Clarence A. Ellis, Matthias Jarke (editors):

Distributed Cooperation in Integrated Information Systems

Dagstuhl-Seminar-Report; 38
5.-9.4.92 (9215)

ISSN 0940-1121

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.

Bezugsadresse: Geschäftsstelle Schloß Dagstuhl
Informatik, Bau 36
Universität des Saarlandes
W - 6600 Saarbrücken
Germany
Tel.: +49 -681 - 302 - 4396
Fax: +49 -681 - 302 - 4397
e-mail: office@dag.uni-sb.de

Third ICIS Workshop "Distributed Cooperation in Integrated Information Systems" April 5-9, 1992, Schloß Dagstuhl, Germany

Clarence A. (Skip) Ellis, University of Colorado, USA
Matthias Jarke, RWTH Aachen, Germany

Increasingly, information systems are being viewed as communication media among cooperating people rather than simply stores for large sets of data. The computer becomes a generalization of the telephone rather than of the pocket calculator. This may have a fundamental impact not only on the way how we design and implement information systems but also how we need to educate students.

The topic area of this workshop was highly interdisciplinary. It joined the expertise of cooperative work researchers and computer scientists to study specifically *the impact of cooperation on computer science and practice,* focusing on distributed information systems: What are the challenges to computer science resulting from viewing information systems as cooperation media? What are the new technical ideas suggested by the approach for solving difficult questions of information systems integration and evolution? Do we need to change the way how we educate computer scientists so that they can better address the hard system and application management questions of today and tomorrow?

"Intelligent and cooperative information systems" (ICIS) denotes a new generation of information systems intended as socio-technical systems to support human-machine cooperation. Dynamic change and exception handling are ubiquitous in such systems. Consensus-seeking through coordination as well as diversity through explicit conflict management should be supported. After an introduction into the technological, conceptual, and industrial background of computer-supported teamwork, specific working groups studied process and coordination models, group user interfaces, and repositories in their role as an organizational memory. Panel discussions explored problems of research agendas, technology transfer, and education. Participants found the interdisciplinary discussions strenuous but challenging.

The workshop continued the series of ICIS workshops organized by Mike Brodie and John Mylopoulos, begun with the Niagara workshop on development methods for ICIS (April 1991) and continued with the Como workshop on ICIS Core Technology (October 1991). The ICIS Workshop Series is sponsored by the ESPRIT program of the European Community, by the US National Science Foundation, and by the Canadian National Science and Engineering Research Council. Full proceedings with extended position papers are available from M. Jarke.

Thanks to the Schloß Dagstuhl team who, combined with perfect weather, provided a stimulating environment for our discussions.

Contents

DIFFERENTIAL VIEWER

Robert Balzer
USC Information Sciences Institute

The Differential Viewer visually highlights differences between versions of a text document by displaying colored areas where the versions differ. These colored areas contain either the replacement text (shown colored) or the replaced text (shown in grey) and can be toggled back and forth individually by mouse clicks between these two states.

Separate colors are associated with each version, thus allowing the differences among many versions to be displayed. Normally these versions are layered on top of one another and later changes obscure earlier ones in the same place, but the user can also declare versions to be mutually exclusive branches or optional features." Within the consistency constraints of these declarations the user can construct the particular configuration of versions to be viewed.

Moreover, the Differential Viewer allows these configurations to be interactively modified through the standard GNU-EMACS text editor, highlighting changes in colored areas as they are being made. The user interactively controls which version is being modified and can define new versions as needed. All of the version and configuration information is persistently retained with the text document.

CSCW APPLICATIONS
ARE TOO IMPORTANT TO BE LEFT TO THE USERS

Rudolf Bayer
Institut für Informatik, TU München

Computer scientists should no longer be preoccupied with their own systems and tools. Instead they should start focussing their attention onto applications, where their rare talents are badly needed for modeling and for creating the proper abstractions (like electronic circulation folders in offices) for significant classes of problems.

Some suggestions:

- finding suitable abstractions for human cooperation (e.g. CoActions) in general before building specific tools to support it.
- modeling the data and the programs (transactions) of an enterprise. Such a model provides the soil required for CSCW to grow on.

PETRI NETS AS MODELING TOOLS FOR DISTRIBUTED SYSTEMS

Wilfried Brauer
TU München.

CSCW could be seen as the new paradigm for informatics (see my paper in LNCS 555). As with other applications of informatics one may argue that the human component is so complex and unpredictable that formalization is not possible (or should not be good) which then leads to ad hoc developments of experimental soft- and hardware systems. But programs are also formalizations, and if they are not based on a sound conceptual, methodological and formal basis they are dangerous because they are not understandable.

There is probably not simply one single formalism which is useful but this does not mean that for each application one has to invent a new formalism. It is often much better to make a systematic extension or variation of an existing formalism which stays within a general methodological framework and is one that the computer based tools and the basic theory of the formalism can still be used or can be augmented. As a good candidate for such an extended and already rich and computer-supported formalism I would like to propose the Petri Nets - not the low level notions which are usually known from older textbooks, but high level nets (see e.g., the book on coloured Petri Nets by Kurt Jensen, soon appearing in the EATCS Monographs on Theoretical Computer Science, or the articles in the volumes of the subseries "Advances on Petri Nets" of LNCS) which combine the explicit representation of concurrency with algebraic specification mechanisms.

CSCW: A PROTOTYPICAL FUTURE INFORMATION SYSTEM CLASS ?

Michael L. Brodie
GTE Laboratories Incorporated, Waltham, MA, USA

The Dagstuhl workshop was the third in a series of three to investigate concepts for future information systems called Intelligent and Cooperative Information Systems (ICISs). Although some ICIS features are agreed upon (e.g., multi-agent/cooperative, distributed, intelligent), no one knows their exact nature. The Dagstuhl workshop focused on Computer Supported Cooperative Work (CSCW) as a potential example of an ICIS application class (e.g., CSCW agents corresponding to component ISs in an ICIS). A goal was to gain insight into CSCW features (e.g., forms of cooperation) for use in future information systems. Another goal was to understand CSCW requirements (e.g., forms of interoperability) for core technologies (e.g., object management, complex transactions).

CSCW was not clearly described in terms that could be translated into requirements for core technologies (e.g., CSCW primitives and means for composing primitives to describe CSCW entities and operations). From a systems point of view, there was broad agreement that CSCW require, alternately ICISs support: interoperability, heterogeneity, distribution, flexibility, autonomy (vs. global memory), process description, and evolution (to deal with constant change); and provide powerful information repository and user interface features. However, these requirements were described largely intuitively.

The participants expertise reflected the inherently multi-disciplinary nature of CSCW. There were difficulties communicating across some of the disciplines. For example, there was a socio-technical split. Some participants presented the compelling CSCW motivation of the desire to use computers to support organized human activity to meet important human or organizational needs. Discussions of this philosophical and socially-oriented type caused discomfort for those who were more comfortable discussing more precise concepts and developing analytical solutions. This socio-technical split corresponds to the solution- vs. technology-oriented approach to computer science. It is the responsibility of computer scientists to attempt to bridge this gap. Similar difficulties were encountered in the two previous ICIS workshops. At Niagara Falls, there was a split between the (information) systems and the non-information systems folks. At Como, there was a split between the systems and "intelligence/knowledge" folks. As opposed to the desired multi-way exchange between the participating disciplines (at each workshop), there appeared to be only one-way exchanges of ideas, problems, and requirements to the core technology folks. However, a very positive sign is that the participating disciplines saw the objectives as important enough not only to attend but also to work hard to bridge the terminological and cultural differences and present their ideas and hopes.

It was clear to all participants that there is a (r)evolution occurring in computer science in the problems it is addressing and the approaches taken. Contrary to the fear that "The party is over" for computer science (e.g., no new ideas, problems beyond our reach), the party has moved. It is no longer at our conventional computer science house. For example, computer science notions of programming in the small do not provide the advantages they once did. The vision of computing, and of CSCW, pose new challenges that are beyond our current concepts, tools, and techniques. Let's go find the party!

INFRASTRUCTURES

Janis Bubenko, jr.
SYSLAB, Sweden

It is time to revise and to clarify the concepts of "information systems". We should more openly advocate that information systems, and computing in general, is not one single, well bounded discipline, but rather an evolving network of related, but multi-paradigm disciplines, who all have one objective in common. This objective is to develop and to disseminate knowledge (principles, theories, methods, and practices) which deal with the development, use, and maintenance of large information and software systems which are active components of real-world, human oriented infrastructures. This area then naturally includes an evolving, mulitdimensional spectrum of research topics, some of which, for instance, are oriented towards the social sciences, some towards engineering, some towards business organization, and some towards the "core of computer science". This view seems to be shared by several participants in this workshop.

SHARED MENTAL MODELS OF COMPUTER SUPPORTED COLLABORATIVE WORK

Tung Bui
Naval Postgraduate School, Monterey

In order to design CSCW, it is crucial to understand the way people communicate among themselves, collaborate, cooperate in carrying out their work, make decisions, and, when necessary, resolve individual differences. It seems to me that, so far, we've been able to group only pieces of this understanding and a general, yet a comprehensive coordination theory or framework is desperately needed.

In my presentation, I propose a deductive approach to identify / analyze shared mental models, as a means to understand team work. A shared mental model is a representation of pertinent elements involved in team work and its interactions. Interpretations of the shared mental model could help define functional requirements of a CSCW.

COMPOSITION AND FACTORIZATION OF 'AGENTS'

A.B. Cremers
Universität Bonn, Germany

The development of the EPSILON distributed knowledge base management system (ESPRIT P530) offered the opportunity to study some aspects of 'intelligent' and cooperative information systems. An EPSILON knowledge base is a collection of theories and links between theories. Theories represent entities of the real world. As basic knowledge base units they can contain knowledge represented in different formalisms (logic programs, database tuples, ...). Each theory has an associated engine which implements the corresponding information structure and control mechanisms. Links specify relationships between theories (e.g. various types of open, closed, inheritance, dictionary referral), allowing combinations that go beyond the ability of most object-oriented programming systems. Using EPSILON one can separate general strategies from application knowledge: engine theories contain the definition of new cooperation strategies and corresponding types of links; objects theories contain the application - dependent knowledge. Explicitly represented in different theories, general strategies and application knowledge can both be reused in other contexts, either by applying the same strategies to other applications or by running the application under control or other strategies. We can view the different instances of the EPSILON kernel on different nodes and the user-defined engines as cooperating 'agents'. These engines administer theories which are connected by a variety of links. Engines do not only serve as interfaces to external systems, they also accomplish a variety of tasks essential to an interoperable system, e.g. extraction and verification of meta information, comparison of data from different sources, management of uncertain information, global constraints checking.

ON MODELING PROCESSES

Vasant Dhar
Stern School, New York University, USA

It is recognized that having access to the history of a protracted group problem solving activity can be very useful. Capturing this process enables one to query the database (history) in an flexible manner. Application areas include software development and almost any type of planning. My work is geared towards the development of a representation that allows history to be captured in a useful way.

CSCW AND COMPUTER SCIENCE

Clarence (Skip) Ellis
University of Colorado, USA

Our field of information systems must move from its traditioned emphasis upon "the computation" to an interdisciplinary emphasis on the communication, coordination and collaboration of active agents. This workshop is helping to forge concepts and thinking in the direction of computers as a tool for the interaction of people.

SUPPORTING INDIRECT, LONG-TERM COLLABORATIVE DESIGN WITH INTEGRATED, KNOWLEDGE-BASED DESIGN ENVIRONMENTS

Gerhard Fischer
University of Colorado, Boulder

We are developing a conceptual framework and a demonstration system for collaboration among members of design teams when direct communication among these members is impossible or impractical. Our research focuses on the long-term, indirect communication needs of project teams rather than the short-term needs occurring in face-to-face communication or electronic mail.

We address these needs with integrated, domain-oriented design environments. Our conceptual framework and our system building efforts address two major issues:

1) How does individual work blend into project work?
2) What role do the work objects play in such a coordination?

COORDINATION MECHANICS

Anatol Holt
University of Milano, Italy

Coordination Mechanics is a small new field of theory and science which has evolved in conjunction with the utilization of high technology for the support of distributed organized, multi-person activity. Its concern is organize human activity per se, and its focus is on the achievement of coordination between many interdependent actions / interactions / transactions. Even more specifically its concern is the mechanics of coordination. The mechanics depends on two organizations: the organization of an activity - into interdependent roles / responsibilities / tasks, and the organization of the physical artifactual environment which supports an activity - e.g. organization into rooms, places, connecting devices, such as corridors, streets, telephone wires, material, tools, machines, etc. Thus the focal concern of Coordination Mechanics unites consideration of the structure of a support system and the structure of the supported activities. For this, reason and requirements for coordination can serve as middle term allowing users and builders to understand each other - requirements for coordination rather then requirements for computation or requirements for communication. (Both computation and communication depend on coordination.) It is my hope that Coordination Mechanics will become a significant part of the theory and science pertinent to information machines of whatever sort and their employment.

REQUIREMENTS ENGINEERING: AN APPLICATION AREA FOR CSCW AND REPOSITORIES

Stephan Jacobs
RWTH Aachen, Germany

Requirements engineering is the most crucial part in software engineering. Different persons, e.g. customers, manager, specifiers, designers, end-users, etc. speaking different "languages" and looking from different views have to cooperate during the development of a software system. Additional development isn´t straight forward but consists of many loops, iterations, backtrackings, refinements, etc. Supporting the whole develpoment process is only possible, when cooperation of the persons as well as the process history with it´s dependencies, relationships, decisions, etc. is understood and modeled. CSCW and repository technology could help to solve the problem.

TECHNOLOGY AND COOPERATION: APPLICATION AND CHALLENGES

Matthias Jarke
RWTH Aachen, Germany

Information systems can be viewed as communication media among people. The computer becomes a generalization of the phone rather than of a calculating machine. This may have a fundamental impact on the way how we design and implement information systems, especially in design applications. The only thing we know about the next generation of information systems is that it, again, will just be the legacy for the generation after it. Cooperation support may play a major role in managing the change.

Cooperation plays its role at two levels of aggregation (at least): teamwork support and large-scale alliances. Technical challenges include: semantics preserving metamodels and techniques for efficiently generating interoperability code from them, cooperative processes for flexible coordination, and the changing balance between computer and communication costs. In our work, we are investigating the role of deductive object bases in repositories for areas such as software engineering, hypermedia co-authoring, and total quality management in industry.

TASK SHARING: A MODEL FOR DISTRIBUTED WORK MANAGAEMENT

Thomas Kreifelts
GMD Bonn, Germany

Distributed work management is concerned with situation in which people at different places collaborate to perform work over both short and long periods of time. As a conceptual basis for generic activity coordination facilities the ESPRIT project EuroCoOp has developed a model of activity coordination: the EuroCoOp Action Coordination Model (EACM). This model takes into account asynchronous activities and events, users roles, groups, and resources. It supports both prestructured and semi-structured activities, and the "ad-hoc" definition of activities. A notation is provided for describing collaborative activities and a mechanism is defined for supporting participants in those activities. In contrast to former approaches, activity descriptions are not used in a prescriptive way to control collaboration, but rather as a medium for the organization of distributed work by the users themselves.

The main support functions of the model and a according prototype system that has been developed include:

- offering facilities for specifying cooperative activities,
- offering a better and more consistent overview of complex actions,
- documenting and monitoring progress of cooperative work,
- allowing for dynamic changes during performance,
- allowing access to, and communication of, necessary background material (documents, notes, comments, etc.).

DESIGN INFORMATION SYSTEMS TO DESIGN INFORMATION SYSTEMS

Kalle Lyytinen
University of Jyväskylä, Finland

Most support technologies for information system design support solitary users. Moreover their design and implementation is seldom examined as one instance of a general information system design problem. In my talk I present and state some general requirements for design information systems. These involve modeling flexibility and support for "schema" evolution in design information systems, models for task execution, planning and cooperative environment. Another set of requirements involve multitask interfaces that help task management and communication. These requirements can be met only by developing more powerful modeling mechanisms and object-oriented implementation architectures.

CONVERSATIONS AND ROUTINES

Giorgio De Michelis,
Univ. of Milano, Italy

Within cooperative work we can distinguish many different basic elements: messages, actions, actors, documents, meetings, channels, etc. It is, anyhow, difficult to build a meaningful model of an adequate work process on the basis of the above mentioned basic atomic elements: No causal link can be established relating them within it. We need therefore more, less elementary, basic structured elements as conversations, procedures, processes, repositories, etc. . Each one of them is well defined and we can recognize its occurrence within a work process. But the interactions between, e.g., conversations and procedures, is still non-deterministic. The relation between one and the other of the above basic structured elements within a work process can be formalized in terms of "being mutually the context of one of the other ones". On the basis of the above findings, we have built a prototype of an office CSCW support system that integrates conventions and routines in a very natural way. The prototype is called Woorks (with UTUCS and COP) and was developed within the ESPRIT-2 Program (ITHACA Project).

SHARING PROCESSES: TEAM COORDINATION IN DESIGN REPOSITORIES

Carlos Maltzahn
Universität Passau, Germany

Information systems support for design environments emphasizes object management and tends to neglect the growing demand for team support. Process mangement is often tackled by rigid technological protocols which are likely to get in the way of group productivity and quality. Group tools must be introduced in an unobtrusive way which extends current practice yet provides structure and documentation of development experiences. The concept of sharing processes allows agents to coordinate the sharing of ideas, tasks, and results by interacting protocol automata which can be dynamically adapted to situational requirements. Inconsistency is managed with equal emphasis as consistency. The sharing process approach has been implemented in a system called ConceptTalk which has been experimentally integrated with design environments for information and hypertext systems.

High Speed Communication

Günter Müller
Universtät Freiburg, Germany

Computer, Kommunikation und Unterhaltungselektronik sind verschiedene Aspekte derselben Technologie. Das Zusammenwachsen ist kein technisches Problem an erster Stelle, sondern erfordert das Denken in globalen Zusammenhängen. Während Rechner hauptsächlich Daten verarbeiten, besteht die Realität bei Kommunikation und Unterhaltungselektronik aus dem Transport, der Darstellung und Verarbeitung isochroner Datentypen.

Die Infrastruktur für die Ermöglichung integrierter Dienste ist teuer und nur finanzierbar, wenn viele Anwendungen zusammen die Kosten aufbringen. Die Komplexität liegt in diesen Anwendungen und sollte nicht nur technologie getrieben, sondern "kundengetrieben" sein. Diese Infrastruktur muss "freie Kommunikation" für "alle" zulassen.

Information Repositories

John Mylopoulos
University of Toronto, Cananda

Many information systems applications (ranging from business to engineering design and process control) require the representation and management of large amounts of disparate information, ranging from real-time plant data to design decisions and office procedures. This information needs to be represented within a repository using a suitable model and needs to be managed to make sure it remains current and consistent. Moreover, repositories need to be tailored to support particular tasks, such as software development or maintenance, carried out by team of cooperating human agents.

My position paper describes some of the modeling requirements and the control functions that need to be supported by such repositories if they are to successfully support their intended applications.

Information Systems Quality and Quality Information Systems

Klaus Pohl
RWTH Aachen, Germany

The quality of information systems has not been a major impact of computer-aided software engineering so far. We characterized the requirements for quality-oriented CASE in information systems and presented a stepwise procedure how something like Total Quality Management can be achieved in software environments through process-oriented repository technology.

DISTRIBUTING AND INTEGRATING CHANGE: INSIGHTS FROM SOFTWARE INFORMATION SYSTEMS

Thomas Rose
University of Toronto, Canada

For the most part, software information systems are devoted to product management. The evolution of products in current systems is addressed primarily in terms of transactions and integrity control mechanisms. We discuss services that need to be provided by software information systems to support software evolution, including explicit representing of change, change coordination and conflict management. We also present experiences with CAD^{o}, a prototype system offering coordination tools which interoperate with repository systems through CAD^{o}'s meta model and captures product management and group support activities in a coherent framework.

COMMUNICATION AND COOPERATION IN DISTRIBUTED SYSTEMS

Josef Schaefer
GMD Bonn, Germany

The contents of the POLIKOM - program shall bring a quantum leap in the quality of the support of human collaboration. Using the example and the challenge of the distributed function of the German government in Bonn and Berlin the new approaches and techniques will be used in real life and tested. A success in this application example will serve as a good sign for the building up of a new Europe of regions.

COMPUTER-MEDIATED COOPERATION THROUGH SHARED KNOWLEDGE

Peter G. Selfridge
AT&T Bell Laboratories Murray Hill, NJ 07974, USA

Cooperation among individuals is usually only a problem in a large project; this in turn make cooperation a distributed problem as well. One central problem underlying effective cooperation is sharing organizational knowledge. Computer-mediated support of distributed cooperation can be achieved by providing individuals access to a shared knowledge base of information relevant to their jobs. The organization and access of such a knowledge base must include:

1. easy access to relevant information;
2. pointers to owners or sources of information and
3. the ability of the knowledge base to evolve as knowledge changes.

We have designed, implemented and deployed such a knowledge base for software design within a large-scale software development project.

MACHINE SUPPORT OF COOPERATION AMONG HUMANS

Donald Steiner
Siemens AG, Germany

In order to truly support cooperation among humans and to actively participate themselves in the cooperative process, machines must have a model of the cooperative process. This requires not only a simulation of the various "communicative" events happening, but also a detailed model of the work process (goal acquisition, determination of tasks and schedules to accomplish the goal, etc.). Based upon this model, along with an "understanding" of the semantics of speech act theory (propose, accept, order, etc.) we claim that computers will be able to intelligently interact with humans about the components of the work process, thus establishing the first steps toward Human-Computer Cooperative Work (HCCW).

Clearly, this will at first be only achievable on a small scale, as many more AI techniques (in particular planning, learning and explanation) are needed to fully integrate computer systems into the community. (right, so this may never happen - it at least serves as a nice goal.)

What, however, truly can happen, is that, by using these techniques a lot of the tredious cooperative tasks can be moved from the domain of the human to that of the machine, thus freeing humans and allow them to make more use of their own intelligence.

POLIKOM

Dennis Tsichritzis
GMD Bonn, Germany

You do research to learn,

if you succeed it is great,
if you fail you have the satisfaction of trying and
you also learn what to do and not to do for the next time.

SOFTWARE REPOSITORIES: NEED TO IDENTIFY SPECIFIC ROLES

Yannis Vassiliou
University of Crete, Greece

In almost all environments, either found in the literature or announced by companies, for Information Systems Development and evolution, repositories pop out as (more often) a cooperation medium between the developers through stored artifacts. It is noticed though, that repositories are at times referred casually, carefully defined and described in fuzzy terms. Everyone seems to know how important they are, how many good things they are supposed to contain, how many hard problems can be solved / divided by just assuming that the right information can be found in the repository.

Yet, seldom one sees a clear statement of the repository's task and, most importantly, how this task influences its structure, contents and organization.

Our experience in building such a repository, limited in the domain of Software Development with the ultimate goal: reusability of software artifacts (objects) revealed that even constraining and identifying the task, several difficult system and conceptual issues come up.

We have found that a software repository brings in several disciplines: databases, knowledge bases, hypermedia, and information retrieval. Extending the well-known notion of the database schema, though a multiple instantiation hierarchy and modeling description capability was found necessary for repositories for large-grain artifact reusability (specifications, designs, code, etc.). Mechanisms for inferencing and reasoning within the repository, as well as the ability to locate artifacts with appropriate methods contribute greatly to tackle the selection problem. Finally the hypermedia paradigm for browsing and filtering make interaction with the repository easier.

KNOWLEDGE-BASED GENERATION OF COOPERATIVE MULTIMEDIA PRESENTATIONS

Wolfgang Wahlster
Universität Saarbrücken & German Research Center for AI (DFKI), Germany

Multimodal presentation systems combining natural language and graphics take advantage of both the individual strength of each communication mode and the fact that both modes can be employed in parallel. It is an important goal of our research not simply to merge the verbalization results of a natural language generator and the visualization results of a knowledge-based graphics generator, but to carefully coordinate natural language and graphics in much a way that they generate a multiplicative improvement in communication capabilities. Allowing all of the modalities to refer and depend upon each other is a key to the richness of multimodal communication. In the WIP system that plans and coordinates multimodal presentations in which all material is generated by the system, we have integrated multiple AI components such as planning, knowledge representation, natural language generation, and graphics generation.

In a distributed setting WIP can generate a possibly infinite variety of presentations of some information stored in a central knowledge base. WIP can tailor these presentations to the individual members of a group of users during real-time collaboration.

The current prototype of WIP generates multimodal explanations and instructions for assembling, using, maintaining and repairing physical devices.

Prof. Robert Balzer
Information Sciences Institute
University of Southern California
Martina del Rey, Ca 90292,USA
balzer@vaxa.isi.edu
balzer@a.isi.edu

Prof. Dr. Rudolf Bayer
Institut für Informatik
TU München
8000 München 2
bayer@informatik.tu-muenchen.de

Prof. Dr. Wilfried Brauer
Institut für Informatik II
TU München
8000 München 2
brauer@informatik.tu-muenchen.de

Dr. Michael L. Brodie
Database Systems Lab
GTE Laboratories Inc.
40 Sylvan Rd,
Waltham, Mass. 02254, USA
Fax: (617)290-9320
brodie@gte.com

Prof. Janis Bubenko
SYSLAB
P.O. Box 1250
S-16428 Kista
SCHWEDEN
janis@sisu.se

Prof. X. Tung Bui
Dept. Administrative Sciences
Naval Postgraduate School
Monterey, Ca 93943, USA
3867p@navpgs.bitnet

Prof. Dr. Armin B. Cremers
Universität Bonn
Römerstraße 164
5300 Bonn 1
abc@niob.informatik.uni-bonn.de

Prof. Vasant Dhar
Dept. Information Systems
Stern School, New York University
90 Trinity Place
New York, N.Y. 10006, USA
vdhar@is-5.stern.nyu.edu

Prof. Clarence A. Ellis
Dept. Computer Science
University of Colorado
Boulder, Co, 80309-0430 USA
skip@cs.colorado.edu

Gerhard Fischer
Department of Computer Science and
Institute of Cognitice Science
Campus Box 430,
University of Colorado
Boulder , CO 80309-0430 USA
gerhard@cs.colorado.edu

Dr. Anatol Holt
Dept. Computer Science
University of Milano
20133 Milano
ITALIEN

Prof. Dr. Matthias Jarke
Lehrstuhl Informatik V
RWTH Aachen
Ahornstr. 55, 5100 Aachen
jarke@informatik.rwth-aachen.de

Dr. Thomas Kreifelts
Inst. Angewandte Informationstechnik
GMD, Schloss Birlinghoven
5205 St. Augustin
tom@f3svb.gmd.de

Prof. Kalle Lyytinen
Dept. Computer Science
University of Jyväskylä
Seminaarinkatu 15
SF 40-100 Jyvaskyla
FINNLAND
kalle@jytko.jyu.fi

Giorgio De Michelis
Dipartimento di Scienze dell'Informazion
Universita' degli Studi di Milano
20133 Milano
ITALIEN
gdemich@hermes.mc.dsi.unimi.it

Prof. Dr. Günter Müller
Institut für Informatik und Gesellschaft
Universität Freiburg
Albert-Ludwigs-Universität
Werthmannplatz, 7800 Freiburg
mue@ibm.ruf.uni-freiburg.de

Prof. John Mylopoulos
Dept. Computer Science
University of Toronto
6 King's College Rd.
Toronto, Ont M5S 1A4
CANADA
jm@ai.toronto.edu

Dr. Thomas Rose
Dept. Computer Science
University of Toronto
6 King's College Rd.
Toronto, Ont M5S 1A4
CANADA
rose@cs.toronto.edu

Josef Schäfer
GMD
Schloss Birlinghoven
5205 St. Augustin

Dr. Peter G. Selfridge
AT&T Bell Laboratories
Room 2B-425
Murray Hill, NJ 07974
pgs@research.att.com

Dr. Donald Steiner
DFKI
Postfach 2080
6750 Kaiserslautern
steiner@dfki.uni-kl.de

Prof. Dr. Dennis Tsichritzis
GMD - Vorstand
Schloss Birlinghoven
5205 St. Augustin
dt@cui.unige.ch

Prof. Yannis Vassiliou
FORTH-CSI
36 Dedalou St.
P.O. Box 1385
Heraklio, Crete
GRIECHENLAND
yannis@csi.forth.gr

Prof. Dr. Wolfgang Wahlster
Fachbereich 14 - Informatik
Universität des Saarlandes
6600 Saarbrücken 11
wahlster@dfki.uni-sb.de

DISCUSSION RECORDERS

Stephan Jacobs
Informatik V
RWTH Aachen
Ahornstr. 55, 5100 Aachen
jacobs@informatik.rwth-aachen.de

Carlos Maltzahn
Lehrstuhl für dialogorientierte Systeme
Fachbereich Informatik
Universität Passau
maltzahn@andorfer.uni-passau.de

Klaus Pohl
Informatik V
RWTH Aachen
Ahornstr. 55, 5100 Aachen
pohl@informatik.rwth-aachen.de

Zuletzt erschienene und geplante Titel:

J. Berstel , J.E. Pin, W. Thomas (editors):
Automata Theory and Applications in Logic and Complexity, Dagstuhl-Seminar-Report; 5, 14.-18.1.1991 (9103)

B. Becker, Ch. Meinel (editors):
Entwerfen, Prüfen, Testen, Dagstuhl-Seminar-Report; 6, 18.-22.2.1991 (9108)

J. P. Finance, S. Jähnichen, J. Loeckx, M. Wirsing (editors):
Logical Theory for Program Construction, Dagstuhl-Seminar-Report; 7, 25.2.-1.3.1991 (9109)

E. W. Mayr, F. Meyer auf der Heide (editors):
Parallel and Distributed Algorithms, Dagstuhl-Seminar-Report; 8, 4.-8.3.1991 (9110)

M. Broy, P. Deussen, E.-R. Olderog, W.P. de Roever (editors):
Concurrent Systems: Semantics, Specification, and Synthesis, Dagstuhl-Seminar-Report; 9, 11.-15.3.1991 (9111)

K. Apt, K. Indermark, M. Rodriguez-Artalejo (editors):
Integration of Functional and Logic Programming, Dagstuhl-Seminar-Report; 10, 18.-22.3.1991 (9112)

E. Novak, J. Traub, H. Wozniakowski (editors):
Algorithms and Complexity for Continuous Problems, Dagstuhl-Seminar-Report; 11, 15-19.4.1991 (9116)

B. Nebel, C. Peltason, K. v. Luck (editors):
Terminological Logics, Dagstuhl-Seminar-Report; 12, 6.5.-18.5.1991 (9119)

R. Giegerich, S. Graham (editors):
Code Generation - Concepts, Tools, Techniques, Dagstuhl-Seminar-Report; 13, 20.-24.5.1991 (9121)

M. Karpinski, M. Luby, U. Vazirani (editors):
Randomized Algorithms, Dagstuhl-Seminar-Report; 14, 10.-14.6.1991 (9124)

J. Ch. Freytag, D. Maier, G. Vossen (editors):
Query Processing in Object-Oriented, Complex-Object and Nested Relation Databases, Dagstuhl-Seminar-Report; 15, 17.-21.6.1991 (9125)

M. Droste, Y. Gurevich (editors):
Semantics of Programming Languages and Model Theory, Dagstuhl-Seminar-Report; 16, 24.-28.6.1991 (9126)

G. Farin, H. Hagen, H. Noltemeier (editors):
Geometric Modelling, Dagstuhl-Seminar-Report; 17, 1.-5.7.1991 (9127)

A. Karshmer, J. Nehmer (editors):
Operating Systems of the 90s and Beyond, Dagstuhl-Seminar-Report; 18, 8.-12.7.1991 (9128)

H. Hagen, H. Müller, G.M. Nielson (editors):
Scientific Visualization, Dagstuhl-Seminar-Report; 19, 26.8.-30.8.91 (9135)

T. Lengauer, R. Möhring, B. Preas (editors):
Theory and Practice of Physical Design of VLSI Systems, Dagstuhl-Seminar-Report; 20, 2.9.-6.9.91 (9136)

F. Bancilhon, P. Lockemann, D. Tsichritzis (editors):
Directions of Future Database Research, Dagstuhl-Seminar-Report; 21, 9.9.-12.9.91 (9137)

H. Alt , B. Chazelle, E. Welzl (editors):
Computational Geometry, Dagstuhl-Seminar-Report; 22, 07.10.-11.10.91 (9141)

F.J. Brandenburg , J. Berstel, D. Wotschke (editors):
Trends and Applications in Formal Language Theory, Dagstuhl-Seminar-Report; 23, 14.10.-18.10.91 (9142)

H. Comon , H. Ganzinger, C. Kirchner, H. Kirchner, J.-L. Lassez , G. Smolka (editors):
Theorem Proving and Logic Programming with Constraints, Dagstuhl-Seminar-Report; 24, 21.10.-25.10.91 (9143)

H. Noltemeier, T. Ottmann, D. Wood (editors):
Data Structures, Dagstuhl-Seminar-Report; 25, 4.11.-8.11.91 (9145)

A. Dress, M. Karpinski, M. Singer(editors):
Efficient Interpolation Algorithms, Dagstuhl-Seminar-Report; 26, 2.-6.12.91 (9149)

B. Buchberger, J. Davenport, F. Schwarz (editors):
Algorithms of Computeralgebra, Dagstuhl-Seminar-Report; 27, 16.-20.12.91 (9151)

K. Compton, J.E. Pin , W. Thomas (editors):
Automata Theory: Infinite Computations, Dagstuhl-Seminar-Report; 28, 6.-10.1.92 (9202)

H. Langmaack, E. Neuhold, M. Paul (editors):
Software Construction - Foundation and Application, Dagstuhl-Seminar-Report; 29, 13..-17.1.92 (9203)

K. Ambos-Spies, S. Homer, U. Schöning (editors):
Structure and Complexity Theory, Dagstuhl-Seminar-Report; 30, 3.-7.02.92 (9206)

B. Booß, W. Coy, J.-M. Pflüger (editors):
Limits of Modelling with Programmed Machines, Dagstuhl-Seminar-Report; 31, 10.-14.2.92 (9207)

K. Compton, J.E. Pin , W. Thomas (editors):
Automata Theory: Infinite Computations, Dagstuhl-Seminar-Report; 28, 6.-10.1.92 (9202)

H. Langmaack, E. Neuhold, M. Paul (editors):
Software Construction - Foundation and Application, Dagstuhl-Seminar-Report; 29, 13.-17.1.92 (9203)

K. Ambos-Spies, S. Homer, U. Schöning (editors):
Structure and Complexity Theory, Dagstuhl-Seminar-Report; 30, 3.-7.2.92 (9206)

B. Booß, W. Coy, J.-M. Pflüger (editors):
Limits of Information-technological Models, Dagstuhl-Seminar-Report; 31, 10.-14.2.92 (9207)

N. Habermann, W.F. Tichy (editors):
Future Directions in Software Engineering, Dagstuhl-Seminar-Report; 32; 17.2.-21.2.92 (9208)

R. Cole, E.W. Mayr, F. Meyer auf der Heide (editors):
Parallel and Distributed Algorithms; Dagstuhl-Seminar-Report; 33; 2.3.-6.3.92 (9210)

P. Klint, T. Reps, G. Snelting (editors):
Programming Environments; Dagstuhl-Seminar-Report; 34; 9.3.-13.3.92 (9211)

H.-D. Ehrich, J.A. Goguen, A. Sernadas (editors):
Foundations of Information Systems Specification and Design; Dagstuhl-Seminar-Report; 35; 16.3.-19.3.9 (9212)

W. Damm, Ch. Hankin, J. Hughes (editors):
Functional Languages:
Compiler Technology and Parallelism; Dagstuhl-Seminar-Report; 36; 23.3.-27.3.92 (9213)

Th. Beth, W. Diffie, G.J. Simmons (editors):
System Security; Dagstuhl-Seminar-Report; 37; 30.3.-3.4.92 (9214)

C.A. Ellis, M. Jarke (editors):
Distributed Cooperation in Integrated Information Systems; Dagstuhl-Seminar-Report; 38; 5.4.-9.4.92 (9215)