

Matthias Jarke, Peri Loucopoulos,
John Mylopoulos, Alistair Sutcliffe (editors):

**System Requirements:
Analysis, Management, and Exploitation**

Dagstuhl-Seminar-Report; 99
04.10.-07.10.94 (9440)

ISSN 0940-1121

Copyright © 1995 by IBFI GmbH, Schloss Dagstuhl, D-66687 Wadern, Germany
Tel.: +49 - 6871 - 2458, Fax: +49 - 6871 - 5942

Das Internationale Begegnungs- und Forschungszentrum für Informatik (IBFI) ist eine gemeinnützige GmbH. Sie veranstaltet regelmäßig wissenschaftliche Seminare, welche nach Antrag der Tagungsleiter und Begutachtung durch das wissenschaftliche Direktorium mit persönlich eingeladenen Gästen durchgeführt werden.

Verantwortlich für das Programm ist das Wissenschaftliche Direktorium:

Prof. Dr. Thomas Beth,
Prof. Dr. Oswald Drobnik,
Prof. Dr.-Ing. José Encarnação,
Prof. Dr. Hans Hagen,
Dr. Michael Laska,
Prof. Dr. Thomas Lengauer,
Prof. Dr. Christoph Meinel,
Prof. Dr. Wolfgang Thomas,
Prof. Dr. Reinhard Wilhelm (wissenschaftlicher Direktor)

Gesellschafter: Gesellschaft für Informatik e.V., Bonn,
Universität des Saarlandes,
Universität Frankfurt,
Universität Kaiserslautern,
Universität Karlsruhe,
Universität Stuttgart,
Universität Trier,
TH Darmstadt

Träger: Die Bundesländer Saarland und Rheinland-Pfalz

Bezugsadresse: Geschäftsstelle Schloss Dagstuhl
Universität des Saarlandes
Postfach 15 11 50
D-66041 Saarbrücken, Germany
Tel.: +49 -681 - 302 4396
Fax: +49 -681 - 302 4397
e-mail: office@dag.uni-sb.de
url: <http://www.dag.uni-sb.de>

System Requirements: Analysis, Management, and Exploitation

Organizers: Matthias Jarke, John Mylopoulos, Peri Loucopoulos, Alistair Sutcliffe

1. Introduction

Requirements engineering is a field of growing interest to informatics theory and practice. On the one hand, efforts in formalization and integration of formal and informal representations move the frontier of design support environments towards the upstream phases. Thus, computer science offers the *means* to deal with requirements information as well as with the downstream phases of software specification, design, and implementation.

On the other hand, users demand "corporate ownership" of information technology. They will not use systems they do not trust, and they will not trust systems they do not understand. Requirements modeling, which provides an application perspective on software-intensive systems, offers a way for this understanding. But ways must be found to actually use the expressed requirements (functional and non-functional ones) in the process of developing, deploying, and using information systems, even down to the task of estimating consequences of information technology to workplace and society. This places strong demands on the structure and content of requirements information and on its integration in development environments.

Research groups throughout the world are attacking these issues with very different approaches, ranging from Formal Methods to Enterprise Modeling to User Interface Technology to Participative Design to Socio-Technical and Ethnographic Methodologies. It was the intention of this Dagstuhl seminar to bring leading representatives of these approaches together in an intensive discussion setting to achieve real progress in understanding their interrelationships and possible roles in future practice. Practitioners with a proven understanding for research were invited to provide a real-world perspective; individual researchers, including junior ones, brought in their ideas and got feedback for their further research. The discussion was focused through a number of hypotheses concerning possible integration recently developed in collaborative basic research projects in which the organizers are involved.

This seminar brought together researchers from several disciplines and business practitioners to discuss needs and possibilities for better handling of requirements to complex organizational information systems. The discussion touched three dimensions of the requirements process:

- what requirements do we need to know about?
- what representations should be used?
- how can agreement between stakeholders be reached?

In addition, we discussed the business modeling context and three aspects of the RE process:

- modeling of methods and guidance
- modeling of CASE environments that guide, execute, and trace the process
- teaching of requirements engineering.

Participants prepared position papers which are available in the Technical Report Series at RWTH Aachen (Aachener Informatik Berichte 94-29).

Matthias Jarke

2. Final Programme

Tuesday, Oct. 4, 1994

	morning session chair: John Mylopoulos	
9:00 - 9:30	<i>Welcome</i>	
9:30 - 10:00	Matthias Jarke	Goals and Organization of the Seminar
10:30 - 12:00		Perspectives on RE
	Sol Greenspan	RE: An American Perspective
	Marcel Franckson	RE: An European Perspective
12:00 - 13:30	<i>Lunch Break</i>	
13:30 - 17:30		What is RE good for?
		Exploitation of RE Information
		Chair: Kalle Lyytinen
	Arne Sølvberg	RE: Old Wine in new Bottles?
	Markus Nuettgens	RE and Business Process Managment
	Bala Ramesh	Requirements Traceability
	Manfred Nagl	RE and Software Architecture

Wednesday, Oct. 5, 1994

Morning

Parallel Sessions (Part I - Collecting Issues, Positions, and Arguments):

Session 1: What should be in a Requirements Specification?

How can we get it there?

(Domain Models, Architectures, Standards, Scenarios)

Moderator: Alistair Sutcliffe

Session 2: When do we need which Representation for which kind of Information in RE?

How do we represent, analyze, transform and integrate Requirements Models?
(Language Concepts, Reasoning Techniques, Interface Issues)

Moderator: Eric Dubois

Session 3: What are the Cooperation Problems in Requirements Specifications?

What is available to help us solve them?

(Social Aspects, Models and Tools for Cooperation, Goal-Oriented Development)

Moderator: Colin Potts

Session 4: What is the Environment of RE?

How can we relate Requirements Models to the Environment?

(Composite Systems, Business Modeling and Engineering, Impact Analysis)

Moderator: Janis Bubenko

Afternoon

Excursion

Thursday, Oct. 6, 1994

Morning

Parallel Sessions (Part II - Organizing and Summarizing the Groups Results):

Topics and Moderators as before

Plenary: Reports from the parallel Sessions, general Discussion

Afternoon

Session 5: How do we model and manage the RE Process?
(Process Modeling, Traceability, Guidance, Improvement)
Moderator: Colette Rolland

Session 6: How and to whom should we teach RE?
(Undergraduate/Graduate Curriculum, Industrial Training, Technology Transfer)
Moderator: John Mylopoulos

Session 7: How do we organize Requirements Engineering Environments?
(Tools, Environments, Repositories)
Moderator: Panos Constantopoulos

Friday, Oct. 7, 1994

Plenary: General Discussions
Moderator: Peri Loucopoulos

3. Abstracts of Presentations

Configuration of Situational Methods and Tools to Facilitate Project-Specific Requirements Engineering

Sjaak Brinkkemper
University of Twente, Twente, The Netherlands

Proven fragments of existing requirements engineering techniques can be configured into a situational method i.e. a project approach that is tuned to the project at hand. We are currently developing a method engineering language, called MEL, that is capable of describing method fragment and of manipulating for the purpose of selection and assembly. Furthermore, a Computer Aided Method Engineering tool is under construction. This tool provides functionality for the configuration of method fragments that have been selected from the Method Base. Generator transform the fragment data into input data for a meta CASE tool. Empirical research is performed to collect current method tuning practices as well as suitable project characterization factors.

What is the Future of Requirements Engineering

Janis A. Bubenko
Swedish Institute of Systems Development, Kista, Sweden

I consider Requirements Engineering (RE) (for Information Systems (IS)) as the "early part" of the IS-area. As such RE deals with issues related to the actual business, which requires support of one or more inf. systems. Therefore, RE implies work in different application domains, and must consider many "situational factors". RE also implies very intensive co-working with domain experts and users. In RE, the importance of good knowledge of the application domain is essential, may it be insurance, health case, or car manufacturing. In RE, knowledge of formal, Computer Science (CS) related topics is not directly essential. It does not seem possible that IS and CS can in its domain accommodate knowledge about many different application domains, which is necessary to develop knowledge about how to apply RE to the domains. It is therefore, my conclusion that in 10-20 years, RE will be mainly performed by specialists from different "core domains". CS and IS will gradually move towards developing theoretical, genuine RE concepts and techniques, possible to specialize to a large number of domains.

Integrating Specifications

Panos Constantopoulos
Foundation for Research of Technology-Hellas, Heraklion, Greece

Requirements analysis usually results into a set of different specifications for the same system, which must be integrated. Integration involves elimination of discrepancies, completion and validation, and proceeds in stages of analysis and synthesis. Realizing that discrepancies between specifications may be due to differences in representation models and/or modeling perspectives and practices, we propose an approach to the analysis stage using meta-modeling and similarity analysis, whereby comparison of components is achieved through their classification under domain- and model-independent abstractions, and a newly developed model of similarity.

Albert at the Age of Two

Eric Dubois
University of Namur, Namur, Belgium

Our work is dealing with the modeling and the analysis of system requirements related to complex real-time distributed and cooperative systems. To this end, we had designed the Albert language, a formal language based on the concept of 'agent' in terms of which one may express real-time requirements as well as some "non functional" requirements related to the reliability and security aspects of agents. In the context of the application of this language is the content of C.I.N. applications, we are also investigating some methodological guidelines regarding the use of the Albert language.

Canonical Exceptions

Martin S. Feather
University of Southern California, Marina de Rey, USA

Software components are employed within many systems because of the flexibility that they provide. They can be tuned, extended, customized, etc., to the particulars of the application within which they are embedded. Yet, this flexibility comes at a price, primarily one of design rather than materials. A prevalent contributing factor to this design cost is the need to accommodate exceptional conditions, one of the areas of flexibility for which software is prized.

My focus is the disciplined design of exemption handling. My belief is, that it is during requirements engineering that exceptions and their treatments should be related to the idealized goals the system seeks to achieve. Compromises and approximations of these goals give rise to the need to tolerate exceptions, and the appropriate responses when those exceptions arise.

By canonical exceptions I mean commonly recurring forms of exceptions (and their handling) that arise in many systems, across many domains. In the realms of programming and specification the support of exception handling has given rise to general-purpose tools, language constructs and methodologies. The challenge is to extend these to the requirements phase of system design.

A Course on Requirements Engineering

Anthony Finkelstein
City University, London, UK

I only wrote a one line abstract - which will probably annoy my biographer...

"This short paper outlines the content of a course on requirements engineering."

Goals, Business Rules and IS Specification: A Ideological View of Requirements

Peri Loucopoulos
UMIST, Manchester UK

A critical factor in successful requirements analysis appears to be the understanding not only of 'what' the system should do, but also 'why'. To capture the reason of a system one needs mechanisms to describe the behavior of the organization in which the software system will be embedded. This approach suggest further understanding and modeling of organizational goals. Goals of a high level of abstraction represent objectives whose operationalization results in the organizational process. These goals will give rise to the systems requirements and constraints. In software systems development we often make the distinction between the enterprise world and the system world. The former describes the terrain about which the proposed system is to provide some service, while the latter is concerned with specifying on what the system is intending to do. Bridging the two worlds is a problem that requires a formal yet natural transition from the enterprise goals to the specification of system behavior. An enterprise model is needed which offers (among others) an ontology of goals. The formality engineered in this enterprise model would be further used for the elicitation of business rules, successfully externalizing business policy from software components.

Requirements Engineering as Knowledge Creation Towards Incremental Approach to Method Adoption and Use

Kalle Lyytinen
University of Jyväskylä, Jyväskylä, Finland

My work deals with knowledge theory and its application to RE process. It is used to extend and understand the role of RE in creating the knowledge of some organizational domain.

Requirements Engineering from the Viewpoint of Software Design

Manfred Nagl
RWTH Aachen, Aachen, Germany

In the last year we have studied integration of RE and Design languages, we have built integrated RE and Design tools, and we have realized integration tools from RE to Design. Further SDE tools are not of interest for this workshop. Tool building has been mechanized by assuming a framework and using generator tools, which derive tools from an internal specification.

Enough Requirements to Make Decisions

Marcel Franckson
SEMA Group, Montrouge, France

An important type of decision in the customer-supplier process is the investment decision. An investment decision is about whether to proceed or not in an IS development. It needs to investigate system characteristics to determine the feasibility, costs, benefits and risks of the IS development. An important issue is what kind of requirements and how much requirement do decision makers need for such a type of decision. This issue should be treated within the situational approach insofar as there are likely to be as many answers as there are different types of situations. This could be an interesting research focus for empirical research.

ACME and SOS

Sol Greenspan
GTE Labs, USA

Formalization of requirements can be done through conceptual modeling using representation and reasoning techniques. ACME, A Conceptual Modeling Environment, is an experimental environment that accepts one's favorite modeling framework (as a metamodel) and then provides editing and analysis facilities for it. A framework for Business Processing Re-Engineering (BPR) has been installed in ACME to produce ACME/BPR, which has been used on several large BPR projects.

Another line of investigation involves the design of a suitable ontology for combining business aspects of enterprise with the systems aspects. The Service-Orient Systems (SOS) framework supports modeling of business goals (in terms of the services offered by an enterprise organization, and resources (agents) and capabilities of systems. Requirements Engineering in this context is an analysis of the relationships (e.g. responsibility, support) between the various enterprise viewpoints, knowledge representation and reasoning techniques, i.e. those in ACME, are to be used to construct and use these models.

Describing the Usage of Information (Technology)

Peter Holm
Swedish Institute of Systems Development, Kista, Sweden

In the NATURE-project a framework has been proposed that gives an overview of different system contexts. This framework originates from the DAIDA project. One of the contexts is the usage of information (technology). In many current modeling methods there is an extremely poor terminology in describing different usage of information and information technology. Such a terminology could help us making software design models easier to understand and help developers to verbalize their expectations on organizational effects of implementing a particular software design suggestion. I hold the formulation of such a terminology to be a research topic of major importance for RE. My position paper to this seminar concerns a historical inheritance in our thinking about information. (techn.) that blinds us for the different usage of information (tech.). This idea is called "the database version of the descriptive fallacy", i.e. the assumption that information technology is used solely to describe a universe of discourse (a domain).

From Requirements to Implementation

Markus Nüttgens
Universität Saarbrücken, Saarbrücken, Germany

It was a pleasure to discuss topics along the border of organization and IT. RE seems to link the technology view to the social part of organizations. RE as a Socio-technical discipline? Are we looking for a common understanding of IS based processes? We are on the way!

Modeling Business Processes and Related Objects: An Integrated, High-level Petri Net Based Approach

Andreas Oberweis
Universität Karlsruhe, Karlsruhe, Germany

Requirements Engineering requires a model of relevant business processes and organizational aspects which are related to a system to be developed. For business process modeling high-level Petri Nets are suggested which allow an integrated representation of aspects like activity and data structure, organization concepts, exception handling. Petri Nets support a stepwise formalization of process models. An evolutionary simulation supported methodology is proposed which supports this modeling step. Interfaces are provided to map existing informal, application oriented notations for business processes and organizations modeling on to high-level Petri Nets.

Requirements: How to Organize them for Reuse

Barbara Pernici
Politecnico di Milano, Milano, Italy

Reusable requirements can be the basis for improving the requirements engineering practice, providing a starting point for the elicitation and analysis process and for restructuring existing applications. The design of reusable components and their reuse require that they are organized adequately to facilitate their storage and retrieval in a repository or library. The main points to provide such an organization are the following: to facilitate access to results of previous projects, to find "families" of related artifacts (requirements, conceptual schemes, and the like); to facilitate the comparison of parts of artifacts both for retrieval of components for reuse and also to build the components themselves from previous developments, providing similarity evaluation criteria and supporting links between related components; to organize the components at different levels of abstraction according to different types of criteria (e.g. refinement level, development phase, purpose of requirements). Clustering of requirements for creating abstraction levels and similarity criteria for supporting the analysis of the contents of a repository to create families of artifacts and support the creation of reusable components and of generic components are being studied at the Information System Group of Politecnico of Milano.

There is no more a Requirements Process than there is an Invention Process

Colin Potts
Georgia Institute of Technology, Atlanta, USA

Our studies of requirements activity in industrial software projects have revealed that many projects do not elicit requirements from a customer, but rather invent or negotiate product features as part of a broader conceptual design activity. The idea that this type of cooperative, intellectual work can be pre-planned or scripted like manufacturing processes or administrative workflows is as credible as the notion that our own research can be so scripted.

Design teams don't need process support that leads them by the nose down some path specified for them by 'experts' who do not practice. They need unobtrusive tools to support requirements work, tools to help them keep track of unresolved questions, unfulfilled obligations and consequences of assumptions. If this is process support, so be it; but the process implied is very like those of invention and discovery.

Requirements Traceability in Systems Development

Bala Ramesh
Naval Postgraduate School, Monterey, USA

Requirements traceability is concerned with following the life of a requirement identifying where requirements come from and linking requirements to various aspects produced during the system development process. Based on empirical studies involving various categories of stakeholders involved in large scale system development efforts, we have identified issues that must be addressed in implementing a comprehensive traceability scheme. Observations on current practices and lessons learnt from introducing traceability schemes are presented.

Agent-based Support for Communication between Developers and Users in Software Design

David F. Redmiles
University of California, Irvine, USA

Research in knowledge-based software engineering has led to advances in the ability to specify and automatically generate software. Advances in the support of upstream activities have focused on assisting software developers. We examine the possibility of extending computer-based support in the software development process to allow end users to participate, providing feedback directly to the developers. The approach uses the notion of "agents" developed in artificial intelligence research and concepts of participatory design. Namely, agents monitor end-users working with prototype systems and report mismatches between developers expectations and a system's actual usage. At the same time, the agents provide end users with an opportunity to communicate with developers, either synchronously or asynchronously. The use of agents is based on an actual software development experience. This research is carried out jointly with Andreas Lingensohn of Nynex Corporation, White Plains, and Frank Shipman of the University of Texas, College Station.

AI & Requirements Engineering

Kevin Ryan
University of Limerick, Limerick, Ireland

I see two orthogonal aspects to the interaction between AI and RE. The more obvious one is that we can look to AI technology to help solve many of the most difficult and crucial problems facing RE, particularly those of knowledge acquisition and storage. The second and more interesting aspect is that the achievements, limitations and experience of the AI field have lessons for the RE community. AI's limited success in solving many of its long standing problems should be a cautionary tale for RE researchers. Where AI has succeeded, and has been accepted by society, is within narrow niches where heuristic knowledge and sub-optimal solutions are integral to the domain. So RE should beware of the grand plans of early AI and concentrate instead on small scale, high leverage technology that can be readily evaluated and accepted within a social context.

RE: What is it?

Arne Sølvyberg
University of Trondheim, Trondheim, Norway

I am critical of the word Requirements Engineering on this field of such great importance. The word implies that the world should be viewed as what is external to the software, and that software exists in separation from the rest of the world (into which it is embedded). So let us find a new and better word on an existing research field.

Requirements Engineering through Application Frameworks Evolution

Paul Sorenson
University of Alberta, Canada

Much of my previous research in the RE area has dealt with the facilitation/development of a meta-system to support requirements and design environments. I am currently examining how to build an application framework for the domain of "size engineering". I plan to use this case study to determine aspects of process/product/tool evolution in the context of an application framework approach to RE.

Enactment of Process by Analysis of Traces

George Spanoudakis

Foundation for Research of Technology-Hellas, Heraklion, Greece

RE activities are highly creative and human oriented. Therefore, process models representing them tend to be general with regard to the alternative steps they predict at various stages of the underlying activities. Such models can not always provide detailed guidance to the humans who are responsible for the execution of the relevant activities. It seems that accumulation of experiences of executing the activities in different contexts of application domains is the only way to understand better their various alternatives.

We propose the tracing and analysis of past enactments of processes as a means of facilitating decisions. In particular reasoning by analogy on the basis of recorded traces could reveal similar situations to the one faced in some current enactment. In such cases detected analogies may be used as a criterion for following the scene enactment path and also provide concrete examples on how to operationalize the followed activity.

A Short History of Requirements Engineering

Alistair Sutcliffe

City University, London, UK

In the beginning there was the world, only one, and the designer was the Requirements Engineer. The designer had created the world and it only took 7 days with no budget overrun. Then the designer let two users interact with his wonderful creation, but they didn't understand their commitments and misused the system even though the constraints were formally specified. A little later another designer who was also a requirements engineer built a large floating artifact which was very successful for 40 days but it was seriously underspecified in its navigation functionality and ended up on top of a mountain. Time passed and there were very many more designers who were also requirements engineers. They decided to build a tower. Unfortunately they all had their own viewpoint expressed informally in a variety of notations and languages. Aha! thought the designers we need people of great foresight and understanding who communicate with us about the future which we can't know about. And so Requirements Engineers came to exist. And they were called priests and prophets who were sometimes greatly respected by the people and designers but only too often they got the future wrong and designers decided that collaboration for solving ambiguous wicked problems like the real world, should not be entrusted to such people. So designers became requirements engineers and developed complex multi functional systems which were the subject of occasional use in the real world. And so the trace was a history of RE which seemed to be going towards an unobtainable ideal. Alas, there was no one left to use the trace. So the moral of this tale is I came (to Dagstuhl), I thought, I wandered, and parted a little wiser.

Integrating Diagrammatic and Formal Requirements Descriptions

Martin Wirsing
Universität München, München, Germany

The goal of this work is to bridge part of the gap between semi-formal diagrammatic notations currently used in re-engineering methods and formal techniques. Algebraic/axiomatic specifications are known to be a suitable tool for the formal description of the functional aspects of a software system including data structure, control and reactive behavior. We have shown that they are also suitable for formally describing entity relationship diagrams, data flow diagrams, scenarios and access rights. Moreover, the notion of refinement between specifications provides a formal tool for checking the correctness of requirements and for tracing design decisions back to requirements. Hence formal specification techniques complement pre-formal and informal ones. The integration of all those leads to an improved requirements analysis.

Modeling "Why" in Requirements

Eric Yu and John Mylopoulos
University of Toronto, Toronto, Canada

Traditionally, requirements include a prescription for the system to be built as well as a model of the organizational environment within which it will function. The model of the organization is usually given in terms of entities, activities, agents and the like.

We argue that requirements ought to include information about organizational goals and intentions. To accomplish this, we propose a set of primitive concepts which make it possible to represent "dependencies" among organizational actors. the concepts are illustrated through an example.

Dagstuhl-Seminar 9440:

Sjaak Brinkkemper
University of Twente
Dept. of Computer Science
Postbus 217
NL-7500 AE Enschede
The Netherlands
sjbr@cs.utwente.nl
tel.: +31-53-89-3690

Janis Bubenko
SISU
Electrum 212
S-16440 Kista
Sweden
janis@sisu.se
tel.: +46-8-752-16 08

Panos Constantopoulos
Foundation for Research
and Technology-Hellas
Institute of Computer Science
Vassilika Vouton
GR-71110 Heraklion - Crete
Greece
panos@ics.forth.gr
tel.: +30-81-391-634

Eric Dubois
University of Namur
Facultés Universitaires de Namur
Institut d'Informatique
Rue Grandgagnage 21
B-5000 Namur
Belgium
edu@info.fundp.ac.be
tel.: +32-8172-4980

Martin S. Feather
University of Southern California
Information Sciences Institute
4676 Admiralty Way
Marina del Rey CA 90292-6695
USA
feather@isi.edu
tel.: +1-310-822-1511 / ext. 246

Anthony Finkelstein
Imperial College of Science
Department of Computing
180 Queen's Gate
London SW7 2BZ
Great Britain
acwf@doc.ic.ac.uk
tel.: +44-71-589-8261

Participants

Marcel Francksson
Sema Group SA
Direction Technique R&D
Barbes 16
F-92126 Montrouge
France
marcel@retnadt.fr
tel.: +33-1-4092-4300

Sol Greenspan
GTE Laboratories Incorporated
Computer and Intelligent Systems Lab.
40 Sylvan Road
Waltham MA 02254
USA
greenspan@gte.com
tel.: +1-617-466-29 62

Georges Grosz
Université Paris
CRI
17 Rue de Tolbiac
F-75013 Paris Cedex
France
grosz@masi.ibp.fr
tel.: +33-1-44-24-9365

Peter Holm
SISU - Kista
Swedish Inst. of System Development
Electrum 212
S-16440 Kista
Sweden
pholm@sisu.se
tel.: +46-8-752-1600

Matthias Jarke
RWTH Aachen
Informatik V
Ahornstr. 55
D-52072 Aachen
Germany
jarke@informatik.rwth-aachen.de
tel.: +49-241-8021-5 00

Pericles Loucopoulos
The University of Manchester
Inst. of Science and Technology
Dept. of Computation
P.O. Box 88
Manchester M60 1QD
Great Britain
pl@sna.co.umist.ac.uk
tel.: +44-61-200-3332

Kalle Lyytinen
University of Jyväskylä
Dept. of Computer Science
and Information Systems
P.O. Box 35
SF- 40351 Jyväskylä
Finland
kalle@iytko.iyu.fi
tel.: +358-41- 603-025

Neil Maiden
City University
Department Business Computing
School of Informatics
Northampton Square
London EC1V OHB
Great Britain
n.a.m.maiden@city.ac.uk
tel.: +44-71-477-8412

John Mylopoulos
University of Toronto
Dept. of Computer Science
6 King's College Road
Toronto Ontario M5S 1A4
Canada
jm@ai.toronto.edu
tel.: +1-416-978-5180

Markus Nüttgens
Universität des Saarlandes
Institut für Wirtschaftsinformatik
Geb. 14.1
Postfach 1511 50
D-66041 Saarbrücken
Germany
manue@iwi.uni-sb.de
tel.: +49-681-9762-228

Manfred Nagl
RWTH Aachen
Fachbereich Informatik III
Ahornstr. 55
D-52074 Aachen
Germany
nagl@rwthi3.informatik.rwth-aachen.de
tel.: +49-241-802-1300

Andreas Oberweis
Universität Karlsruhe
Institut Angewandte Informatik
und Formale Beschreibungsverfahren
Postfach 6980
D-76128 Karlsruhe
Germany
oberweis@aifb.uni-karlsruhe.de
tel.: +49-721-608-4283

Barbara Pernici
Politecnico di Milano
Dipart. di Elettronica e Informazione
Piazza Leonardo da Vinci 32
I-20133 Milano
Italy
pernici@elet.polimi.it
tel.: +39-2-2399 -35 26

Klaus Pohl
RWTH Aachen
Fachbereich Informatik V
Ahornstr. 55
D-52074 Aachen
Germany
pohl@informatik.rwth-aachen.de
tel.: +49-241-21513

Colin Potts
Georgia Institute of Technology
College of Computing
Atlanta GA 30332-0280
USA
potts@cc.gatech.edu
tel.: +1-404-894-5551

Balasubramaniam Ramesh
Naval Postgraduate School
Department of Systems Management
Code SM/RA
Monterey CA 93943
USA
ramesh@nps.navy.mil
tel.: +1-408-656-2439

David Redmiles
University of California
Department of Information and
Computer Science
Irvine CA 92717-3425
USA
redmiles@ics.uci.edu
tel.: +1-714-725-3823

Colette Rolland
Université Paris
1 UFR 06
17 Rue de la Sorbonne
75231 Paris Cedex
France
rolland@masi.ibp.fr
tel.: +33-1-40462785

Kevin Ryan
University of Limerick
College of Informatics and Electronics
Limerick
Ireland
ryan@ul.ie
tel.: +353-61-333-644

Arne Sölvberg
The Norwegian Institute of Technology
Faculty of Electrical Engineering
& Computer Science
N-7034 Trondheim
Norway
arne.solvberg@idt.unit.no
tel.: +47-7-359-3438

Paul G. Sorenson
Univeristy of Alberta
Computer Science Dept.
Edmonton Alberta T6G 2H1
Canada

George Spanoudakis
Foundation for Research and
Technology - Hellas
Science and Technology Park
P.O. Box 1385
GR-71110 Heraklion-Crete
Greece
gs@csi.forth.gr
tel.: +30-81-391-633

Alistair G. Sutcliffe
City University
Department Business Computing
School of Informatics
North Hampton Square
London EC1V OHB
Great Britain
a.g.sutcliffe@city.ac.uk
tel.: +44-071-477-8411

Martin Wirsing
Universität München
Institut für Informatik
Leopoldstraße 11B
D-80802 München
Germany
wirsing@informatik.uni-muenchen.de
tel.: +49-89-2180-6318 /-6317

Zuletzt erschienene und geplante Titel:

- R. Giegerich, J. Hughes (editors):
Functional Programming in the Real World, Dagstuhl-Seminar-Report; 89; 16.05.-20.05.94 (9420)
- H. Hagen, H. Müller, G.M. Nielson (editors):
Scientific Visualization , Dagstuhl-Seminar-Report; 90; 23.05.-27.05.94 (9421)
- T. Dietterich, W. Maass, H.U. Simon, M. Warmuth (editors):
Theory and Praxis of Machine Learning, Dagstuhl-Seminar-Report; 91; 27.06.-01.07.94 (9426)
- J. Encarnação, J. Foley, R.G. Herrtwich (editors):
Fundamentals and Perspectives of Multimedia Systems, Dagstuhl-Seminar-Report; 92; 04.07.-08.07.94 (9427)
- W. Hoepfner, H. Horacek, J. Moore (editors):
Principles of Natural Language Generation, Dagstuhl-Seminar-Report; 93; 25.07.-29.07.94 (9430)
- A. Lesgold, F. Schmalhofer (editors):
Expert- and Tutoring-Systems as Media for Embodying and Sharing Knowledge, Dagstuhl-Seminar-Report; 94; 01.08.-05.08.94 (9431)
- H.-D. Ehrich, G. Engels, J. Paredaens, P. Wegner (editors):
Fundamentals of Object-Oriented Languages, Systems, and Methods, Dagstuhl-Seminar-Report; 95; 22.08.-26.08.94 (9434)
- K. Birman, F. Cristian, F. Mattern, A. Schiper (editors):
Unifying Theory and Practice in Distributed Systems, Dagstuhl-Seminar-Report; 96; 05.09.-09.09.94 (9436)
- L. Bannon, R. Keil-Slawik, I. Wagner (editors):
Interdisciplinary Foundations of Systems Design and Evaluation, Dagstuhl-Seminar-Report; 97; 19.09.-23.09.94 (9438)
- M. Broy, L. Lamport (editors):
Specification and Refinement of Reactive Systems - A Case Study, Dagstuhl-Seminar-Report; 98; 26.09.-30.09.94 (9439)
- M. Jarke, P. Loucopoulos, J. Mylopoulos, A. Sutcliffe (editors):
System Requirements: Analysis, Management, and Exploitation, Dagstuhl-Seminar-Report; 99; 04.10.-07.10.94 (9440)
- J. Buchmann, H. Niederreiter, A.M. Odlyzko, H.G. Zimmer (editors):
Algorithms and Number Theory, Dagstuhl-Seminar-Report; 100; 10.10.-14.10.94 (9441)
- S. Heinrich, J. Traub, H. Wozniakowski (editors):
Algorithms and Complexity for Continuous Problems, Dagstuhl-Seminar-Report; 101; 17.10.-21.10.94 (9442)
- H. Bunke, T. Kanade, H. Noltemeier (editors):
Environment Modelling and Motion Planning for Autonomous Robots, Dagstuhl-Seminar-Report; 102; 24.10.-28.10.94 (9443)
- W. Maass, Ch. v.d. Malsburg, E. Sontag, I. Wegener (editors):
Neural Computing, Dagstuhl-Seminar-Report; 103; 07.11.-11.11.94 (9445)
- G. Berry, W.P. de Roever, A. Pogné, A. Pnueli (editors):
Synchronous Languages, Dagstuhl-Seminar-Report; 104; 28.11.-02.12.94 (9448)
- B. Becker, R. Bryant, O. Coudert, Ch. Meinel (editors):
Computer Aided Design and Test, Dagstuhl-Seminar-Report; 105; 13.02.-17.02.95 (9507)
- D. Garlan, F. Paulisch, W. Tichy (editors):
Software Architectures, Dagstuhl-Seminar-Report; 106; 20.02.-24.02.95 (9508)

- W.J. Cullyer, W.A. Halang, B. Krämer (editors):
High Integrity Programmable Electronic Systems, Dagstuhl-Seminar-Report; 107; 27.02.-03.03.95 (9509)
- J. Gruska, H. Umeo, R. Vollmar (editors):
Cellular Automata, Dagstuhl-Seminar-Report; 108; 06.03.-10.03.95 (9510)
- H. Alt, B. Chazelle, R. Seidel (editors):
Computational Geometry, Dagstuhl-Seminar-Report; 109; 13.03.-17.03.95 (9511)
- W. Bibel, K. Furukawa, M. Stickel (editors):
Deduction, Dagstuhl-Seminar-Report; 110; 20.03.-24.03.95 (9512)
- B. Freitag, C.B. Jones, Ch. Lengauer, H.-J. Scheck (editors):
Object-Orientation with Parallelism and Persistence, Dagstuhl-Seminar-Report; 111; 03.04.-07.04.95 (9514)
- J. Doran, N. Gilbert, U. Mueller, K. Troitzsch (editors):
Social Science Microsimulation: A Challenge for Computer Science, Dagstuhl-Seminar-Report; 112; 01.05.-05.05.95 (9518)
- R. Studer, M. Musen (editors):
Shareable and Reusable Problem Solving Methods, Dagstuhl-Seminar-Report; 113; 08.05.-12.05.95 (9519)
- J. Blazewicz, K. Ecker, L. Welch (editors):
Scheduling in Computer & Manufacturing Systems, Dagstuhl-Seminar-Report; 114; 15.05.-19.05.95 (9520)
- H. Beilner, G. Ciardo, C. Lindemann, K. Trivedi (editors):
Performance and Dependability Modeling with Stochastic Petri Nets, Dagstuhl-Seminar-Report; 115; 22.05.-26.05.95 (9521)
- M. Aigner, J. Spencer, E. Triesch (editors):
Computing with Faulty Inputs, Dagstuhl-Seminar-Report; 116; 29.05.-02.06.95 (9522)
- J.-R. Abrial, E. Börger, H. Langmaack (editors):
Methods for Semantics and Specification, Dagstuhl-Seminar-Report; 117; 05.06.-09.06.95 (9523)
- W. Effelsberg, D. Ferrari, O. Spaniol, A. Danthine (editors):
Architecture and Protocols for High Performance Networks, Dagstuhl-Seminar-Report; 118; 19.06.-23.06.95 (9525)
- Ph. Flajolet, R. Kemp, H. Prodinger, R. Sedgewick (editors):
'Average-Case'-Analysis of Algorithms, Dagstuhl-Seminar-Report; 119; 03.07.-07.07.95 (9527)
- D. Gustfield, T. Lengauer, C. Sander (editors):
Molecular Bioinformatics, Dagstuhl-Seminar-Report; 120; 10.07.-14.07.95 (9528)
- J. Chomicki, G. Saake, C. Sernadas (editors):
Role of Logics in Information Systems, Dagstuhl-Seminar-Report; 121; 17.07.-21.07.95 (9529)
- R.S. Boyer, A. Bundy, D. Kapur, Ch. Walther (editors):
Automation of Proof by Mathematical Induction, Dagstuhl-Seminar-Report; 122; 24.07.-28.07.95 (9530)
- P. Cousot, R. Cousot, A. Mycroft (editors):
Abstract Interpretation, Dagstuhl-Seminar-Report; 123; 28.08.-01.09.95 (9535)
- P. Brunet, D. Roller, J. Rossignac (editors):
CAD Tools for Products, Dagstuhl-Seminar-Report; 124; 04.09.-08.09.95 (9536)
- C. Dwork, E.W. Mayr, F. Meyer a.d. Heide (editors):
Parallel and Distributed Algorithms, Dagstuhl-Seminar-Report; 125; 11.09.-15.09.95 (9537)
- C. Hankin, H. R. Nielson (editors):
New Trends In the Integration of Paradigms, Dagstuhl-Seminar-Report; 126; 18.09.-22.09.95 (9538)