Francois Bancilhon, Peter Lockemann,
Dionysis Tsichritzis (editors):

Directions of Future Database Research

Dagstuhl-Seminar-Report; 21
9.9.-12.9.91 (9137)

ISSN 0940-1121

Tel.: +49-6871 - 2458
Fax: +49-6871 - 5942

Das Internationales 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:
Prof. Dr.-Ing. José Encarnaçao,
Prof. Dr. Winfried Görke,
Prof. Dr. Theo Härder,
Dr. Michael Laska,
Prof. Dr. Thomas Lengauer,
Prof. Ph. D. Walter Tichy,
Prof. Dr. Reinhard Wilhelm (wissenschaftlicher Direktor).

Gesellschafter: Universität des Saarlandes,
Universität Kaiserslautern,
Universität Karlsruhe,
Gesellschaft für Informatik e.V., Bonn

Träger: Die Bundesländer Saarland und Rheinland Pfalz.

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

DAGSTUHL SEMINAR

DIRECTIONS OF FUTURE DATABASE RESEARCH: FACING THE IMPACT OF EMERGING THEORIES, TECHNOLOGIES AND APPLICATIONS

Organized by:

Dionysis Tsichritzis (Université de Genève)
Francois Bancilhon (O2 Technologies, Paris)
Peter Lockemann (Universität Karlsruhe)

September 9-12, 1991

Summary

Database technology is a field with a well-established tradition, that has matured over the past 20 years and achieved simplicity and rigidity of its basic concepts, well-founded and well-rounded theories on the static aspects of data structuring and data distribution and on the dynamic aspects of transaction management, and solid technical solutions that transform the theoretical results to well-performing algorithms and widely available and employed database systems.

The success of database systems seems to breed among many database researchers a kind of identity crisis. On the one hand, some feel that all major results have been achieved so that there are few avenues left towards widely acclaimed theoretical, methodical or technical breakthroughs. On the other hand, many simply feel overwhelmed by the wide spectrum of challenges arising from new applications, new hardware and software techniques, and the need for interaction with other computer science disciplines, leaving them with no clear sense of direction.

There is an evident need for highly respected and experienced database researchers known for both their academic achievements and clear visions to convey a sense of direction, to identify promising avenues of research, to develop new perspectives and, above all, to convince the many young researchers who will have to carry the load to meet the scientific challenges of the future that there are more challenges than ever and that databases will remain an area of vibrant research.

With these objectives in their back, 21 reputed database researchers from all over Europe and from the United States met at Dagstuhl from September 9 to 12, 1991 for an intensive meeting of minds and an exchange of ideas on the challenges posed by evolving technologies and ever more demanding applications. They were encouraged to discuss half-baked ideas, speculate on future trends, report on experiences that they felt they had not been able to adequately confront so far. This report contains the summaries of their presentations and discussions.

All participants were enthusiastic on the quiet setting, intimate atmosphere and outstanding hospitality of Dagstuhl Castle which they felt contributed greatly to the success of the workshop. The organizers wish to extend their gratefulness to the IBFI organization.

F.Bancilhon
P.C.Lockemann
D.Tischritzis

From DBMS (database management systems) to URMS (universal resource management systems)

Klaus R. Dittrich
Universität Zürich

Today´s software systems are typically built as collections of large, closed components like operating systems, database management systems, language systems and similar. While this is mostly due to historic developments, it creates a lot of problems: frequently, one system has to address tasks that are similar to those solved in a different component, local decisions interfere with those made elsewhere, synergies remain unexploited, and, above all, no modular growth is possible because no "internals" can be made use of.

We propose to reconsider software systems as a whole and reconstruct them as open, coherent, well-architected systems for the management and usage of general (physical and logical) resources in a uniform way (→ URMS). A URMS would be based on a URMS-kernel providing a powerful basic abstraction concept (some sort of object + activity), together wir supporting concepts like triggering, inferencing, communication, protection and recovery primitives (mechanisms: no strategies anticipated!). This kernel is supposed to be "open" for various environments in terms of hardware and communication facilities (distribution, multiprocessor, ...). Above, sets of URMS-services would provide the functionality needed for the tests to be accomplished in a specific system: starting from OS via DBMS to other services. Applications are just a natural extension - the strict boundaries of system types diminish.

While it is clear that we have to live with and improve the "legacy" systems up and running today, research should also be radical enough to reconsider grown ad-hoc decisions made - under different circumstances - in our past.

Merging database theory and programming language theory

Bernhard Thalheim
Universität Rostock

Database theory and programming language theory are at present well developed branches of Computer Science. There are a lot of similarities of both branches which could be used for the further development of database technology. Some similarities have been used already during the development of database technology. Similarities which are useful are: integrating different programming paradigms on a higher abstraction level (abstract generic structures, control and constraints); using several approaches of object-oriented programming; using domain theory etc. in defining database semantics.

Especially for the development of design methodologies, methods of artificial intelligence could lead to a further improvement of design tools.

Open object-oriented systems for complex applications

Erich J. Neuhold
GMD-IPSI and TH Darmstadt

Many multi-media systems of the future, like systems for integrated manufacturing, distributed electronic publishing, advising on and maintaining of products as well as for communication between people, groups or enterprises will have to be open to constant changes and growth.

The object-oriented approach with powerful concepts like structural and behavioral modularisation, information hiding and inheritance for reuse offers a possible platform paradigm and an interface structure for such complex systems. We have found, however, that it will be important to provide extensibility of the o.o. language concepts themselves in order to adopt such a system to special application fields like linguistics (dictionary knowledge representation), hyperdocuments (information structuring, linking and tracking), cooperative work (argumentative and decision-making structures). In addition, a careful separation of type-concepts, like polymorphic type hierarchies, from the class concept as a hierarchy of object instance collections increases the flexibility of the o.o. tool in case preexisting structures and components have to be integrated into the system.

In addition, open large complex systems by their very nature will keep offering new information, new manipulation possibilities and new concepts. Therefore, nobody will be able to work with the system without constantly learning about its features. Knowledge based machine-learning features will be needed to support the user when he only can formulate imprecise, incomplete or even inconsistent manipulation requests from the system. Both adaptation of the system to the user world as well as learning by the user about the system have to complement each other to keep the human costs in training/using such systems at an acceptable low level.

From the Kernel to the COSMOS

Hans-J. Schek
Swiss Federal Institute of Technology, Zürich

COSMOS is the research program of the database research group at ETH Zürich. It is a natural follow-on of DASDBS, the Darmstadt Database System project. While most emphasis in DASDBS was on a database KERNEL system, the project at ETH focuses on the cooperation between database systems and their environment. The environment consists of clients asking for database service and other (application) systems offering service to the database system.

The cooperation between databases and between applications and databases should be a federation (following the Swiss tradition) rather than an integration of services under global and central control. Various degrees of autonomy have to be considered.

The research objective ist the exploration ot the architecture of a COoperative System for the Management of Objects (COSMOS). In short we are "on the way from the Kernel to the COSMOS."

Databases for Databases

Yannis Vassiliou
FORTH and University of Crete

Assuming that the predictions of many database researchers, application developers and users become true, then in the future there will exist databases with three characteristics: huge in size, very complex, multimedia contents.

Traditionally, in the database area we have dealt with size and complexity by introducing abstraction mechanisms and by using them for organizing better. Two are the most common (and successful) abstractions: horizontal (file structures, indices) and vertical (schemas, views, directories, dictionaries). These may not be enough to carry us with success in future databases.

My proposal is to introduce more abstractions (organizational principles), exploit directly and fully these abstractions, limit ourselves in specific application domains and embed the methodological guidelines.

All these can be realized with the creation of new databases through which we can access other databases. In essense, creating a continuum of databases - a new abstraction.

These databases would be different from traditional ones in that they are: visual (employ graphical representation), use many links/associations, but mainly are for exploration rather than simple querying only. The principle is that first you search to discover what you are looking for (sometimes with fuzzy queries), then you go and get it from the traditional very large database.

These ideas have been tried with preliminary success in the area of software engineering, where a software information base has been created which contains descriptions of software artifacts, thus allowing browsing/searching in multiple ways - identifying the software components needed, and then, through simple mechanisms (pointers) fetching them from a software base (repository database).

Some hints on future directions

Francois Bancilhon
O_2 Technology

Database research has been very successful in the past at transfering its results and technology to the industry world. The relational model and its success in terms of market is the best example. There are some indications now that this transfer is not as effective as in the past: the new wave of object-oriented databases does not, in its majority, find its root in the research community.

I suggest that we improve this situation by

1. improving the re-utilisation factor among research projects by financing the distribution of "industrial-strenght prototypes"
2. by discovering new ways of organizing technology transfer.

Concerning new research directions, if I consider short-term needs for technology the main areas are:

- heterogeneous database systems
- appliciation generators
- design methology for OODBMS.

Concernig long-term research, I think it is time to move the emphasis away from research in systems to research in applications. The idea is to define an application which would be ambitious enough to imply the use of many technologies and whose result would be a major breakthrough. An example of such application could be a large knowledge base for economy or education at the country or continent level.

Integrated tools for object-oriented persistent applications

Michel Adiba
Grenoble University

The integration of programming languages and database functionalities allows for providing new kinds of systems with very powerful capabilities. The approach concerns the extension of relational database systems, other approaches can be called object-oriented databases.

In this framework, we present the main characteristics of an integrated environment for object-oriented, persistent application development. We first discuss the rationale of our approach and then give our analysis of current database systems or persistent languages. Using these systems or languages is still a very difficult task because persistent application development combines the complexity of database schema design together with a software engineering problem.

We describe the main components of an integrated environment which helps the designer to define the different elements of his/her application and which provides the necessary tools for generating specific code able to run under specific systems.

Mobile databases

Peter Lockemann
University of Karlsruhe

Today´s society is highly mobile. Over his/her lifetime, a person changes his/her residence, employer, professional field more often than before. Workplaces have become much more mobile, division of labor and international cooperation have become more pronounced, travel has become a fact of professional life. The transport industry, with the automobile and the airplane, has made mobility easy even over long distances. Telecommunications allows one to bridge geographical distances and lets - figuratively speaking - the individual be any place at any time. The telecommunications industry has sped mobility, but also had continuously to cope with the ensuing new demands. Service industries have sprung up that provide all the necessities and amenities that the individual has come to expect from his/her customary place.

Part of the necessities and amenities is instant and selective access to all kinds of information. Information is also the glue that holds together the diverging or disconnected activities in business, industry and science. Database management systems (DBMS) are the technical means for collecting information from outside sources, integrating the related information, preserving its integrity, and making it available to the outside world on demand or on its own intiative.

How well are DBMS prepared to follow the transport and telecommunications industry to facilitate mobility and to adapt to the needs arising from mobility? To answer the question, we first demonstrate how the latest hardware and communication resources and developments offer a suitable foundation to make databases more mobile. This will be followed by the consideration of a number of application areas where access to large volumes of data is needed in a mobile world, and allows to draw a set of 12 conclusions as to the requirements on DBMS. With these in mind we examine to which extent present-day database technology and research in distributed, server/client and interoperable databases can meet the requirements, and whether database technology is in a position to team up with telecommunications to reach the goals. We show that a series of open technical issues - to be summarized under the heading of "intentional" or "planned uncertainty" - must be met before the requirements imposed by mobility are satisfied.

On OBJECTS (facts, dreams and fantasies)

D. Tsichritzis, Université de Genève

(Object) Classes are like poems
It is easy to talk about them
It is very hard to write a good one.

Uncertainty reasoning in deductive databases

Ulrich Güntzer
University of Tübingen

An approach to non-monotonic uncertainty reasoning, which is ubiquitous in real-life database applications, is presented. Founded on the paradigm of conditional probabilities we develop a rule-based calculus and prove its soundness. Thus the merits of doing consistent judgements in uncertain domains and the advantages of modularity and incrementality of rule-based application development come together. We give an implementation of our calculus on top of a database system with a datalog interface. In this way current database technology can be extended in order to be able to cope with uncertainty.

Spreadsheet computing revisited

Claude Delobel
GIP Altair

Spread sheets are very popular tools in business applications for end users. Today very little attention has been paid to theory underlying the basic concepts. We present briefly the main directions where extensions are possible in connection with deductive capabilities, logic programming with constraints and object orientation.

Future directions for deductive databases

Laurent Vieille
ENCP-CERMICS, Paris

Deductive Databases Technology now provides adequate support for highly declarative database systems. These techniques can be used either to build deductive database systems per se, or to be integrated in other environments. For instance, they can be used for the development of extended relational systems (consider the future SQL 2/3 standards). Or they can even be used to support declarative functionalities (queries, constraints) of object-oriented databases. We present a system architecture and an intermediate formal language, pursuing these objectives.

Supporting database applications development

Roberto Zicari
Politecnico di Milano

We need better tools to support the development of (complex) database applications. At the moment the task of the application builder is a hard task. We should give the maximum degree of freedom in developing applications. Multi-paradigms is a promising idea. Integrated toolsets is the way to help guide the designer.

As a last remark, we need better, and more easy-to-use (complex) systems, at end-user level.

Adaptable systems for database processing

Bharat Bhargava
Purdue University

We must build systems that are adaptable. Smaller components (software for algorithms) can be switched to different versions or configured in a variety of ways to meet the requirements of reliability and performance. This research has an engineering component and another scientific component. The science of adaptability requires the identification of conditions that must hold during switchover and reconfigurability. Such systems will require an efficient communication software where interprocess time is in microseconds rather that milliseconds. This requires research in identifying communication requirements of transaction processing such as multicasting, wide-area networking, and reliable communication.

Communication will play a major role in the design of algorithms that manipulate resources on multiple machines, and unless communication is efficient, modularity, adaptability, and interoperability will not be possible. We need more experimental research that provides us with data that gives us new directions for research.

One some pragmatic directions for research in databases

Shamkant B. Navathe
Georgia Institute of Technology

Database management is a very practical field. The research we do leads to technology which is made into products that are supposed to help users with their data management problems. I feel that there are some important problems that industry is facing today which we should devote some attention to. Otherwise, the gap between us and them will widen, and they will never be ready for the new technologies and solutions we will produce for them.

For most applications I see an open architecture with a "databus" on which hang different databases, software systems (including database management systems), catalogs or meta-databases, and tools with user interfaces. This general architecture applies to CAM/CAD, Office Systems, Large Factories, etc..

There are some simple problems that industry is struggling with today. We have not given them any organization-structure-driven conceptual-database multi-level schemas to work with. We cannot tell them how consistent their data is or what quality of data design in terms of lack of keys, duplication etc. they have.

We should really emphasize better tools and interfaces ranging from database design to performance monitoring tools. Multi-model, multi-paradigm support is essential; abstractions of complex functionality could be useful. Reverse engineering and knowledge extraction tools are critical. We need to provide software bridging tools to existing systems.

Finally, we can borrow ideas and results from other fields including information retrieval, artificial intelligence, software engineering methodologies and computer aided instruction. Packaging valuable information on diskettes for mass distribution in underdeveloped countries may be a quicker way than printing and making large bodies of expert acknowledgement available.

With advancing technologies of communications, broadcast databases may become feasible. Applications like robots with vision systems may demand self-adapting databases.

Issues in database languages

Serge Abiteboul
INRIA

1. A lot of efforts have to be done on database programming languages. In a global market with rapidly changing customer demands, an industry will only successfully compete utilizing extremely flexible information technology with highly available and fully exploitable data. One requirement is to provide the tools for developing simple applications in a clear manner, very fast.

2. I argue that the complexity of the problems and the increased pace of the market will impose the recourse to more generic and automatic solutions which will require solid foundations. Examples of areas requiring theories are type systems for future (polymorphic and high-order) typed languages, and knowledge bases in more intelligent database systems.

Programless World: Knowledge modeling instead of programming

Stanley Y. W. Su
University of Florida, Gainesville

The development of application and database management systems to support various areas of database applications is a very costly and time-consuming process. Writing programs using the existing programming languages is difficult even for well-trained computer specialists, not to mention the casual users of databases. The existing object-oriented database management technology can be extended to provide a high-level knowledge representation model and a declarative language to specify the structural properties, operational behaviors, and knowledge rules (triggers, constraint rules, and deductive rules) associated with not only application objects (anything of interest to an application world) but also program objects (components of software which can be modelled as objects). The model and the language can be used to specify data structures, contracts, and logics that constitute a program or a software system. The structural relationships among the components of a complex software system can be explicitly modeled as associations. It is envisioned that the future object-oriented knowledge base management system (KBMS) can be built based on an extensible kernel model whose semantics can be modified and extended to further the functionalities of the KBMS. Customized KBMS can then be constructed to suit various applications. Objects of interest found in an application domain and all software systems (operating systems, KBMS, and application systems) can be uniformly modeled and managed by the KBMS. The collection of application and program objects from the common repository of knowledge are resources which can be shared by all applications domains. The traditional programming process is replaced by the browsing and selection of useful application and program objects from the common repository and by the addition of object classes and object needed for a specific application. A complex software system can be developed evolutionarily by a series of executable and testable knowledge models. Thus, the traditional concept of programming can be replaced by knowledge modeling, and programmers can be replaced by modelers in the future. By developing the proposed KBMS technology, database researchers can play an important role in closing the gap between the users and the shared knowledge repository and also provide the technology for building application systems, KBMSs and other complex software systems.

Managing long-lived database activities by Contracts

Andreas Reuter
Stuttgart University, IPVR

The major contribution of database systems is to provide the application with means to store, manipulate and retrieve data that has to be persistent. Transactions have become <u>the</u> execution model fo all operations on persistent data. Their property of atomicity has proved very useful in defining formal consistency criteria, but on the other hand it has forced transactions to be short, unrelated computations.

In distributed systems and advanced applications, however, computations tend to be long-lived, with a lot of pre-defined control flow and structure. Consequently, the system must be able to manage such computations as persistent, identifiable objects. The Contract model provides a control structure on top of transactions that allows to describe and control such long-lived computations. It provides four basic mechanisms: explicit control flow description; context variables that are persistent and reflect the complete execution history; synchronization on shared state by pre- and post-conditions; explicit means for conflict resolution.

In case of errors, crashes or other abnormal event,s a long-lived computation undergoes forward recovery. The Contract model implies a two-tier programming model, and a notion of local consistency as opposed to the classic notion of global consistency. The latter aspect will need a fair amount of theoretical work in the future.

Distributed computing architectures for intelligent and cooperation information systems

Michael L. Brodie
GTE Laboratories, Waltham, MA

Three principles that guide my future research are:

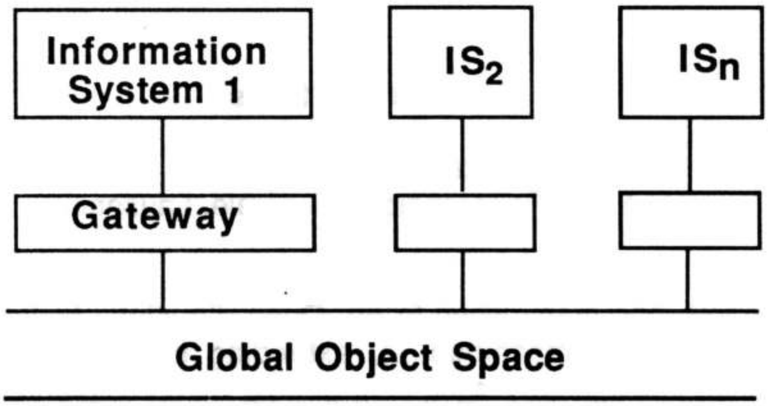
1) Contribute to solutions of real significant problems
2) High quality research
3) Contributing to making the world a better place or doing work consistent with beliefs/moral outlook.

These guidelines provide a longer view than next year's research. Rather, I consider what might be written on my goal store. Current real problem: Problems of large-scale information systems (inflexibility, biased data, software crisis) that pose some of the pressing problems for large organizations. The goal is to develop technology to facilitate the enhancement and evolution of existing and future computing resources. In addition to traditional software engineering goals (e.g., high-level tools, extensible systems), system goals (performance, etc.), there is the critical goal of interoperability.

To this end I propose a two-pronged research program:

1) *Distributed Computing Architectures*
The objective is to have a layer of distributed, sharable systems services available transparently to all applications (frameworks). This notion, sometimes called middleware, provides a challenge for a new architecture (i.e., collections of servers or objects that collectively provide the functions of systems currently referred to as DBMS, OS, GUI, name servers, asset managers, repository etc. This will involve a rethinking of existing boundaries between currently disjoint technologies.

2) *Intelligent and Cooperative Informations Systems*
The objective here is to facilitate the cooperation between possibly autonomous information systems as indicated below.



Research goals here include: adding intelligence to support cooperation (below the application level, e.g., to make transparent details such as language, format and location of other ISs) and to support applications or ISs in providing more "intelligent" functionality.

A major challenge is to deal with the legacy of existing technology and ISs. Migration from where we are to these new architectures and ICISs must be dealt with by the research community .

The Role of semantic models in DB

Arne Sölvberg
The Norwegian Institute of Technology,
The University of Trondheim

DB research directions should be determined relative to society´s needs. An investigation of long-range challenges in the application of IT among Norwegian DP managers shows that the following broad objectives are agreed upon:

1) Provide better support for cooperation among organisation and among humans
2) Provide better support for human judgement (e.g., decision support systems)
3) Improve system development efficiency
4) Develop reliable technology platforms

In order to meet these broad objectives it seems to be necessary to improve the semantic descriptions of database contents (among many other things). One approach is to make sharper distinctions between models of application domains, and models of the data about those domains. In database modelling the tendency has been to collapse these two types of models into one "Conceptual Schema" model of the database. It was proposed that the conceptual modelling field should be revived, because of its central importance for the meeting of the objectives mentioned above.

User Interfaces

Stefano Spaccapietra
Ecole Polytechnique Fédérale, Lausanne

Database research has many ways to go. Open directions include coping with new hardwares (like parallel machines,...), extending into or benefitting from other domains (artificial intelligence, logic, ...), getting better implementations using new concepts (like the object-oriented approaches). One of the most traditional, but still far from its goals, research direction is aiming at improved user satisfaction. It is my opinion that to make significant advances in this area we should concentrate on user requirements, independently of the available database technology. In other words, we should aim at "user-oriented" DBMS. One way to go along this line is to develop user interfaces first. This would fix what a DBMS should do, before one gets concerned on how to do it. As a DBMS is a complex system, we will need to establish a set of user interfaces, with the main requirement that they are consistent. This means that the user should be able to use the same paradigm to perform the same task in different contexts, but should also be able to go to a different paradigm if asking for different funtionalities.

The set of interfaces should at least cover schema definition, data manipulation and the design methodology (views, integration, ...), but could extend into new areas, like transaction management, for instance. Graphical support should be provided for all aspects, whenever applicable.

Dagstuhl-Seminar 9137 **Participants**

Serge **Abiteboul**
INRIA
Domaine de Voluceau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France
abitebou@inria.inria.fr
tel.: +33-1-39 63 55 37

Michel **Adiba**
IMAG
Laboratoire de Génie Informatique
B.P. 53x
38041 Grenoble Cedex
France
adiba@imag.imag.fr
tel.: +33-76 51 46 27

Francois **Bancilhon**
O2 Technology
7 rue du Parc de Clagny
78000 Versailles
France

Bharat **Bhargava**
Purdue University
Computer Science Department
West Lafayette IN 47906
USA
bb@cs.purdue.edu
tel.: +1-317-497-6013

Michael L. **Brodie**
Computer and Intelligent Systems Laboratory
GTE Laboratories Incorporated
40 Sylvan Road
Waltham MA 02254
USA
brodie@qte.com
tel.: +1-617-466-2256

Claude **Delobel**
GIP Altair
Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
claude@bdblues.altair.fr
tel.: +33-1-39 63 54 17

Klaus **Dittrich**
Universität Zürich
Institut für Informatik
Winterthurerstr. 190
CH-8057 Zürich
Switzerland
dittrich@ifi.unizh.ch
tel.: +41-1-257-4312

Ulrich **Güntzer**
Universität Tübingen
Wilhelm-Schickard-Institut für Informatik
Sand 13
W-7400 Tübingen
Germany
guentzer@informatik.uni-tuebingen.de
tel.: +49-7071-29 5477

Peter **Lockemann**
Universität Karlsruhe
Institut für Betriebs- und Dialogsysteme
Postfach 6980
W-7500 Karlsruhe
Germany
lockemann@informatik.uni-karlsruhe.de
tel.: +49-721-608 3968

Shamkant B. **Navathe**
Georgia Institute of Technology
College of Computing
Code 0280
Atlanta GA 30332
USA
sham@cc.gatech.edu
tel.: +1-404-853-0537/894-3152

Erich J. **Neuhold**
GMD-IPSI
Dolivostr. 15
W-6100 Darmstadt
Germany
neuhold@darmstadt.gmd.de
tel.: +49-6151-875 803

Andreas **Reuter**
Universität Stuttgart
Institut für Parallele und Verteilte
Höchstleistungsrechner (IPVR)
Breitwiesenstraße 20-22
W-7000 Stuttgart 80
Germany
reuter@ipvr.informatik.uni-stuttgart.dbp.de
tel.: +49-711-7816 449

Arne **Sölvberg**
NTH
Dept. of Electrical Engineering
and Computer Science
N-7034 Trondheim
Norway
asolvber@idt.unit.no
tel.: +47-7-59 34 38

Hans-J. **Schek**
ETH - Zürich
Institut für Informationssysteme
ETH-Zentrum
CH-8092 Zürich
Switzerland
scheck@inf.ethz.ch
tel.: +41-1-254-7240

Stefano **Spaccapietra**
EPFL
Departement d'Informatique-LBD
IN--Ecublens
CH-1015 Lausanne
Switzerland
spaccapietra@elma.eplf.ch
tel.: +41-21-693 5210

Stanley **Su**
University of Florida
Database Systems Research and
Development Center 470 CSE
Gainesville FL 32611
USA
su@pacer.cis.ufl.edu
tel.: +1-904-392-2683 / 392-2680

Bernhard **Thalheim**
Universität Rostock
Fachbereich Informatik
Albert-Einstein-Str. 21
O-2500 Rostock 6
Germany
thalheim@informatik.uni-rostock.dbp.de
tel.: +37-81-44424 ext. 64

Dionysis **Tsichritzis**
Université de Genève
Centre Universitaire d'Informatique
12 rue du Lac
CH-1207 Genève
Switzerland
dt@cui.unige.ch
tel.: +41-22-787 6583

Yannis **Vassiliou**
FORTH
Institute of Computer Science
36 Daedalou str.
GR - 711 10 Heraklion - Crete
Greece
yannis@csi.forth.gr
tel.: +30-81-22 9324

Laurent **Vieille**
ENPC-CERMICS
La Courtine
F - 93167 Noisy le Grand Cedex
France
laurent@enpc.fr
tel.: +33-1-43 04 40 98 ext. 3480

Roberto **Zicari**
Politecnico di Milano
Dipartimento di Elettronica
Piazza Leonardo da Vinci 32
I-20133 Milano
Italy
zicari@ipmel2.elet.polimi.it
tel.: +39-2-2399-3523

Bisher erschienene und geplante Titel:

W. Gentzsch, W.J. Paul (editors):
Architecture and Performance, Dagstuhl-Seminar-Report; 1, 18.-20.6.1990; (9025)

K. Harbusch, W. Wahlster (editors):
Tree Adjoining Grammars, 1st. International Worshop on TAGs: Formal Theory and Applications, Dagstuhl-Seminar-Report; 2, 15.-17.8.1990 (9033)

Ch. Hankin, R. Wilhelm (editors):
Functional Languages: Optimization for Parallelism, Dagstuhl-Seminar-Report; 3, 3.-7.9.1990 (9036)

H. Alt, E. Welzl (editors):
Algorithmic Geometry, Dagstuhl-Seminar-Report; 4, 8.-12.10.1990 (9041)

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 1990s, 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)