Release Planning, University of Berkeley, Berkeley, USA, July 2003.

Do We Really Need Software Engineering Decision Support? SEKE"03, San Francisco, USA, July 2003 (Panel discussion, Chair)

Requirements Negotiation under Incompleteness and Uncertainty, University of California, Berkeley, July 2003.

Software Engineering Decision Support and Empirical Investigations - A Proposed Marriage, ISESE 2003, Roma, Italy, September 2003.

Technology Transfer for Software Release Planning, CASCON 2003, Toronto, Canada, October 2003 (Panel discussion)

Experimentation and Simulation - How to make Synergy, ISERN Roma, Italy, October 2003.

Approaching Uncertainty in Software Engineering Decision Support, University of Sannio, Benevento, Italy, October 2003.

Release Planner Inc., Inno-Centre Alberta, Calgary, Canada, November 2003.

Intelligent Support for Software Release Planning, QSSD'2004, Banff, February 2004.