Information and Computer Science

University of California, Irvine

Overview of the Department of Information and Computer Science

The University of California, Irvine (UCI) was designed around the ideals of high-quality education, advanced research, and community outreach. The UCI Department of Information and Computer Science (ICS) was formed in 1968 to further these ideals in a newly emerging discipline. The field has grown considerably in significance in the ensuing years, and ICS has kept pace.

ICS offers degree programs leading to the Bachelors of Science, Masters of Science and Doctor of Philosophy in Information and Computer Science. ICS offers a large undergraduate program having over 800 majors, and offers many courses to UCI students regardless of major. The ICS undergraduate program provides all students with a firm foundation in technology, and allows students the flexibility to explore in depth advanced topics including software systems, networks and distributed systems, information systems and artificial intelligence. Students are encouraged to complement their education with courses in related fields including the arts, biology, business, economics, linguistics, management, mathematics, the physical sciences, psychology, and the social sciences. In the ICS doctoral program, students both advance the state-of-the-art in Information and Computer Science and gain research skills which allow them to continue to contribute to the advancement of the field.

Faculty research concentrates on the analysis of algorithms, artificial intelligence and knowledge-discovery in databases, embedded systems, medical informatics and computation biology, multimedia, databases, information retrieval and visualization, computer supported cooperative work, parallel and distributed processing, human computer interfaces, communication networks, programming languages, software development environments, advanced software technology, hardware-software co-design, and the managerial and social aspects of computing technology.

We have experienced many changes in ICS. The highlights include:

- ICS created a terminal M.S. degree for students interested in advanced careers in the high technology industry. Previously, the only students awarded M.S. degrees were those in the Ph.D. program.
- ICS hired three new faculty members: Sharad Mehrotra (Ph.D. University of Texas), Wanda Pratt (Ph.D. Stanford University), and Nalini Venkatasubramanian (Ph.D. University of Illinois).
- Four faculty were promoted to Associate Professor: Mark Ackerman, Rajesh Gupta, David Rosenblum and Padhraic Smyth.
- Nikil Dutt and David Eppstein were promoted to Professor.
- Daniel Gajski was promoted to Professor Above Scale.
- The Center for Embedded Computing Systems was established. Directed by Daniel Gajski, the center includes faculty from Information and Computer Science and faculty from Electrical and Computer Engineering.
- The National Science Foundation approved an Industry-University Cooperative Research Center at UCI on Information, Technology and Organizations. The center focuses on interdisciplinary research involving faculty from the Graduate School of Management and Information and Computer Science.
- With funding from both the State of California and corporate sponsors, 165 400 MHz Pentium II PCs were purchased to upgrade and expand undergraduate computing laboratories.
- ICS received a grant from the Department of Education to provide 12 full fellowships to incoming Ph.D. students.
- PC Week has named UCI one of the nation's top 10 universities for information technology education. UCI was one of only two public universities included in the listings, which highlight programs that are preparing students to fill the nation's shortage of information technology workers.

In its first thirty years, ICS has emerged as a major presence in the computing world, graduating over 3700 students making an impact in California, the nation, and the world.

Contents

Cover Photography by Michele Rousseau

Research Areas in the Department of Information and Computer Science

Artificial Intelligence

Research in Artificial Intelligence (AI) is aimed at understanding the computational mechanisms that underlie intelligent behavior, and at designing computational systems that exhibit it. The AI group at ICS is involved in research on machine learning and knowledge discovery, deductive and probabilistic reasoning, constraint satisfaction techniques, neural networks and cognitive architectures, sophisticated image and signal processing, scientific reasoning in domains such as molecular biology, medicine and space science, intelligent web-based agents, and the psychological investigation of human learners. The group is interested in basic research into the fundamental principles of intelligence; the methods by which knowledge is acquired, summarized, organized, and utilized to solve complex problems; the construction of computational artifacts that support algorithmically, cognitively, or conceptually challenging tasks and embody behavior associated with intelligent systems; and applications that confront intelligent systems with real-world tasks.

Machine Learning

Machine learning investigates the mechanisms by which knowledge is acquired through experience. Research at UCI spans the spectrum of models for learning, including those based on statistics, logic, mathematics, neural structures, information theory, and heuristic search algorithms.

Our research involves the development and analysis of algorithms that identify patterns in observed data in order to make predictions about unseen data. New learning algorithms often result from research into the effect of problem properties on the accuracy and run-time of existing algorithms.

We investigate learning from structured databases (for applications such as screening loan applicants), image data (for applications such as character recognition), and text collections (for applications such as locating relevant sites on the World Wide Web). UCI also maintains the international machine learning database repository, an archive of over 100 databases used specifically for evaluating machine learning algorithms.

Knowledge Discovery and Data Mining

Databases with millions of records and thousands of fields are now common in business, medicine, engineering, and the sciences. The problem of extracting useful information from such data sets is an important practical problem. Research on this topic focuses on key questions such as how can one build useful descriptive models that are both accurate and understandable? Probabilistic and statistical techniques in particular, play a key role in both analyzing the inference process from a theoretical viewpoint and providing a principled basis for algorithm development. Ongoing projects include the integration of image and text health-care data for finding diagnostic rules, automated analysis of time-series engineering data from the Space Shuttle, and discovery of recurrent spatial patterns in historical pressure records of the Earth's upper-atmosphere.

Automated Reasoning; Constraint Based Reasoning

Automated reasoning investigates methods by which knowledge is represented and used to emulate human-like thought processes. Although most reasoning tasks were found to be computationally hard, it is believed that approximation methods based on tractable models can effectively cover a significant portion of intelligent activities. Accordingly, research at UCI is focused on: 1. identifying structured knowledge-domains that are tractable for reasoning, 2. compiling knowledge-bases into such tractable structures, 3. developing reasoning algorithms that will be effective in practice, 4. exploiting parallelism and distributive computation, inspired by brain architecture. In summary, UCI researchers investigate developing flexible and expressive representations that accommodate efficient reasoning.

Research at UCI has focused on constraint-based reasoning as the primary model for addressing these issues as it unifies and cuts across many traditional areas in Artificial Intelligence. Constraint processing is a paradigm for formulating knowledge in terms of a set of existing or desired relationships among entities, without specifying methods for achieving such relationships. A variety of constraint processing techniques have been developed and applied to diverse tasks such as vision, design, diagnosis, truth maintenance, scheduling, default reasoning, spatio-temporal reasoning, logic programming, and user interface.

The current focus at UCI is on extending the constraint model to new areas of reasoning, such as belief networks and decision theoretic planning, and to new computational paradigms.

Brain Modeling

The mind is what the brain does, and the brain is now coming to be understood as a machine. As such, the anatomical architectures and physiological operation of specific brain circuitries can be analyzed formally. Such investigations have led to the identification of mechanisms by which these biological design features can dictate the recognition, memory and motor capabilities of these circuits. Of primary scientific interest is increasing our knowledge of how the brain actually works; also of interest is the engineering application of such knowledge to the construction of novel devices that have brainlike abilities. Modeling of particular brain circuits has led to the derivation of novel and powerful computational devices for tasks ranging from signal classification, temporal signal processing and memory storage to motor coordination and robotics. As might be expected of circuitry evolved to process complex environmental information, these devices have been shown to equal or outperform the best extant engineering approaches on a range of difficult applications such as radar detection and medical image analysis. Devices based on this work are now installed and used at U.S. Navy and other laboratories, and are under commercial development for signal processing uses in a number of applications domains.

Biomedical Computing: Computational Biology and Medical Informatics

There are a number of significant problems in biology and medicine for which computational approaches can yield important insights. The world-wide efforts to construct databases of protein and small molecule structures, DNA sequences, metabolic pathways, regulatory mechanisms, pharmaceutical structures and activities, patient response data, etc., have created many opportunities for intelligent systems. Central questions include: What function is encoded in a protein sequence? What structure will it fold into? How can we make better pharmaceutical drugs? What factors effect patient response to treatment? Partial solutions to these problems have been found using extensions of research on knowledge representation, search, and learning.

ICS faculty are involved in a project to develop a knowledge-based systems for recommending a customized multiple-drug therapy for HIV infected patients. We are also exploring an approach to creating knowledge-based systems by learning from patient data and have identified guidelines for screening for forms of dementia such as Alzheimer's disease. We have developed knowledge-based approaches to protein structure prediction, and implemented novel sequence-structure search algorithms and recognition methods. We are modeling DNA mutation and repair in connection with cancer-related studies.

The biomedical domain is a rich source of application problems on which to test AI methods, and we welcome inquiries from academic and industrial colleagues with interesting biomedical research problems amenable to an AI-based computational approach. AI methods are applicable to many of the questions in computational biology.

Computer Systems Design

New application areas are placing increasingly challenging demands on existing computer systems. For instance, some estimates indicate that a typical person in North America comes into contact with over eighty computing devices in the course of a day. In the near future, we can expect to see computing devices being embedded into biological systems. The rapidly expanding number of application domains, coupled with an increasingly fast-paced market place, creates a unique system design challenge for each design that must be addressed by increasingly smaller teams of system designers and even smaller groups of hardware and software designers within a matter of a few weeks. What makes this possible is sophistication in design methods that are strongly grounded in sound analytical principles. These methods have led to the evolution of a design science that uses formal models and methods to assist advancements in various applications and technology areas. Thus the future development of efficient and high performance embedded computer systems requires a good understanding of the complex interactions between applications, software and hardware.

Research in Computer Systems Design addresses the various aspects of bridging the gap between the demands of new applications and available technology. The CSD concentration is designed to produce computer scientists with an increased awareness of the demands imposed on computers by the application domains which have traditionally been viewed as extrinsic to computer science. This application sensitivity will give students a unique advantage in the increasingly important area of integrated software/hardware computer and information systems and will prepare them to meet the challenges of real-world problems. The CSD research paradigm prepares our students to conceptualize a system design, prototype it and take it all the way to an efficient system implementation with the right balance of hardware and software components.

The CSD concentration can be divided further into five inter-related subfields: System Specification and Modeling, Hardware Design Synthesis, System and Application Software Synthesis, Design Environments, and Hardware/Software Codesign. The common thread that binds these subfields together is the search for effective ways of organizing computational resources into complex systems and the need to balance hardware and software solutions during systems development.

System Specification and Modeling

As computing systems continue to become more complex and time-to-market shrinks, system-level designers need powerful specification and implementation languages and techniques that enable rapid conceptualization of the intended functionality and design constraints. The given specification can be validated for correctness and can be simulated under different operating conditions to determine the design's desired environmental and operating constraints. The design specification can then be refined into an implementation composed of hardware and software components. Verification of the refinement is necessary at each design step to ensure integrity of the intended behavior within the design constraints. After design implementation, the final product has to be tested for possible defects, both in hardware and software. Researchers in the design sciences area have developed several conceptualization tools that span the system and behavioral levels to enable rapid specification, validation and testing of designs from concept to implementation.

Hardware Design Synthesis deals with the translation of specification or abstract design descriptions into design structures consisting of components from a given technology library. This translation may be performed on several different levels of abstraction (system, microarchitecture, logic, and layout level) depending on the complexity of components. On the system level, we start with executable specifications and generate system descriptions using processors, memories and switches as components, and software to execute the specification on the generated system. On the microarchitecture level, the design descriptions generated include registers, counters, ALUs, and multipliers. Logic design is expressed in terms of gates and flip-flops, while layout description contains transistors, wires, and contacts. Our research group has developed several design tools that use a mixture of algorithmic and heuristic techniques to generate designs that are competitive in quality with designs generated by human designers.

System and Application Software Synthesis deals with the generation of the the software subsystem for systems-on-silicon that contain several (possibly heterogeneous) programmable processors, as well as reconfigurable hardware blocks and large amounts of on-chip memory. While software is already a significant component in today's embedded systems, it is expected to greatly dominate future generations of systems-on-silicon. Software synthesis techniques include the generation of real-time operating systems, compilers, code generators, and component-based embedded software for different types and versions of programmable processors. Our research group is actively pursuing various facets of embedded software synthesis, including coarse-grain parallelism extraction for the multiple processors on chip, instruction-level parallelism exploitation for individual processors, and retargetable code generation for versions of the on-chip processors. In addition, we are developing analytical techniques for estimating system memory requirements, code size, execution time, and power dissipation for embedded software.

Design Environments

The design process for computing systems spans several levels from abstract communicating processes to their realization into software and hardware. Since this design process is very complex, complete automation is often inadequate or even impossible. An interactive design environment can allow expert human designers to guide the design process, initially towards feasible solutions and later towards optimal implementations. Such a design environment requires an infrastructure of powerful front-ends and languages, robust, multi-level design representations, accurate estimators to predict the effects of design decisions and facilities to support interactive design refinement from concept to implementation. The design environment also needs a powerful database that can manage different design versions, and provide rapid feedback on design constraints for different implementations. Researchers in the design sciences area are working towards building interactive environments to support incremental design refinement to help expert human designers efficiently explore design alternatives and generate optimal design implementations.

Software/Hardware Codesign

Complete computer-based systems include both hardware and software components. The current technological trend is to migrate computationally intensive parts of software into application specific circuits (ASICs). Thus a complex system may consist of several standard processors and several ASICs with software being partitioned among them. In the past software and hardware were conceived, developed and tested separately. New system design methodologies require that software are hardware are codesigned and traded off with respect to performance and cost. Thus, a system must be conceptualized without any special implementation in mind. The specification must be partitioned into software and hardware parts, comparing different quality metrics for each part and then allocated to different standard or custom components for implementation. Researchers in the design sciences area have developed new algorithms for software/hardware partitioning, estimation, interface synthesis, and design space exploration, and have embedded them into a framework for software/hardware codesign.

Computing, Organizations, Policy, and Society (CORPS)

UCI is an internationally recognized center for research on the social and managerial dimensions of computerization, computer-supported cooperative work, and human-computer interaction. One thread of the CORPS group's research examines the impacts and policy issues that surround computerization, including high-level system requirements. Other threads focus on computer support for groups. These include field studies of groupware deployment and use, as well as the design and development of prototype systems. The latter include systems to support organizational learning and memory, as well as systems that provide social awareness and filtering over networks including the Internet and WWW. Faculty in CORPS and Software are also very active in studies of human-computer interaction and development of user interface tools.

CORPS researchers study these topics in various public and private settings including government agencies, technology companies, and other commercial enterprises. Examples of specific projects include studies of new WWW applications, technology use in the air transport industry, electronic calendar use in organizations, agent-based information retrieval over the WWW, digital libraries, and studies of interface design consistency.

CORPS researchers have focused on understanding the "reality" of computerization, in contrast with utopian promises and antiutopian fears. Most of the research is empirical and conducted "vivo" - in organizations or other social settings where computer-based systems are developed and used routinely. The empirical work is tied to important lines of theory development, especially with respect to the organization of work and organizational control patterns.

This work has lead to a conceptual approach which is distinctive of "the Irvine school" of analysis. We examine computerization as a social and technical process. Technologies are social and organizational creations with important political characteristics; they are not just a set of artifacts whose information processing characteristics are most important. Moreover, computer-based technologies are developed over time in social settings whose character is shaped through their histories. Our conception of computer science is anchored in a view that the development and use of computer-based systems has essential social features.

One byproduct of the CORPS group's research is a broad ranging concern with "advanced requirements analysis." CORPS researchers do not take requirements as given, but inquire how they are constructed from the preferences of participants, how they change, how people form perceptions about them, how people and groups mobilize support for them, and how they shape the tasks of producing new systems.

A unique feature of the CORPS program is its emphasis on social issues within a department of information and computer science. It is the only program of its kind in the United States. CORPS research is necessarily interdisciplinary, drawing on the disciplines of sociology, economics, political science, administrative science, and history. Students and faculty address both the technical and organizational aspects of computing technology, thus strengthening both aspects of analysis.

Software

Research Emphases

Software research at UCI encompasses several major areas of study typically labeled as within software engineering, as well as several areas not so typically labeled. "Software" was chosen as the name of the area to reflect the existence of perspectives in addition to that of engineering, namely science, mathematics, and human-computer interaction. The central goal of all the research is improvement in software development, evolution, deployment, quality, understandability, and cost-effectiveness. The context for the research is broad: the software group is interested, for example, in various issues associated with the World Wide Web (WWW) as well as the traditional topics associated with development of large-scale, complex systems.

Specific research emphases are as follows: software architecture, hypermedia, analysis and testing, software understanding, environments, user interface software, process/workflow, distributed component-based systems, extensible systems, and mobile code. All the faculty are involved in various aspects of both software environment research and software architecture. Hypermedia is a particular emphasis of Taylor's. Analysis and testing is a focus of Richardson and Rosenblum; understanding is a focus of Redmiles and Richardson. Taylor and Redmiles focus on work in developing interactive software. Taylor, Richardson, and Rosenblum all have interest and work in the process and workflow area. Distributed component-based systems, especially problems related to their validation, is a key focus of Rosenblum's research. Extensible, component-based systems and mobile (machine-independent) code are the specialty areas of Franz.

Software Architecture

UCI's research in software architecture is directed at reducing the cost of application development by focusing attention on high level design issues and increasing the potential for reuse among systems in closely related product families. UCI's work in software architecture provides style-based design guidance, component-based architectural composition, architectural visualization and analysis, system generation capabilities, modification of systems at runtime, and architecture-based rationale capture and delivery.

Research challenges arise from numerous, and sometimes conflicting goals. For example it must be possible to effectively utilize existing custom and off-the-shelf components; it must be possible for components to be developed independently, in different languages and by different authors; architectures must be able to accommodate components of various granularities and running in heterogeneous, distributed, multi-user environments. Prototypical solutions are being assembled in the context of an integrated architecture-based development environment including graphical design facilities, active design critiquing, a runtime system for dynamic architectures, and tools for subtyping and wrapping architectural components. The environment is integrated with the WWW, supporting a component marketplace model of distribution.

Hypermedia

Hypermedia can be an effective technology for helping manage the myriad of heterogeneous artifacts, systems, and relationships which exist in large-scale software engineering projects. Pursuing this objective, the hyperware group is developing open, heterogeneous, distributed hyperprogram technology. The group's focus is on developing mechanisms and standards to integrate link-server hypermedia functionality with the WWW, while adding support for hyperweb configuration management, semantically-rich link generation, cooperation agents, and semi-automated rationale capture of software development processes.

One product of this research is Chimera, an open hypermedia system with explicit support for heterogeneity, augmented by a deep integration with the WWW. Chimera provides a set of flexible abstractions which enable the integration of hypermedia services into familiar tools. n-ary links, view-specific anchors, and hierarchical hyperwebs are all provided. Hyperwebs may be local or remote, and may be accessed via HTTP.

Another aspect of the research is directly focused on improving the WWW infrastructure to support the needs of software engineers. UCI's work on the WWW standards, in collaboration with the World Wide Web Consortium (W3C) and the Internet Engineering Taskforce (IETF), has included writing the core protocol specification for the HTTP protocol and Uniform Resource Identifiers (URI) standard. Current work includes a focus on the next layer of enabling technology: distributed authoring and versioning on the web (WebDAV).

Analysis and Testing

Professor Richardson's primary research is directed toward the integration of formal specification methods and analysis with software testing. Her current work focuses on enabling testing technology to be applicable throughout the software lifecycle, especially specification-based testing approaches. Richardson is developing capabilities to support analysis and testing thoughout the software lifecycle, from early requirements analysis through operational use. She is also extending specification-based testing techniques to be applicable at the level of software architecture. Richardson collaborated on developing analysis and testing capabilities within a process-centered environment to support integration of and experimentation with a variety of techniques. She developed ProDAG, a program dependence analysis toolset that provides automated support for software understanding, debugging, testing adequacy criteria, and maintenance, and TAOS, a testing environment that supports management of test assets, monitored test execution, automatic test result checking, and test coverage measurement. More recently, she developed the EASOF model of specification-based testing with support for execution-time checking of test results against formal specifications of required behavior.

Professor Rosenblum is studying the problem of selective regression testing, an approach to regression testing in which static and dynamic test-coverage and program-change analysis are employed to eliminate unnecessary test cases during maintenance of a software system. Cost-effectiveness is a fundamental concern in selective regression testing, since the cost of the analysis employed by a method can easily exceed the savings it achieves in test execution time. Rosenblum has developed strategies for using test coverage information from a single version of a system to predict whether a selective regression testing method will be cost-effective for use on all future versions of the system. In addition, he is undertaking comparative evaluations of different selective regression testing methods to further identify the factors that influence the cost-effectiveness of a method.

Software Understanding

In many software development projects, communication of information is not always synchronous between two human beings. Members of a development team may be widely spread geographically or a team may work at a distance from the intended end users of the software being developed. Moreover, developers may desire to reuse software components developed years before. Researchers at UCI are investigating a variety of techniques for both capturing information about software that can later be used to help in its understanding as well as techniques to assist directly with the task of software understanding and development.

In evolving software systems, for example, designers face many cognitive challenges. Individual designers typically do not have all the knowledge they need to make informed decisions, they lack knowledge to order decisions effectively, and their understanding of complex systems requires major mental effort. The Argo software architecture design environment explores how software can provide support for these cognitive challenges of design. The specific kinds of support Argo provides are based on cognitive theories of reflection-in-action, opportunistic design, and comprehension and problem solving. The Knowledge Depot system enhances project awareness among all members of a project and authorized parties. Project awareness allows people to maintain an understanding of changes in design and scheduling of components that they depend upon. This type of information is valuable for coordinating interdependent project teams. The system captures email and, based on the subject of the message, forwards a summary to users who have registered an interest in the subject. In addition to email, the system also stores and organizes other types of textual and graphical documents. This structured information creates a group memory that assists groups in capturing design rationale and project history data.

Richardson's research also addresses software understanding: ProDAG, a program dependence analysis toolset provides automated support for software understanding by allowing the software engineer to scrutinize and investigate features of program dependences between components.

Environments

Environments research is pervasive at Irvine. The fundamental objective of the environments work is to develop technically rich solutions across the breadth of the problem (developer support, product architectures, tool technologies) that are technically compatible, functionally comprehensive, and mutually reinforcing. The focus areas include interoperability, process, analysis and test, user interfaces, and understanding. In a very real sense the topic of environments is the bringing together of the results of the research in the other focus areas.

User Interface Software

One aspect of this research, that pursued by Professor Redmiles, combines topics from human-computer interaction and software engineering, focusing on the processes and technologies needed to develop and support useful and usable interactive software. The research conceptualizes evolutionary software development as a process of on-going communication. Methods such as protocol analysis, statistical testing, and cognitive walkthroughs are adopted. Current research is investigating the use of intelligent software agents both as a means of enhancing the communication model and reducing the cost of incorporating usability information. For example, agents can facilitate synchronous or asynchronous dialogs between end users and developers or automatically observe and collect data on usage patterns for prototypes. In general, the goal is to more tightly couple development to end users' situations.

Another aspect of the work in interactive systems is Professor Taylor's work in user interface software systems and architecture. This work is directed at reducing long-term costs associated with graphical user interfaces for large-scale, complex applications by introducing series of layers that insulate components of an application from other components that may experience change. This work has also explored the use of multi-threaded architectures for GUI systems.

Process/Workflow

Few tasks in industry are performed by individuals acting alone. Almost always an individual's work assignment is part of a larger activity or process. Effectively participating in the larger activity requires good communication and team coordination. Researchers in computer supported cooperative work (CSCW), workflow, and software process are all tackling various aspects of this broad problem. Researchers at UCI are focused on bringing together lessons from all three communities, identifying effective abstractions and developing computer-based techniques and approaches for facilitating the work of small to medium-sized distributed project teams.

Research topics, strategies, and approaches include object-oriented modeling and programming formalisms, jointly utilizing procedural and rule-based process descriptions, facilitating technology insertion via leveraging Java and component object technologies, exception handling, dynamic change and customization, the modeling of roles and organizational structure, and integrating formal process support mechanisms with technologies designed to support informal communication. One vehicle for exploring these issues and demonstrating solutions is Endeavors, an open, distributed, extensible process execution environment which combines a sophisticated process modeling language with features designed for easy customization by both technical and non-technical users.

Component-Based Software and Extensible Software Systems

Many of our preconceived notions of software need rethinking as the computer industry is moving forward to embrace software portability schemes such as Java and compound-document architectures such as OpenDoc and JavaBeans. Instead of providing monolithic application programs that require only a limited amount of interprocess communication, many software systems in the future will have a much finer-grained internal structure. They will be composed of a host of quasi-independent, dynamically-loadable applets that cooperate in such a way that they appear to the end-user as a single unified application, and they will be extensible by further such applets at run-time. For example, the different data types of a multimedia document integrating text, graphics, and video might at run-time be supported by separate part editors that could be downloaded as needed, each of them responsible for only a sub-section of the document.

As a consequence of their highly dynamic nature, such run-time extensible systems cannot be exhaustively tested. They have no final form or final integration phase, nor can they be subjected to final total analysis. Yet, they need to provide mutual independence of future extensions constructedby independent authors. Simultaneously satisfying the requirements of run-time extensibility, component independence, safety, and efficiency is extraordinarily difficult.

Current object-oriented development practice is centered around application frameworks. This approach distracts from the ultimate goal of composing software out of pre-fabricated components originating from different sources. Professor Franz is investigating programming languages that model the component approach more faithfully and is working on compilation techniques to implement these new languages efficiently. In particular, the programming language Lagoona that is being developed at UCI is centered around a model of software composition that is based on passing of "first-class messages" rather than on inheritance.

In most object-oriented programming languages, messages and the methods that get executed in response to receiving them are only "second class citizens". In these languages, one can send a message to an object, but one cannot further manipulate the message itself as a data object. As a consequence, many of the operations that a naive observer might expect to be available are in fact not usually offered. Examples of such missing operations are the ability to store arriving messages in a data structure and execute them asynchronously later, perhaps in a different order, or the capability of forwarding a received message to another object without first having to decode it. Lagoona, on the other hand, supports "first-class messages" efficiently. This additional language capability greatly simplifies the design of extensible, component-oriented systems, and leads to much more uniform overall system architectures.

Professor Rosenblum's research in this area is currently focused on problems related to the validation of distributed component-based software systems, especially systems built using object-oriented component models and event-based infrastructures. This research has three specific thrusts: (1) development of a theory of test adequacy for component-based systems; (2) investigation of post-deployment monitoring techniques that can be used when opportunities for full-scale pre-deployment testing are limited; and (3) investigation of event observation and notification methods to support rapid construction of event-based distributed applications over wide-area networks. Much of this research requires a deeper understanding of the relationship between software architecture and component-based development. Hence the research also involves investigation of the architectural foundations of component-based software engineering. These problems are currently being studied in the context of the JavaBeans component model and the C2 architectural style.

Mobile (Machine-Independent) Code

Virtual Machines (such as the one used for the distribution of Java "applets") do not scale well. The complexity of verifying an incoming byte-code sequence for correctness and the complexity of generating optimized native code on-the-fly both grow faster than linear program size. This is a strong indication that eventually the Java Virtual Machine will acquire competitors that provide additional capabilities beyond mere machine-independence. Professor Franz is systematically studying alternative machine-independent software distribution formats and is working on just-in-time compilation techniques.

To demonstrate that it is surprisingly simple to provide an alternative to the Java platform, Franz' group at UCI is developing such an alternative called "Juice". Juice is based on a much more advanced intermediate representation than Java and implemented in the form of a browser plug-in that generates native code on-the-fly. Once the appropriate Juice plug-in has been installed on a Windows PC or a Macintosh computer, end-users can no longer distinguish between applets that are based on Java and those that are based on Juice. The two kinds of applets can even coexist on the same Web-page.

Theory: Algorithms and Data Structures

The goal of research in theoretical computer science is to produce results, supported by rigorous proof, about problems dealing with computers and their applications. The questions to be investigated are often motivated by practical problems, but the goal of understanding the fundamental structure of the problem is often as important as producing a solution of immediate applicability. Despite this emphasis, it turns out that results that first might appear to be only of theoretical value are sometimes of profound relevance to practical problems.

In particular, one of the major subareas of theoretical computer science, and the one pursued by the faculty and graduate students at UCI, is concrete complexity: We look at specific problems and try to determine the complexity (i.e., the amount of resources required) for obtaining a solution. Our work falls into three main areas: design of algorithms and data structures; analysis; problem complexity.

Design of Algorithms and Data Structures

Given a problem, we try to find efficient solution methods. A data structure is a way of organizing information; sometimes the design of an appropriate data structure can be the foundation for an efficient algorithm, and we have made a number of significant contributions to the field of data structures. In addition, one of our members has written a number of excellent texts on data structures.

In addition to the design of new data structures, we are also interested in efficient algorithms for problems arising in a variety of fields. Often such problems can be represented in terms of trees, graphs, or strings, and we are interested in the design of efficient solutions for such problems.

The field of computational geometry investigates the complexity of problems involving two-dimensional (or higher) spaces. This is an active research area which has not only theoretical depth but also practical applications in areas such as pattern recognition, VLSI layout, statistics, and image processing. One major area of our work is the investigation of certain properties of geometric constructs which can be modeled by graphs. We have also explored how solutions to geometric problems such as linear programming or the minimum spanning tree can be made dynamic, i.e., how we can efficiently maintain the solution when the input data are subject to change.

Also of interest is the compression of data. For example, we have reduced the complexity of algorithms for compressing strings, and have also investigated the compression of structures such as quadtrees which are used for storing spatial data.

Current work in genetics provides an exciting application area for algorithms. Some work done long ago by our present faculty, on longest common subsequences and on PQ-trees, has turned out to be related to problems that arise in genetics. More recently, one of our faculty has introduced sophisticated new methods for speeding the solution of problems such as DNA sequence comparison.

Much of our work has dealt with the fast solution of problems by a single processor. The combination of declining cost of processors and the desire for fast solutions to problems has led to a great deal of interest in the use of parallelism to speed up computation. One natural question is thus: how long does it take to solve a given problem with a limited number of parallel processors? Some of us have been especially interested in solving problems on graphs very quickly without using an excessive number of processors.

Analysis

Once a solution method has been proposed, we seek to find a rigorous statement about its efficiency; analysis of algorithms can go hand-in-hand with their design, or can be applied to known algorithms. Some of this work is motivated in part by the theory of NP-completeness, which strongly suggests that certain problems are just too hard to solve exactly and efficiently all of the time. It may be, though, that the difficult cases are relatively rare, so we attempt to investigate the behavior of problems and algorithms under assumptions about the distribution of inputs.

Our group at UCI has made major contributions in the area of probabilistic analysis. We have done work in algorithms for problems such as packing, partitioning, marking algorithms, and hashing. In particular, we have obtained a surprising result about the behavior of a well known marking algorithm, and an elegant analysis of double hashing.

Probability can provide a powerful tool even when we do not assume a probability distribution of inputs. In an approach called randomization, one can introduce randomness into the algorithm itself so that even on worst-case input it works well with high probability. For example, for the classical List Update Problem, which involves organizing data so that we can perform searches efficiently, one of our faculty has shown how to use randomization to beat the inherent limit of a deterministic approach.

An area of considerable recent interest is on-line algorithms. Here we investigate the performance of algorithms which must provide answers based on part of the input before the remainder is available. A good example is memory paging--when swapping, the system must decide which memory pages to keep in its cache before it sees which ones will actually be used later. Earlier analysis of this problem had not been fully successful in explaining why a common heuristic performs so well. One of our faculty developed a new approach which formally models locality of reference, and thus can better explain the performance of paging algorithms.

Problem Complexity

When efficient solutions appear difficult, negative results can sometimes provide very helpful guidance. Two major types of results are possible here:

a) In some cases one can actually prove that, under some model, the problem does not admit solution without a certain level of resources.

b) For many problems, good bounds of the above type are not available, but the problem can be shown to be equivalent in complexity to some well-known class of problems. For example, if a problem is NP-complete it cannot be solved in polynomial time unless P=NP, which is a major open question.

Such results can save wasted effort by researchers, and in some cases might also suggest that algorithms from a different model should be considered.

Computer Systems and Networks

This area of research and study is concerned primarily with the development of systems software, that is, the software infrastructure that makes the development and use of applications possible. This includes compilers, operating systems, network systems, performance tools, and software development environments, which are studied from both theoretical and practical points of view.

Our research group is particularly interested in systems software in the context of parallel or decentralized systems, that it, systems having multiple CPUs, including computer networks and both shared- and distributed-memory multiprocessor architectures. These offer great new opportunities in distributed computing and communication as well as performance-oriented scientific computing, but also pose major technological challenges to researchers and developers of parallel/distributed computer systems. The specific topics studied by the faculty members of this concentration area include the following: design and study of parallelizing compilers; design and analysis of high-speed multimedia networks; wireless networks and mobile computing; object oriented frameworks for distributed systems; approaches to portable and mobile programs; development of coordination paradigms for distributed applications; support for distributed simulations; design and study of extensible and run-time adaptable operating systems and software environments; system support for high-performance scientific computing; distributed algorithms. The major foci of our research activity are the following:

Distributed Computing

Computer networks have experienced explosive growth recently, primarily due to the steadily decreasing costs of hardware. They permit not only the exchange and sharing of information and services but, collectively, represent a major computational resource, potentially exceeding the capacity of any supercomputer. One of the goals of our research is to devise ways of exploiting the untapped potential of workstations interconnected via local area networks for the purposes of general purpose computing, that is, to solve large compute-intensive problems in a variety of application domains. This include parallel programming issues, such as problem partitioning, inter-process communication, and synchronization, as well as runtime systems issues, including performance monitoring and evaluation, visualization, and fault-tolerance. We are particularly interested in mobile code, that is, processes or threads of execution that are capable of migrating through the underlying computing network at runtime. Our main application areas are currently biomedical simulations, for example, simulating the dynamic properties of the human cardiovascular system or the distribution of a toxic substance or drug through the various organs of a human body over time. We are also interested in simulations of group behaviors in biology, ecology, or artificial life, or particle interactions in physics.

Computer Networks

In the past few years, the demand for communication services of all kinds has intensified. Applications requiring voice and data, as well as video, are rapidly expanding. Hence, future communication services must be able to facilitate a wide variety of diversified services in a practical and easily expandable fashion.

ICS researchers have been investigating various issues in design and analysis of high speed networks for multimedia applications. Work by ICS researchers in this field includes:

(1) multicast video over ATM
(2) ATM traffic control schemes
(3) a distance learning system that utilizes ATM and satellite networks to provide interactive video courseware for K-12
(4) agent-based, object-oriented mobile computing
(5) quality of service algorithms for wireless networks

Extensible Software Systems

Future software systems will have a much finer-grained internal structure than today's. Using compound-document architectures such as OpenDoc and JavaBeans, they will consist of a host of quasi-independent, dynamically-loadable "applets" that cooperate in such a way that they appear to the end-user as a single unified application. Since these systems are extensible at run-time by downloadable components, comprehensive analysis and testing of the system as a whole becomes impossible. Run-time extensibility, component independence, safety, and efficiency are conflicting goals that need to be simultaneously satisfied before any software component technology becomes commercially viable; unfortunately, existing prototypes today often sacrifice certain safety or efficiency concerns to achieve extensibility. Our research is directed at eliminating these shortcomings: we are devising a software architecture and an environment for the construction of modular software extensions. We are investigating programming languages that make software extension easier and safer. And we have pioneered a just-in-time compilation technology that makes downloadable software components independent of the target machine on which they will eventually run, without sacrificing efficiency.

Parallel Processing

Recent technological advances have made it possible to manufacture computer systems consisting of many independent processors and memory modules. The main problem with such multiprocessor machines is programmability, i.e., how to extract the necessary parallelism from a given program to take advantage of the underlying parallel hardware. Our main focus is on parallelizing compilation techniques to extract parallelism from conventional languages. Parallel programs, especially those with truly critical time (speed) requirements, are difficult to design. The process is extremely error-prone, tedious and time-consuming. The first goal of our work is to design and implement a system of program transformations that support the semi-automatic (and eventually fully-automatic) exploitation of substantially all the parallelism available in a given program.

Mark Ackerman

Associate Professor
ackerman@ics.uci.edu
http://www.ics.uci.edu/~ackerman/

Computer-Supported Cooperative Work, Human-Computer Interaction, Social Analysis of Computing Systems, Knowledge Management, Information Access

Dr. Ackerman's research focuses on how to make systems usable, especially for groups of people. Knowing why systems are usable requires a dual emphasis on both building prototype systems as well as studying how systems are used. This research falls within human-computer interaction (HCI) as well as computer-supported cooperative work (or groupware).

Dr. Ackerman's current research emphasizes information access, examining how groups and organizations can effectively store and retrieve their accumulated knowledge. His system, Answer Garden, has been used and studied in a number of organizations. His research group is also working on collaborative refining of knowledge, privacy critics, intelligent agents for information access, and the social analysis of Internet-based virtual worlds. Dr. Ackerman has also published on topics in user interfaces, computer-supported cooperative work, cooperative information systems, multimedia, and HCI.

Selected Works

McDonald, David W., and Mark S. Ackerman. "Just Talk to Me: A Field Study of Expertise Location." Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW '98), November, 1998, accepted for publication.

Ackerman, Mark S., Debby Hindus, Scott Mainwaring, and Brian Starr. "Hanging on the Wire: A Field Study of an Audio-Only Media Space." ACM Transactions on Computer-Human Interaction (TOCHI), March, 1997, 4 (1), pp. 39-66.

Ackerman, Mark S., and David McDonald. "Answer Garden 2: Merging Organizational Memory and Collective Help." Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW '96), November, 1996, pp. 97-105.

Ackerman, Mark S., and Brian Starr. "Social Activity Indicators for Groupware." IEEE Computer, June, 1996, 29(6), pp. 37-44.

Starr, Brian, Mark S. Ackerman, and Michael Pazzani. "Do-I-Care: A Collaborative Web Agent." Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'96), v. 2, short papers, April, 1996, pp. 273-274.

Ackerman, Mark S., and Leysia Palen. "The Zephyr Help Instance: Promoting Ongoing Activity in a CSCW System." Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'96), April, 1996, pp. 268-275.

Lubomir Bic

Professor
bic@ics.uci.edu
http://www.ics.uci.edu/~bic/

MESSENGERS: Distributed Computing using Autonomous Objects

The project's objective is to develop a new technology for distributed computing based on the Autonomous Objects (AO) paradigm. A conventional distributed system consists of processing nodes communicating with each other via messages. Autonomous Objects provide an entirely different model of computation. Each network node has the same program--a basic interpreter. Messages, called Messengers in our system, then carry complete programs, rather than data. These are interpreted by all receiving nodes concurrently. Interpretation is incremental in that each node interprets a small portion of the received program and passes the rest of it on to one or more of its neighboring nodes. This is repeated until the given problem is solved. The self-replication of the program along existing links allows Messengers to "colonize" a portion of the underlying computational network and utilize it as a computing resource, without any prior knowledge of its size or topology. The best intuitive way to characterize Messengers is to compare them to message-passing systems. While the latter can be viewed as collections of "intelligent" but stationary agents communicating with one another through "dumb" messages, autonomous-objects-based systems consist of "dumb" stations (interpreters), being visited by mobile "intelligent" agents.

The main advantage of the Messengers paradigm is that the resulting applications are inherently open-ended. Typical examples of such applications are those requiring human intervention while the computation is in progress, e.g., simulation scenarios where humans are integral components of the system being simulated. Messengers offer a natural formalism to satisfy the requirements of such simulations, since the common agreed-upon basis of understanding is not an encoding of facts but a complete language, which is universally understandable at any node containing the basic interpreter. Hence Messengers allow the generation of arbitrary new behaviors of simulated objects without having to change anything in the compiled node programs. This allows the user to experiment with a given situation in ways that may not have been anticipated in advance. Another strength of Messengers is their ability to utilize a computational network without any knowledge of its topology and thus may be used as a high-level network control language or for the composition and coordination of program ensambles composed dynamically from individual compiled functions distributed throughout the network.

Selected Works

Lubomir Bic, "Distributed Computing using Autonomous Objects," *5th IEEE CS Workshop on Future Trends of Distributed Computing Systems*, Cheju Island, Korea, August 1995.

"Intra- and Inter-Object Coordination with MESSENGERS," with M. Fukuda and M. Dillencourt, *First Int'l Conference on Coordination Models and Languages* (COORDINATION'96), Cesena, Italy, April 1996.

"Distributed Computing Using Autonomous Objects," with M. Fukuda and M. Dillencourt *IEEE COMPUTER*, August, 1996.

"Performance of the MESSENGERS Autonomous-Objects-Based System," with M. Fukuda, M. Dillencourt. *First Int'l Conf. on Worldwide Computing and Its Applications* (WWCA97), Tsukuba, Japan, March 1997.

"Messages versus Messengers in Distributed Programming," with M. Fukuda, M. Dillencourt, F. Merchant. *Int'l Conf. on Distributed Computing Systems* (ICDCS-97), Baltimore, MD, May 1997.

Rina Dechter

Professor
dechter@ics.uci.edu
http://www.ics.uci.edu/~dechter/

Automated Reasoning in Artificial Intelligence

Prof. Dechter's research is focused on automated reasoning in Artificial Intelligence, particularly in the areas of search, constraint-based reasoning and reasoning under uncertainty.

Her ongoing focus is on constraint processing, which emerges as a unifying theme that cuts across many traditional areas in Artificial Intelligence. A variety of techniques have been developed for processing different kinds of constraint expressions, and are being applied to diverse tasks such as vision, design, diagnosis, truth maintenance, scheduling, spatio-temporal reasoning, logic programming, and user interface. Many of these methods were incorporated into constraint programming languages constraints which enhance practical applications substantially.

Since most reasoning tasks are computationally intractable, the primary aim of Prof. Dechter's research is to devise methods through the understanding and exploitation of tractable reasoning tasks. Her previous works on greedy problems, the mechanical generation of heuristics, the identification of tractable constraint models via topological decompositions, and the establishment of boundaries of local computations, have been driven by this principal concern. Dechter analyzes algorithms both analytically and empirically using real life applications such as scheduling, planning, and diagnosis.

Dr. Dechter's current focus is on extending the constraint model to new areas of reasoning, especially to reasoning under uncertainty, through an algorithmic framework called bucket elimination. This framework unifies dynamic programming for combinatorial optimization with algorithms for theorem proving, logic programs, temporal reasoning, probabilistic reasoning and planning under uncertainty. Within this framework she develops efficient (exact and approximate) reasoning algorithms guided by the domain's properties and applies those to areas such as medical diagnosis, probabilistic decoding.

Selected Works

R. Dechter and J. Pearl, "Tree-clustering schemes for constraint-processing."Artificial Intelligence, Vol. 38(3), April 1989, pp. 353-366.

R. Dechter. "Enhancement schemes for constraint processing: Backjumping, learning and cutset decomposition." Artificial Intelligence, Vol. 41(3), 1990, pp. 273-312.

E. Schwalb and R. Dechter, "Processing Disjunctions in Temporal constraint networks." Artificial Intelligence, in press, 1997.

R. Dechter and P. van Beek, "Local and global relational consistency" Journal of Theoretical Computer Science, Vol 173(1), pp. 283-308, 1997.

R. Dechter, "Bucket elimination: a unifying framework for probabilistic reasoning" In Uncertainty in Artificial Intelligence (UAI96), 1996, pp. 211-219.

Michael B. Dillencourt

Associate Professor
dillenco@ics.uci.edu
http://www.ics.uci.edu/~dillenco/

Design and Analysis of Algorithms and Data Structures, Computational Geometry, Graph Theory, Distributed Computing

Dr. Dillencourt's research interests are in the areas of computational geometry, graph theory, and distributed computing. Within the area of computational geometry, Prof. Dillencourt has investigated the structure of Delaunay triangulations and the closely related family of inscribable polyhedra. He has also developed efficient algorithms for component-labeling and boundary extraction in binary images that are applicable to a variety of image-representation schemes.

Dr. Dillencourt and his students are developing an environment for doing computer-assisted research in graph theory and combinatorial geometry. They have developed GraphTool, a tool to support the development and analysis of algorithms in graph theory and computational geometry, and they have developed a large database of polyhedra.

In collaboration with Dr. Lubomir Bic and graduate students, Dr. Dillencourt has developed MESSENGERS, a distributed system based on the paradigm of autonomous objects. This system is being used as the basis for a distributed simulation environment for toxicologists, currently being jointly developed with collaborators from the Department of Community and Environmental Medicine.

Selected Works

M. B. Dillencourt and H. Samet, "Using Topological Sweep To Extract the Boundaries of Regions in Maps Represented by Region Quadtrees," *Algorithmica*, 15(1), 199--More6, pp. 82-102.

M. B. Dillencourt, "Polyhedra of small order and their Hamiltonian properties," *Journal of Combinatorial Theory Series B,* 66(1), 1996, pp. 87-122.

M. B. Dillencourt, "Finding Hamiltonian Cycles in Delaunay Triangulations Is NP-Complete," *Discrete Applied Mathematics*, 64(3), 1996, pp. 207-217.

L. F. Bic, M. Fukuda, and M. B. Dillencourt, "Distributed Computing using Autonomous Objects," *IEEE Computer*, 29(8), 1996, pp. 55-61.

M. Fukuda, L. F. Bic, M. B. Dillencourt, and F. Merchant, "Messages versus Messengers in Distributed Programming," *17th International Conference on Distributed Computing Systems (ICDCS'97)*, Baltimore, Maryland, May, 1997, 347-354.

T. K. Dey, M. B. Dillencourt, S. K. Ghosh, and J. M. Cahill, "Triangulating with High Connectivity," *Computational Geometry: Theory & Applications*, 8(1) June 1997, 39-56.

M. Fukuda, L. F. Bic, M. B. Dillencourt, and F. Merchant, "Distributed Coordination with MESSENGERS," to appear in *Science of Computer Programming.*

Nikil D. Dutt

Professor
dutt@ics.uci.edu
http://www.ics.uci.edu/~dutt/

Embedded Systems Design Automation: Specification, Representation, and Synthesis

The convergence of computers, communications and multimedia into electronic systems embedded within everyday consumer products has driven the need for design tools that bring new products into the market under increasingly shorter time frames. Given the complexity of this design process, a system-level designer needs early exploration tools that allow evaluation of different hardware and software design alternatives, before committing to a specific design path. Dr. Dutt's research interests focus on the techniques and tools for the automation of this embedded electronic system design process, starting from abstract behavioral specifications. He currently has research projects on design exploration for memory-intensive embedded systems, system-level technology mapping, memory issues for synthesis and compilation, fine-grain compiler techniques for application-specific architectures, and high-level synthesis.

In the past, Dr. Dutt has studied the use of hardware description languages (HDLs) for design specification in the context of synthesis, simulation and verification. He received two consecutive Best Paper Awards at the International Symposium on Computer Hardware Description Languages (*CHDL89* and *CHDL91*).

Selected Works

"High Level Synthesis: Introduction to Chip and System Design," With D. Gajski, A. Wu and S. Lin, *Kluwer Academic Publishers*, 1992.

"Optimal Register Assignment to Loops for Embedded Code Generation," with D. J. Kolson, A. Nicolau, and K. Kennedy, *ACM Transactions on Design Automation of Electronic Systems*, Vol. 1, No. 2, April 1996.

"High-Level Library Mapping for Arithmetic Components" With P. Jha, *IEEE Transactions on VLSI Systems*, Vol. 4, No. 2, June 1996.

"A Unified Lower Bound Estimation Technique for High-Level Synthesis," With S. Y. Ohm and F. J. Kurdahi, *IEEE Transactions on*

Computer-Aided Design of Integrated Circuits and Systems, Vol. 16, No. 5, May 1997.

"Architectural Exploration and Optimization of Local Memory in Embedded Systems," With P. R. Panda and A. Nicolau, *Proceedings of the 1997 International Symposium on System Synthesis (ISSS'97)*, September 1997.

"Data Organization for Improved Performance in Embedded Processor Applications," With P. R. Panda and A. Nicolau, *ACM Transactions on Design Automation of Electronic Systems*, Volume 2, Number 4, October, 1997.

"Exploiting Off-Chip Memory Access Modes in High-Level Synthesis," With P. R. Panda and A. Nicolau, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 17, No. 2, pp. 96-109, February, 1998.

"Low Power Memory Mapping through Reducing Address Bus Activity," With P.R. Panda, *IEEE Transactions on VLSI Systems*, Accepted for publication, to appear.

"On-Chip vs. Off-Chip Memory: The Data Partitioning Problem in Embedded Processor-based Systems," With P. R. Panda and A. Nicolau, *ACM Transactions on Design Automation of Electronic Systems*, Accepted for publication, to appear.

David Eppstein
Professor
eppstein@ics.uci.edu
http://www.ics.uci.edu/~eppstein/

Design and Analysis of Algorithms

Prof. Eppstein's research interests are in the design of algorithms and the analysis of related combinatorial structures particularly focusing on the areas of computational geometry and graph theory.

Major topics in Eppstein's research include:

Computational molecular biology: How can we design efficient algorithms for tasks such as finding optimal alignments of biological sequences, listing multiple suboptimal alignments, or using sequence alignment information to reconstruct evolutionary trees? Computational statistics: how can we efficiently implement methods for basic statistical tasks such as hierarchical clustering? Which few outliers, if removed from a dataset, would most change the weighted average of the remaining points? Can we fit data robustly, so that even large sets of outliers can not greatly decrease the quality of fit?

Mesh generation: If we are given some domain with a complex shape (such as the air around an airplane wing), how can we best divide it up into simple elements (tetrahedra or cuboids) to make the problem suitable for scientific computation? Eppstein has investigated a number of methods for these problems including quadtrees, disk packing, local improvement methods, and generalized linear programming, with an emphasis on robustness: can we guarantee that our methods will produce a high-quality mesh no matter what input they are given?

Dynamic minimum spanning trees: If we are given a changing set of objects, how can we maintain a network that connects them all together and is as short as possible? The objects may be vertices in a graph or points in a geometric space; they may be changing continuously or by discrete insertions and deletions. Data structures invented by Eppstein for this problem have also found many other applications including clustering, collision detection, and drawing offset curves.

Selected Works:

D. Eppstein. "Finding the k shortest paths." *35th IEEE Symp. Foundations of Comp. Sci.* (1994) 154-165.

M. Bern, D. Eppstein, and J. Gilbert. "Provably Good Mesh Generation." *J. Comp. and Sys. Sci.* 48 (1994) 384-409.

D. Eppstein. "Fast hierarchical clustering and other applications of dynamic closest pairs." *9th ACM Symp. Discrete Algorithms* (1998) 619-628.

D. Eppstein and J. Erickson. "Raising roofs, crashing cycles, and playing pool: applications of a data structure for finding pairwise interactions." *12th ACM Symp. Comp. Geom.* (1998) 58-67.

P. K. Agarwal, D. Eppstein, L. J. Guibas, and M. R. Henzinger. "Parametric and kinetic minimum spanning trees." *39th IEEE Symp. Foundations of Comp. Sci.* (1998) to appear.

Michael Franz

Assistant Professor
franz@ics.uci.edu
http://www.ics.uci.edu/~franz/

Programming Languages and Their Implementation. Extensible, Component-Based Software Systems

Dr. Franz's research marries the work of the traditionally separate compiler and operating system communities. His group has been building an extensible system that is founded on dynamic code-generation in conjunction with a platform-independent software distribution format called "Slim Binaries".

Unlike other intermediate representations designed for software portability, such as p-code or Java byte-codes, Slim Binaries are based on adaptive compression of syntax trees, and not on a virtual-machine. The high semantic level of a tree-based representation makes on-the-fly code generation and optimization particularly efficient, while its extreme compactness is useful when software is transferred over networks. Hence, slim binaries are well suited for realizing portable software components, such as downloadable content-specific editor applets for multimedia documents on the World Wide Web, or "intelligent agents" that move from machine to machine and perform computing tasks locally.

A related research project is profile-driven run-time code generation. In Franz's system implementation, a background process that executes only during otherwise idle processor cycles perpetually recompiles parts of the already running system, guided by real-time profiling data. This re-optimization encompasses not only user programs, but also the system libraries, eventually combining an application program and its dynamically loaded extensions and libraries into a fully cross-optimized code image. Because of the feed-back loop, and because it is able to cross-optimize application code with library code, dynamic code generation can produce better object code than static compilation.

Besides creating enabling technologies for implementing executable document-content and autonomous agents, Dr. Franz is investigating novel means of putting these technologies to work. His research encompasses the definition of software architectures, the creation of programming languages and tools for extensible computing, and the examination of safety issues that arise in the context of mobile code.

Selected Works

M. Franz, "Open Standards Beyond Java: On the Future of Mobile Code for the Intrnet." *Journal of Universal Computer Science*, 4:5, 521-532; May 1998.

M. Franz, "On the Architecture of Software Component Systems." in R. Nigel Horspool (Ed.), Systems Implementation 2000, Chapman & Hall, ISBN 0-412-83530-4, 207-220; February 1998.

M. Franz and T. Kistler, "Slim Binaries" *Communications of the ACM*, 40:12, 87-94; December 1997.

M. Franz, "Run-Time Code Generation as a Central System Service." in *The Sixth Workshop on Hot Topics in Operating Systems (HotOS-VI)*, IEEE Computer Society Press, ISBN 0-8186-7834-8, 112-117; May 1997.

M. Franz, "The Programming Language Lagoona: A Fresh Look at Object-Orientation." *Software - Concepts and Tools*, 18:1, 14-26; March 1997.

M. Franz, "Dynamic Linking of Software Components." *IEEE Computer*, 30:3, 74-81; March 1997.

Daniel D. Gajski

Professor
gajski@ics.uci.edu
http://www.ics.uci.edu/~gajski/

Design Science, Design Synthesis, Computer-Aided Design, Parallel Algorithms and Architectures

Professor Gajski is working on many aspects of Design Science, studying design process from specification to manufacturing to business models. He has developed, with his students, new methodologies for design process, and techniques for modeling of computer based systems. He is also working on specification languages and algorithms for design partitioning, estimation and synthesis of software and hardware.

Professor Gajski has developed many CAD tools for design of embedded computer systems. These tools take a high level description, such as an executable specification, algorithm or instruction set, and generate a design that will execute the given specification, algorithm or the instruction set. Since the design task is very complex, vague, and ill-structured, various techniques must be used to minimize the search through enormous design space. He has developed, with his students, algorithms and synthesis tools for VLSI layout, logic design, microarchitecture and computer systems including software/hardware co-design.

Daniel Gajski is also working on modeling design manufacturing and business processes. His group developed a methodology for 100 hour design process which improves design productivity by 2 orders of magnitude. He is also working on understanding of human interaction with the design process. He has published several books on the subject and an undergraduate textbook on principles of design.

Selected Works

D.D. Gajski, N.D. Dutt, Allen C-H Wu, Steve Y-L Lin, *High-Level Synthesis: Introduction to Chip and System Design*, Kluwer Academic Publishers, Boston, MA, 1992, 359 pages.

D.D. Gajski, F. Vahid, S. Narayan, J. Gong, *Specification and Design of Embedded Systems*, Prentice Hall, Englewood Cliffs, NJ, 1994, 450 pages.

D.D. Gajski, *Principles of Digital Design*, Prentice Hall, Upper Saddle River, NJ, 1997, 450 pages.

D.D. Gajski, J. Zhu, R. Doemer, "Essential Issues in Codesign," in *Introduction to Codesign: Principles and Applications* (Wolf, Soundstrap, eds.), Kluwer, Boston, MA, 1997.

H.-P. Juan, D.D. Gajski, S. Bakshi, "Clock Optimization for High-Performance Pipelined Design," European Design Automation Conference, Geneva, Switzerland, September 16-20, 1996, pp. 370-375. (Best paper award).

D.D. Gajski, S. Narayan, L. Ramachandran, F. Vahid, and P. Fung, "System Design Methodologies: Aiming at the 100 Hour Design Cycle," IEEE Trans. on VLSI Systems, Vol. 4, No. 1, March, 1996, pp. 70-82.

J. Gong, D.D. Gajski, S. Bakshi, "Model Refinement for Hardware-Software Codesign", ACM Transactions on DA of Electronic Systems, Vol. 2, No. 1, January 1997, pp. 22--41.

J. Zhu, P. Argrawal, D.D. Gajski, "RT Level Power Analysis," *Proc. ASP-DAC*, Chiba, Japan, January 28--31, 1997, pp. 517--522.

Richard Granger
Professor
granger@ics.uci.edu
http://www.ics.uci.edu/~granger/

Computational Neuroscience, Neural Modeling, Learning and Memory

The aim of our research is to understand how mammalian forebrain encodes, organizes, and retrieves information from the environment and then utilizes this information to perform goal-directed behavior.

Our studies focus on synaptic long-term potentiation (LTP), the biological change in the strength of cell-to-cell connections in the brain that underlies long-term memory. The program involves studies at several levels of brain organization, from the cellular machinery underlying synaptic LTP to the modeling of interactions among multiple forebrain regions.

Two current lines of research are i) the testing of behavioral concomitants of LTP, and ii) the computational formalization of brain rules for learning and memory. These projects are briefly described here, with a list of selected relevant publications.

A novel class of drugs that facilitate the induction of LTP in the brain have been shown to improve the encoding of memory (Granger et al., 1993; see http://www.ics.uci.edu/~granger for further references), and to offset age-associated memory impairments (Granger et al., 1996). Initial human tests of these drugs have been completed (Lynch et al., 1997, Ingvar et al., 1997) and further tests are ongoing.

Using models of specific cortical circuits, LTP was found to result in unusual systems of memory organization with remarkable capacity (Ambros-Ingerson et al., 1990; Anton et al., 1991; Gluck and Granger, 1993; Granger et al., 1994); these findings led to the derivation of novel brain-based algorithms for processing complex real-world perceptual information. These algorithms were shown to recognize complex signals (sonar) to a degree that matched or exceeded the capabilities of more conventional approaches, in tests carried out by government sponsoring agencies (Kowtha et al., 1994). These networks are now installed and used at U.S. Navy laboratories, and are under commercial development for signal processing uses in a number of applications domains.

Selected Works

Granger, R. Whitson, J., Larson, J., & Lynch, G. (1994). Non-Hebbian properties of LTP enable high-capacity encoding of temporal sequences. *Proc. Nat'l Acad. Sci*, 91: 10104-10108.

Kilborn, K., Granger, R., and Lynch, G. (1996). Effects of LTP on response selectivity of simulated cortical neurons. *J. Cog. Neurosci.*, 8:338-353.

Granger, R., Davis, M., Moskowitz, B., Kessler, M., and Lynch, G. (1996). Facilitation of glutamate receptors reverses an age-associated memory impairment in rats. *Synapse*, 22: 332-337.

Lynch, G., Granger, R., Davis, M., Ambros-Ingerson, J., Kessler, M., Schehr, R. (1997). Evidence that a positive modulator of glutamate receptors improves recall in elderly human subjects. *Experimental Neurol.*, 144: 1-4.

Granger, R., Wiebe, S., Taketani, M., Ambros-Ingerson, J., & Lynch, G. (1997). Distinct memory circuits comprising the hippocampal region. *Hippocampus*, 6: 567-578.

Ingvar, M., Ambros-Ingerson, J., Davis, M., Granger, R., Kessler, M., Rogers, G, Schehr, R., & Lynch., G. (1997). Enhancement by an ampakine of memory encoding in humans. *Experimental Neurol.*, 146: 553-559.

Barrows, G., Granger, R., and Stenger, D. (1997). Evaluation of a biological derived hierarchical clusterer when applied to emitter identification. *Proc.Computational and Neural Systems Conf.*, Boston: Kluwer.

Davis, M., Moskovitz, B., Nguyen, M., Lynch, G., Rogers, G., Granger, R. (1997). A profile of the behavioral changes produced by facilitation of AMPA-type glutamate receptors. *Psychopharmacology*, 133: 161-167.

Davis, M. Ambros-Ingerson, J., Granger, R., Wu, J., Zabaneh, R., Abdelnaby, M, and Lynch, G. (1998). Triazolam impairs delayed recall but not acquisition of various everyday memory tasks in humans. *Experimental Neurol*., 148: 124-134.

Jonathan Grudin

Professor
grudin@ics.uci.edu
http://www.ics.uci.edu/~grudin/

Computer-Supported Cooperative Work (CSCW), Human-Computer Interaction (HCI)

Dr. Grudin's research currently focuses on the design, adoption and use of technologies that support group and organizational activity. He and his students are looking at the use of a number of such technologies: group calendars used in organizations, telecommuting practices linking office and home, telemedicine, and TeleFly, a distributed "virtual world" technology used by distributed design teams.

Dr. Grudin has been active in many aspects of the rapidly-growing computer science specialization of HCI. He has examined specific interface design features, such as names and abbreviations, analyzed theoretical constructs, such as interface consistency, and applied new research and development methods. He has also analyzed the historical and organizational influences that shape the way that interfaces are developed and used. He is Editor-In-Chief of the ACM Transactions on Computer-Human Interaction, and Co-Chair of the CSCW'98 Conference.

Selected Works

R. M. Baecker, J. Grudin, W. A. S. Buxton, and S. Greenberg, (Eds.), "Readings in Human-Computer Interaction: Toward the Year 2000," San Mateo, CA Morgan Kaufmann, 1995.

J. Grudin and S.E. Poltrock, "Computer-supported cooperative work and groupware." In M. Zelkowitz (Ed.), "Advances In Computers." Orlando, FL. Academic Press, 1997.

J. Grudin and M. L. Markus, "Organizational issues in development and implementaion of interactive systems." In M. G. Helander, T. K. Landauer, and P. Prabhu (Eds.), "Handbook of Human-Computer Interaction (Second Edition)." Amsterdam, The Netherlands: Elsevier Science, 1997.

J. Grudin, "Groupware and Social Dynamics: Eight Challenges for Developers." *Communication of the ACM*, 37, 1, 92-105, 1994.

Rajesh Gupta

Associate Professor
rgupta@ics.uci.edu
http://www.ics.uci.edu/~rgupta/

Architecture and CAD for Embedded Systems

Our goal is to develop techniques and tools for efficient exploration of architectural alternatives in building high performance single-chip VLSI systems. Since these designs are often severely constrained by power, cost considerations and bounds on the timing performance. Therefore, constraint analysis and system optimization techniques play a key role in system design. This research is organized into three components: language-level system optimizations, timing performance analysis and software synthesis for embedded systems.

Our current research focus in system analysis is on efficient techniques for deterministic analysis of latency and throughput-type constraints. Recently we developed a close bound on the process execution rates in a network of interacting processes, and used it to build a tool that allows the system designer to identify constraint violations and, if needed, debug these violations by using pipeline transformation techniques. In the area of system optimizations, we have developed a tabular model of system behavior that is naturally suited for optimizations using Don't Cares (DCs) either derived or externally specified. We have developed DC-based optimizations to carry out presynthesis optimizations for hardware and software targets.

Finally, the software content in embedded systems represents an area of rapid growth and enormous challenges to generate correct and performance constrained embeddable code. We have developed a notion of software serializability as a correctness condition and developed tests to guarantee serializability. Our current focus is on automatic synthesis of application-specific retargetable code that is derived directly from system models. Other software optimizations will explore use of prefetching and pipelining to hide effects of memory latency and meet the execution rate constraints.

The three components of this project, namely: system analysis, optimization, and software synthesis, are being developed in conjunction with a CAD framework for system design that supports system design activity through interface modeling and synthesis.

This research is supported by grants from AT&T Foundation, National Science Foundation MIPS, NSF-HPCC and an NSF CAREER Award.

Selected Works

G. De Micheli and R. Gupta, "Hardware/Software Co-Design," *Proceedings of IEEE*, March 1997.

X. Zhang, A. Dasdan, M. Schulz, R. Gupta, A. Chien, "Architecture Adaptation for Application-Specific Locality Optimization," *International Conference on Computer Design*, Oct 1997.

J. Li and R. Gupta, "HDL Optimizations using Timed Decision Tables," *33rd Design Automation Conference*, Las Vegas, June 1996.

R.Gupta and G. De Micheli, "Specification and Analysis of Timing Constraints for Embedded Systems," *IEEE Transactions on CAD/CAS*, March 1997.

A.Dasdan, A. Mathur, R. Gupta, "RATAN: A Framework for Rate Analysis," *European Design and Test Conference*, March 1997.

Daniel Hirschberg

Professor
dan@ics.uci.edu
http://www.ics.uci.edu/~dan/

Design and Complexity Analysis of Algorithms and Data Structures

Dr. Hirschberg's research has centered in the fields of data structures and concrete computational complexity. Most of his work has concentrated on five areas: common subsequence problems, algorithms using a parallel processor, design of efficient data structures, breakpoint problems and data compression.

Several of the algorithms developed by Dr. Hirschberg formed the basis for many or all subsequent algorithms for those problems. These seminal algorithms include solving the longest common subsequence problem using only linear space, solving the connected components problem on a parallel computer in polylogarithmic time, and asynchronously electing a leader on a ring of n processors using O(n log n) messages.

Dr. Hirschberg's research in the area of data compression has resulted in the development of a simple O(nL)-time algorithm that determines an optimum prefix-free binary code for a weighted alphabet of size n, under the restriction that code strings cannot be longer than L. He has also focused on the case in which an encoder transmits a coded text file to a decoder that has constrained memory resources and developed several algorithms that provide compression performance comparable to state-of-the-art techniques while using significantly less memory.

Selected Works

D. A. Lelewer and D. S. Hirschberg. "Data compression," Computing Surveys 19, 3 (1987) 261-297. Reprinted in Japanese BIT special issue in Computer Science (1989) 165-195.

D. S. Hirschberg and D. A. Leweler. "Efficient decoding of prefix codes," *Communications of the ACM* 33, 4 (1990) 449-459.

L. L. Larmore and D. S. Hirschberg. "A fast algorithm for optimal length-limited codes," *Journal ACM* 37, 3 (1990) 464-473.

D. S. Hirschberg and L. M. Stauffer. "Dictionary compression on the PRAM," *Parallel Processing Letters* 7, 3 (1997) 297-308.

D. S. Hirschberg. "Serial computations of Levenshtein distances," in Pattern Matching Algorithms, A. Apostolico and Z. Galil (Eds.), Oxford University Press, 1997, pp. 123-141.

Sandra Irani

Associate Professor
irani@ics.uci.edu
http://www.ics.uci.edu/~irani/

Design and Analysis of Algorithms

Dr. Irani's principal research interests are in the design and analysis of algorithms. In particular, most of her research has been in the area of on-line algorithms. The term "on-line" refers to a relationship between the input stream and the output stream of an algorithm. Namely, the input is presented to the algorithm incrementally and the algorithm must produce part of the output with limited information about the problem. The question is then what kind of solution quality can one hope to obtain? Many problems that arise in Computer Science are on-line problems. Among these are processor scheduling, data structure management, robot motion planning, and resource allocation. Dr. Irani has worked on problems in on-line data structures, graph coloring, scheduling and memory management. Her research uses a technique called Competitive Analysis which is used to analyze on-line algorithms by comparing the performance of an on-line algorithm to the optimal off-line algorithm. This technique has been useful in that it gives the ability to make strong theoretical statements about the performance of algorithms without making probabilistic assumptions about the input. Some of her recent work has focused on the development of online replacement policies for web caches. These algorithms have been evaluated analytically using competitive analysis. In addition, she has performed empricial studies of these policies using trace data from web caches in a variety of setting.

Dr. Irani has also worked recently on point matching problems with applications to Computer Vision and Document Image Processing.

Selected Works

Scheduling with conflicts, and applications to traffic signal control. With Vitus Leung. In the *Proceedings for the 7th Symposium on Discrete Algorithms*, 1996, pages 85-94.

Probabilistic Analysis for Scheduling with Conflicts. With Vitus Leung. In the *Proceedings for the 8th Symposium on Discrete Algorithms*, 1997, pages 286 - 296.

Page replacement with multi-size pages and applications to web caching. In the *Proceedings for the 29th Symposium on the Theory of Computing*, 1997, pages 701-710.

Cost-Aware WWW Proxy Caching Algorithms. With Pei Cao. In the *Proceedings of the USENIX Syposium on Internet Technologies and Systems*, 1997, pages 193-206.

Combinatorial and experimental results for randomized point matching algorithms, With Prabhakar Raghavan. In the *Proceedings for the 12th ACM Symposium on Compuational Geometry*, 1996, pages 68-77.

Dennis Kibler

Professor
kibler@ics.uci.edu
http://www.ics.uci.edu/~kibler/

Machine Learning

Dr. Kibler's research focuses on computational models of learning. More specifically his research is on concept descriptions and on improving problem solving. Within each area his research exemplifies the value of a scientific experimental methodology.

His past work on learning problem solving concentrated on learning control rules and learning selective macros. His work on concept learning has focused on extending or improving very simple representational schemes. One approach he considered was simply remembering instances, an arena now called Instance Based Learning (IBL). The work on Instance Based Learning has developed into two new areas: Prototype Learning, where one learns ideal instance and Constructive Induction, where one learns better features for describing concepts.

His latest research focuses on extending the power of Linear Threshold Units and on applying learning algorithms to biosequences. Standard machine learning algorithms do not directly apply to biosequences, since positional information is important in these domains. The particular biological question that Kibler's research focusses on now is determining the combination of motifs that are responsible for the control gene expression. This problem requires the integration and extension of several Machine Learning methods, including clustering, constructive induction, characterization and discrimination.

Selected Works

"Minimum Generalization via Reflection: A Fast Linear Threshold Learner", with Steven Hampson, submitted to *Machine Learning*, 1997.

"A Wrapper Approach to Constructive Induction", with Yuh-Jyh Hu, *AAAI-96*, 806-812.

"Learning Prototypical Concept Descriptions", with Piew Datta, *Machine Learning Conference*, 1995.

"Learning Steppingstones for Problem Solving", with David Ruby. *International Journal of Pattern Recognition and Artificial Intelligence*, vol 7 no. 3. 1993.

"Instance-Based Learning Algorithms", with David Aha and Marc Albert, *Machine Learning*, 37-66, 1991.

John Leslie King

Professor
king@ics.uci.edu
http://www.ics.uci.edu/~king/

High-Level Requirements for Information Systems Design, Management and Economics of Computing, Social Impacts of Computing

Dr. King's current research focuses on development of improved methods for establishing high-level requirements for information systems design and development. This research is informed by study of the ways in which organizational and institutional forces shape how information technology is developed (including what gets developed) and how the technologies that do get developed change the course of organizational and institutional behavior. The research draws on the fields of economics and other social sciences as well as the engineering sciences. The goal of the work is to improve the design of information technologies for both organizational and institutional usability through better articulating the processes of requirements analysis, specification, and prototype creation. The work also informs policy and strategy development at the firm, sectoral, and institutional levels. Current projects include a study of the evolution of systems requirements in intermodal transport and logistics; examination of California criminal courts as a venue of computer-supported cooperative work implementation; and study of the technical and institutional co-evolution of standards and technical infrastructure in global land-line and cellular telephony; and the requirements for the creation of digital library capabilities in traditional library settings. Dr. King is Professor in both Information and Computer Science the Graduate School of Management at the University of California, Irvine, and is Editor-and-Chief of The Information Systems Research, a leading academic journal published by INFORMS. He holds a Ph.D. in Administration from UCI.

Selected Works

J. Gelfand and J.L. King. "Shifting the Locus of Control: Legitimation of New Information Products in the Digital Age" In the *Proceedings of the International Conference on the Social Impact of Information Technology*, Forthcoming, 1998.

A. Andeen and J.L. King. "Addressing and the Future of Communications Competition." *Information Infrastructure and Policy*, Forthcoming, 1998.

J.L. King and T. Morgan. "Network Managment." In Dorf, R., Encyclopedia of Technology Management. CRC Press, Forthcoming, 1998.

K.H. Jeong and J.L. King. "Korea's National Information Infrastructure: Visions and Issues." In Kahin, B. and T. Leshan, Global Information Infrastructure. MIT Press, 1997.

J.L. King, R. Grinter and J. Pickering. "The Rise and Fall of Netville: Institution, Infrastructure, And The Saga Of A Cybserspace Construction Boomtown In The Great Divide." In Kiesler, S. (ed.) The Culture of the Internet. Ehrlbaum Publishers, 1997.

Alt, R., Forster, P.W., and J.L. King. "The Great Reversal: Information and Transportation Infrastructure in the Intermodal Vision." In *Conference Proceedings 12: National Conference on Developing a Research Framework for Intermodal Transportation.* TRB, National Research Council, Washington, D.C., 1997.

C.S. Turner, D. Richardson, and J.L. King. "Legal Sufficiency of Testing Processes." *Proceedings of the 15th International Conference on Computer Safety*, *Reliability*, *and Security*, Vienna, October 23-35, 1996.

Richard H. Lathrop

Associate Professor
rickl@ics.uci.edu
http://www.ics.uci.edu/~rickl/

Artificial Intelligence, Computational Molecular Biology

Research in protein structure prediction asks the question: Given a protein sequence, what is its three-dimensional structure? This has been called "the second half of the genetic code," and is a Grand Challenge problem for computer science.

Research in HIV drug resistant mutants seeks to improve the clinical treatment of individual HIV patients, and to understand the basis of HIV drug resistance in the virus. CTSHIV is a rule-based AI computer system developed at UCI and currently in use in human clinical trials on HIV patients. (Collaborative work with Profs. Pazzani, See, and Tilles at UCI.)

Research in Huntington's disease ultimately hopes to develop a treatment for those afflicted with this disease. Computational initiatives are aimed at understanding the structural basis of huntingtin aggregates, and guiding a drug discovery program. (Collaborative work with Profs. Thompson, Tobias, Nowick, and Marsh at UCI.)

Research in genetic regulation seeks to uncover the basis by which the genome exerts dynamic control over life processes. An Affymetrix genechip machine is used to help discover patterns and regularities in the genetic control regions. (Collaborative work with Profs. Kibler, Sandmeyer, and McLaughlin at UCI.)
I encourage those with other interests or ideas to discuss other topics with me.

Selected Works

"Knowledge-based Avoidance of Drug-Resistant HIV Mutants," Lathrop, R.H., Steffen, N.R., Raphael, M., Deeds-Rubin, S., Pazzani, M.J., Cimoch, P.J., See, D.M., Tilles, J.G., *Proc. Innovative Applications of Artificial Intelligence Conf.*, Madison, WI, USA, July 27--29, 1998 (Innovative Application Award winner).

"Solving the Multiple-Instance Problem with Axis-Parallel Rectangles," Dietterich, T.G., Lathrop, R.H., Lozano-P'erez, T., *Artificial Intelligence*, 89:31-71, 1997..

"A Branch and Bound Search Algorithm for Finding the Global Optimum Protein Threading with Gapped Alignment and Empirical Pair Potentials," Lathrop, R.H., Smith, T.F., *J. Molecular Biolology*, 255:641--665, Feb., 1996 (cover article).

"The Protein Threading Problem With Sequence Amino Acid Interaction Preferences Is NP-Complete," Lathrop, R.H., *Protein Engineering*, 7(9):1059--1068, Sept., 1994.

Lathrop, R., Webster, T., Smith, R., Winston, P., Smith, T., "Integrating AI With Sequence Analysis," pp. 210--258 in *Artificial Intelligence and Molecular Biology*, ed. Hunter, L., AAAI Press, Menlo Park, 1993.

George S. Lueker

Professor
lueker@ics.uci.edu
http://www.ics.uci.edu/~lueker/

Algorithms and Data Structures, Probabilistic Analysis

George Lueker's interests are in the area of the design and analysis of algorithms for concrete problems. Dr. Lueker is especially interested in applications of probability to problems in computer science, in particular the analysis of the behavior of algorithms and optimization problems when some assumption is made about the probability distribution of inputs. For example, he has investigated the behavior of the optimum solution to the bin-packing problem and the partition problem. Also, with graduate student Mariko Molodowitch he developed a remarkably simple proof of the result that the expected behavior of double hashing is asymptotically equivalent to that of the theoretically ideal uniform hashing, even for large load factors. Recently he has examined the average-case behavior of on-line algorithms for the knapsack problem. He is presently especially interested in the behavior of the longest common subsequence of two random strings.

Selected Works

"Testing for the Consecutive Ones Property, Interval Graphs, and Graph Planarity Using PQ-Tree Algorithms," with K. S. Booth, *Journal of Computer and System Sciences* 13,3 (1976), pp. 335--379.

"Bin Packing Can Be Solved within 1+ε in Linear Time," with W. Fernandez de la Vega, *Combinatorica* 1,4, pp. 349-355, 1981.

"Probabilistic Analysis of Optimum Partitioning," with N. Karmarkar, R. M. Karp, and A. M. Odlyzko, *Journal of Applied Probability* 23,3, pp. 626-645, 1986.

"Probabilistic Analysis of Packing and Partitioning Algorithms" with Ed Coffman, Wiley Interscience Series in Discrete

Mathematics and Optimization, John Wiley & Sons, 192 pages, 1991.

"More Analysis of Double Hashing," with M. Molodowitch, *Combinatorica* 13,1, pp. 83-96, 1993.

"Average-Case Analysis of Off-Line and On-Line Knapsack Problems," *Proc. Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, San Francisco, CA, pp. 179-188, January 1995; to appear in a special issue of the Journal of Algorithms containing selected papers from this conference.

"Exponentially Small Bounds on the Expected Optimum of the Partition and Subset Sum Problems," *Random Structures and Algorithms* 12 (1998), pp. 51--62.

Sharad Mehrotra

Assistant Professor
sharad@ics.uci.edu
http://www.ics.uci.edu/~sharad

Database Management, Information Retrieval, Multimedia Systems, Distributed Systems, Transaction Management and Fault-tolerance

Sharad Mehrotra's primary research interests are in the area of multimedia information retrieval and database management, multidimensional data structures and indexing mechanisms, spatio-temporal databases, transaction management, and distributed information systems.

Currently he leads three research projects. The first is the *Multimedia Analysis and Retrieval Systems* (MARS) project whose goals are to provide seamless support for multimedia information as first class objects capable of being stored and retrieved in information systems based on their rich internal content. The MARS project is examining approaches to represent multimedia information content, content-based retrieval of multimedia, techniques to efficient indexing of multimedia information, and multimedia database management. The second project is *SpAtio-Temporal UnceRtaiNty management system* (SATURN) whose objective is to develop effective techniques to represent and retrieve spatio-temporal information in databases. Applications of SATURN include supporting reasoning and planning on spatio-temporally registered battlefield information. The third project is the *INCA-based Distributed Computing Environment* (IDCE) whose goals are to provide effective and efficient middleware support and a programming environment for emerging distributed applications like workflows, collaborative sessions, and agent-based computations. As part of the IDCE project, the author is examining approaches to model complex dynamic control and dataflow of applications, programmable interfaces to specify the processing, consistency and fault-tolerance requirements, and approaches to object migration and replication in such environments.

Selected Works

S. Mehrotra, R. Rastogi, Y. Breitbart, H. Korth, and A. Silberschatz. Ensuring Consistency in Multidatabases by Preserving Two-Level Serializability. To appear in *ACM Transactions on Database Systems*, 1998.

R. Rastogi, S. Mehrotra, Y. Breitbart, H. Korth, and A. Silberschatz. On Correctness of Non-serializable Executions. To appear in *Journal of Computer and Systems Science*, 1998.

M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, T. Huang. Supporting boolean Queries in MARS. To appear in *IEEE Transactions on Knowledge and Data Engineering*}, 1998.

Y. Rui, T. Huang, and S. Mehrotra. Table of Content Analysis in Videos. To appear in *ACM Journal of Multimedia Systems*, Special Issue on Video Library, 1998.

Y. Rui, T. Huang, M. Ortega and S. Mehrotra. Relevance Feedback Approach to Image Retrieval. To appear in *IEEE Transaction on Circuits and Systems*, 1998.

S. Mehrotra, K. Hu, and S. Kaplan. Dealing with partial failures in multiple processor primary-backup systems. *Proceedings of the Sixth ACM International Conference on Information and Knowledge Management*, Nov. 1997

K. Chakrabarti and S. Mehrotra. Granular locking approach to Phantom Protection over R-trees. *Proceedings of the IEEE Data Engineering Conference*, 1998.

Alexandru Nicolau

Professor
nicolau@ics.uci.edu
http://www.ics.uci.edu/~nicolau/

Parallel Computation, Architectures, Compilers, Programming Languages

Parallel programs, especially those with truly critical time (speed) requirements, are difficult to design. The process is extremely error-prone, tedious and time-consuming. The first goal of Dr. Nicolau's work is to design and implement a system of program transformations that support the semi-automatic (and eventually fully-automatic) exploitation of substantially all the parallelism available in a given program.

A second, and closely related, goal of Dr. Nicolau's work is to provide a tool for the rigorous study and development of parallelizing compilers. Furthermore, he is interested in precisely (formally) understanding and delimiting the power and limitations of such compilers. In particular, he studies the correctness and completeness (applicability) of parallelizing transformations, as well as comparing their relative power to expose/exploit parallelism. These issues are being examined within a unified formalism which will allow him to make meaningful comparisons between transformations. This should lead to the design of new, more powerful and/or more general transformations. By designing his environment around this formal model and integrating the results of the theoretical work into his environment, he will have the means of empirically corroborating his formal results, and determining the usefulness of his transformations in the context of real code. In turn, this should yield a prototype integrated environment which will satisfy his first goal.

A third goal of this project is to investigate the relative trade-offs between run-time and compile-time parallelism exploitation. In particular, through Dr. Nicolau's formal and experimental tools, he hopes to gain insight as to the proper balance between compile-time parallelization and run-time support, and the power of various architectural models in terms of their ability to exploit parallelism.

A final goal of Dr. Nicolau's work is to examine the relationship between parallel languages and parallelizing compilers. He believes that even when the ability to specify parallelism explicitly in the program at the user level is available, much can be done by the compiler to enhance the parallelism exposed, and to help avoid many errors that could otherwise be introduced by the user, and could be very time-consuming to identify without the system's help.

Selected Works

"Fine-Grain Parallelization and the Wavefront Method," with A. Aiken, Second Workshop on Languages and Compilers for Parallel Computing, August 1989.

"Tradeoffs between Pipeling and Multiple Functional Units in Fine-Grain Parallelism Exploitation," with M. Berternitz, *Fourth International Conference on Supercomputing*, Santa Clara, CA, May 1989.

"A Global Resource Constrained Parallelization Technique," with K. Ebcioglu, in *Proc. ACM Sigarch International Conference on Supercomputing*, Crete, Greece, June 1989.

"Interference Analysis Tools for Recursive Data Structures," with L. Hendren, in *Proc. ACM Sigarch International Conference on Supercomputing*, Crete, Greece, June 1989.

"Parallelizing Programs with Recursive Data Structures," with L. Hendren, in *Proc.*

International Conference on Parallel Processing, St. Charles, Illinois, August 1989.

Michael J. Pazzani

Professor and Chair

pazzani@ics.uci.edu

http://www.ics.uci.edu/~pazzani/

Machine Learning, Knowledge Discovery In Databases, Cognitive Science, Expert Systems

Dr. Pazzani's research focuses on machine learning, the discovery of patterns in existing data that may be used to make predictions about unseen data. Although most of his research involves the development and analysis of learning algorithms, a variety of applications have been developed including learning guidelines for the screening of dementia from clinical records, recommending treatment for HIV patients, learning and revising rules for troubleshooting a telephone network, creating profiles of user interests on the World Wide Web, predicting the effects of economic sanctions, and learning to translate Japanese verbs to English. There are two themes to his research:

The investigation and analysis of learning methods that make use of prior knowledge to guide the learning process. This has been shown to increase the accuracy of learned rules and increase the efficiency of learning algorithms. Most recently, Dr. Pazzani has been investigating how prior knowledge improves the understandability of learned rules.

Understanding the properties of problems that affect the accuracy of existing learning algorithms and the development of extensions to learning algorithms to improve their performance on certain classes of problems.

Pazzani's research program involves the development of new learning algorithms, the psychological investigation of human learners, the experimental and theoretical analysis of learning algorithms, and the application of knowledge discovery systems to a variety of problems. One application of this research, Syskill & Webert, an intelligent agent for finding interesting Web sites, is available for public use at www.ics.uci.edu/~pazzani/ Agents.html on the World Wide Web.

Selected Works

E. Keogh and M. Pazzani. "An Enhanced Representation of Time Series Which Allows Fast and Accurate Classification, Clustering and Relevance Feedback." *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining,* NY, NY. AAAI Press. 1998.

M. Pazzani, S. Mani & W. R. Shankle. "Comprehensible knowledge-discovery in databases." In M. G. Shafto and P. Langley (Ed.), *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, pp. 596-601. Lawrence Erlbaum. 1997.

M. Pazzani, C. Merz, P. Murphy, K. Ali, T. Hume and C. Brunk. "Reducing Misclassification Costs." *Proceedings of the 11th International Conference of Machine Learning*, New Brunswick. Morgan Kaufmann, pp. 217-225. 1994.

M. Pazzani and D. Billsus. "Learning and Revising User Profiles: The identification of interesting web sites." *Machine Learning 27*, pp. 313-331. 1997.

P. Domingos and M. Pazzani. "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss." *Machine Learning*, 1997.

M. Pazzani, D. See, E. Schroder, and J. Tilles. "Application of an Expert System in the Management of HIV-infected Patients." *Journal of Acquired Immune Deficiency Syndromes and Human Retrovirology*, 1997.

Wanda Pratt
Assistant Professor
pratt@ics.uci.edu
http://www.ics.uci.edu/~pratt

Medical informatics; Information access; Knowledge-based systems.

The focus of Dr. Pratt's research is on the development of knowledge-based methods that improve access to information. Critical problems in medicine drive much of her research. Both health-care workers and patients need fast and easy access to the vast amount of medical information found in journal articles, books, web pages, or patient records. The time that physicians and nurses can spend addressing an individual patient's problem is decreasing, yet the amount of information they need is increasing. Patients are starting to actively participate in many of their health care decisions; thus, they need access to medical information in a form that they can understand. These changes point to a need for new techniques that integrate information from multiple sources, display information in informative interfaces, and tailor information to individuals.

Dr. Pratt's recent research addresses one aspect of the problem of information overload: when users of computer-based search engines become overwhelmed by the large number of documents returned. This problem is exceptionally severe when people search the primary medical literature, to which over 31,000 journal citations are added each month. Many search tools address this problem by helping users to make their searches more specific. However, when dozens or hundreds of documents are relevant to their question, users need tools that help them to explore and to understand their search results, rather than ones that eliminate a portion of those results. Dr. Pratt created a system that allows users to issue general queries and helps them to learn about the documents returned from such a query and to decide what areas to explore further. It accomplishes this task by using knowledge of users' queries and a model of the domain terminology to organize the search results into categories that provide a query sensitive summary of topics discussed.

Selected Works

W. Pratt. Dynamic Organization of Search Results Using the UMLS. *Proceedings of the American Medical Informatics Association Fall Symposium (formerly SCAMC)*, Nashville, TN, p. 480-485, October 1997.

R. Fikes, A. Farquhar, & W. Pratt. Information Brokers: Gathering Information from Heterogeneous Information Sources. *Proceedings of the Ninth Florida Artificial Intelligence Research Symposium (FLAIRS-96)*, Key West, Florida, p. 192-197, 1996.

W. Pratt & I. Sim. Physician's Information Customizer (PIC): Using a Shareable User Model to Filter the Medical Literature. *Proceedings of the International Conference on Medical Informatics (MEDINFO'95)*, p. 1447-1452, July 1995.

David F. Redmiles
Assistant Professor
redmiles@ics.uci.edu
http://www.ics.uci.edu/~redmiles/

Human Computer Interaction, Software Engineering, Computer-Supported Cooperative Work

Dr. Redmiles' research combines the area of human-computer interaction and software engineering, focusing on the processes and technologies needed to develop and support useful and usable interactive software. The research conceptualizes evolutionary software development as a process of on-going communication. The content of the communication consists of information based on usability and requirements engineering. Methods such as protocol analysis, statistical testing, activity theory, scenarios, and cognitive walkthroughs are adapted. Tool design adheres to cognitive theories such as reflection in action, opportunistic design, and comprehension and problem solving.

One line of research is investigating the use of intelligent software agents both as a means of enhancing the communication model and reducing the cost of incorporating requirements and usability information. For example, agents can facilitate synchronous or asynchronous dialogs between end users and developers or automatically observe and collect data on usage patterns for prototypes. In general, the goal is to more tightly couple development to end users' situations.

Another area of research investigates software design environments that support high level design of systems through architecture definition languages including industrial standards for object-oriented specification (i.e., the UML -- Unified Modeling Language). The design environment sets up a kind of dialog between designers creating a system and software critics that react to their partial design specifications. This research emphasizes adherence to cognitive models of design and problem solving.

A third area of research involves two projects that examine the requirements of software engineers working in groups. The first project in this area is the study of software maintenance engineers at two large companies in Silicon Valley. A social theory called "activity theory" is applied to analyze the behavior of the software engineers and postulate future automated support for their work. The second project in this general area studies software engineers at an East Coast company using a software tool to support group awareness. The tool automatically captures and classifies organizational knowledge communicated through email and informs (keeps aware) workers about activity related to their current task and interests.

Previous work by Dr. Redmiles was in the area of software comprehension. This research studied the needs of software engineers attempting to reuse large software libraries. Tests with software engineers validated the use of multiple perspective explanations to enhance the use of examples in supporting reuse. This approach followed from theories and models of frame-based representation. The conclusions are applicable to human learning and information retrieval across large repositories (such as the World-Wide Web).

Selected Works

D. Hilbert, D. Redmiles. "Agents for Collecting Application Usage data Over the Internet," Proceedings of the Second International Conference on Autonomous Agents (Agents '98, Minneapolis/St. Paul, MN), ACM, May 10-13, 1998.

J. Robbins, D. Hilbert, D. Redmiles. "Extending Design Environments to Software Architecture Design," Automated Software Engineering, Vol. 5, No. 3, 1998.

J. Robbins, M. Medvidovic, D. Redmiles, D. Rosenblum. "Integrating Architecture Description Languages with a Standard Design Method," Proceedings of the Twentieth International Conference on Software Engineering (ICSE '98, Kyoto, Japan), IEEE Computer Society Press, April 19-25, 1998.

S. Shukla, B. Nardi, D. Redmiles. "Hit Squads & Bug Meisters: Discovering New Artifacts for the Design of Software Supporting Collaborative Work," Human Factors in Computing Systems CHI '98 Conference Summary (Los Angeles, CA), ACM April 1998.

M. Kantor, B. Zimmermann, D. Redmiles. "From Group Memory to Project Awareness Through Use of the Knowledge Depot," Proceedings of the 1997 California Software Symposium (Irvine, CA), UCI Irvine Research Unit in Software, Irvine, CA, November 7, 1997.

D. Redmiles, "Reducing the Variability of Programmers' Performance Through Explained Examples," Human Factors in Computing Systems, INTERCHI '93

Conference Proceedings (Amsterdam, The Netherlands), ACM, pp. 67-73. April 1993.

Debra J. Richardson

Associate Professor

djr@ics.uci.edu

http://www.ics.uci.edu/~djr/

Software Testing, Verification and Validation, Formal Specification Languages and Analysis Methods, Specification- and Architecture-based Testing

Debra J. Richardson is internationally recognized for pioneering the use of formal specifications to guide and evaluate testing and analysis. Professor Richardson's primary research is directed toward investigating and developing specification-based testing through the integration of formal specification and analysis methods with software testing. A particular focus of her research is the application of specification-based testing throughout the software lifecycle, starting with project planning and risk assessment, to requirements analysis, and through operational use. She is addressing an overall research goal of improving software testing processes by designing and implementing new and more powerful software testing technology, evaluating their effectiveness, and developing processes that integrate complementary testing technologies and guide activities within software engineering environments.

Professor Richardson's current projects include research on and development of a variety of software testing approaches based upon formal specification and analysis.

- **Specification-based test oracles** provide execution-time checking of test results against formal specifications of required behavior. Most testing R&D efforts focus on what to test and ignore techniques for verifying the correctness of execution behavior. Since testing criteria often dictate an overwhelming number of test data, leaving comparison to chance greatly limits the conclusions that can be drawn from the testing activity. Richardson is examining how test oracles can be derived from specifications.

- **Specification-based test criteria** describe test adequacy criteria based on formal specifications of required behavior. Most test criteria are based on the implementation and thus fail to test what is supposed to be done. Richardson is formalizing functional testing guidelines, extending structural techniques to be applicable with formal specification languages, and exploring how test classes developed from specifications can be refined through formal mappings and reused in later phases of development.

- **Regression test criteria** indicate the parts of a system that must be tested after a change is made. Richardson is developing regression test criteria based upon program dependence analysis, which identifies the syntactic relationships between program components that represent when a change in the semantics of one component may affect the execution behavior of another.

- **Architecture-based testing** enhances specification-based testing techniques to be applicable at the level of software architecture. Richardson is developing both architecture-based test criteria for integration testing as well as architecture-based test oracles for conformance testing.

- **Software testability measures** indicate the cost and effort involved in the testing activity. Most such measures are not applicable until the code has been developed. Richardson is developing and evaluating software testability measures that are applicable early in the lifecycle and thus facilitate software test planning.

- **Perpetual testing technology** supports seamless, perpetual analysis and testing of software through development, deployment and evolution. Whereas the dominant paradigm treats testing as a phase that succeeds development and precedes delivery, Richardson is building the foundation for treating analysis and testing as on-going activities to improve quality assurance without pause through several generations of product, in the development environment as well as the deployed environment. Software is continuously monitored to ensure that adequate testing is performed, to check conformance to required properties, and to validate and refine the models and assumptions on which previous quality assurance activities depend.

Professor Richardson has developed leading edge software testing and analysis technology. Richardson inspired much of the work in specification-based testing, which she began to explore in 1978 with her conception of the Partition Analysis Method, which proposed incorporating information from both specification and implementation in an integrated application of verification and testing techniques. Richardson developed the Relay model for the formal definition of test data selection criteria and evaluation of their fault detection capabilities. On the Arcadia project, Richardson collaborated on developing analysis and testing capabilities within a process-centered environment to support integration of and experimentation with a variety of techniques. She developed ProDAG, a program dependence analysis system that provides automated support for software understanding, debugging, test adequacy criteria, and maintenance, and TAOS, a testing environment that supports management of test assets, monitored test execution, automatic test result checking, and test coverage measurement. More recently, she developed the EASOF model of specification-based testing with support for execution-time checking of test results against formal specifications of required behavior. Richardson has worked with several companies in adopting sophisticated testing technology for improving the quality of their software products and processes

Richardson received the B.A. from the University of California in 1976 and the M.S. and Ph.D. in Computer Science at the University of Massachusetts in 1978 and 1981, respectively.

Selected Works

A. Reyes and D.J. Richardson. Specification-Based Testing of Ada Units with Low Encapsulation. *ASE'98: Proceedings of the 13th International Conference on Automated SoftwareEngineering*, Honolulu, HA, October 1998.

J. Chang and D.J. Richardson. ADLscope: an Automated Specification-based Unit Testing Tool. *ASE'98: Proceedings of the 13th International Conference on Automated SoftwareEngineering*, Honolulu, HA, October 1998.

C. Liu and D.J. Richardson. Software Components with Retrospectors. *ROSATEA: International Workshop on the Role of Software Architecture in Testing E(&) Analysis*, Marsala, Italy, July 1998.

H. Ziv and D.J. Richardson. Constructing Bayesian-network Models of Software Testing and Maintenance Uncertainties, *ICSM'97: Proceedings of the International Conference on Software Maintenance, Bari, Italy*, September 1997.

N.S. Eickelmann and D.J. Richardson. Leveraging the Cost of Software Testing with Measurable Process Improvement, *ETCE-ASME '97: Proceedings of the Computing in Engineering Conference* , Houston, TX, January 1997.

C.S. Turner, D.J. Richardson, and J.L. King. Legal Sufficiency of Testing Processes, *SAFECOMP`96: Proceedings of the Conference on Safety, Reliability and Security*, Vienna, Austria, October 1996.

D.J. Richardson and A.L. Wolf. Software Testing at the Architectural Level, *ISAW-2: Proceedings of the Second International Software Architecture Workshop*, San Francisco, October 1996.

H. Ziv, R. Klosch, D.J. Richardson. Software Re-Architecturing in the Presence of Partial Documentation, *ISAW-2: Proceedings of the Second International Software Architecture Workshop*, San Francisco, October 1996.

N.S. Eickelmann and D.J. Richardson. Evaluating Software Testability based on Software Architecure, *ISAW-2: Proceedings of the Second International Software Architecture Workshop*, San Francisco, October 1996.

Clarke, A. Podgurski, D.J. Richardson, S.J. Zeil. A Formal Evaluation of Data Flow Path Testing Criteria, in *Automating Specification-Based Software Testing*, IEEE Press, editor Bob Poston, June 1996. Also appears in *IEEE Transactions on Software Engineering*, November 1989.

T.O. O'Malley, D.J. Richardson, and L.K. Dillon. Efficient Specification-Based Test Oracles, *CSS'96: Proceedings of the Second California Software Symposium*, Los Angeles, CA, April 1996.

N.S. Eikelmann and D.J. Richardson. An Evaluation of Software Test Environment Architectures, *ICSE–18: Proceedings of the Eighteenth International Conference on Software Engineering*, Berlin, Germany, March 1996.

D.J. Richardson and M.C. Thompson. An Analysis of Test Data Selection Criteria Using the Relay Model of Fault Detection, *IEEE Transactions on Software Engineering*, June 1993.

David S. Rosenblum

Associate Professor
dsr@ics.uci.edu
http://www.ics.uci.edu/~dsr

Foundations of Component-Based Software Engineering

Dr. Rosenblum's research is centered on the study of techniques for validating large-scale software systems. His research is currently focused on problems related to the validation of distributed component-based software systems. This research has three specific thrusts: (1) development of a theory of test adequacy for component-based systems; (2) investigation of post-deployment monitoring techniques that can be used when opportunities for full-scale pre-deployment testing are limited; and (3) investigation of event observation and notification methods to support rapid construction of event-based distributed applications over wide-area networks. He and his students are investigating these problems in the context of the JavaBeans component model. Much of this research requires a deeper understanding of the relationship between software architecture and component-based development. Hence the research also involves investigation of the architectural foundations of component-based software engineering.

Rosenblum is also continuing his study of the problem of selective regression testing, an approach to regression testing in which static and dynamic test-coverage and program-change analysis are employed to eliminate unnecessary test cases during maintenance of a software system. Cost-effectiveness is a fundamental concern in selective regression testing, since the cost of the analysis employed by a method can easily exceed the savings it achieves in test execution time. TESTTUBE is a selective regression testing system that was built specifically to address the issue of cost-effectiveness, and it uses coarse-grained coverage and change analysis in order to achieve cost-effectiveness. Experience with TESTTUBE has led to the development of strategies for using test coverage information from a single version of a system to predict whether a selective regression testing method will be cost-effective for use on all future versions of the system. In addition, comparative evaluations of TESTTUBE and other methods are helping to further identify the factors that influence the cost-effectiveness of selective regression testing methods.

Selected Works

"A Type Theory for Software Architectures", with N. Medvidovic and R.N. Taylor, Department of Information and Computer

Science, University of California, Irvine, Technical Report UCI-ICS-98-14, May 1998, 10 pp.

"Integrating Architecture Description Languages with a Standard Design Method", with J.E. Robbins, N. Medvidovic and D.F. Redmiles, *Proc. 20th Int'l Conf. on Software Engineering*, Kyoto, Japan, Apr. 1998, pp. 209-218.

"Self-Adaptive Software", with P. Oreizy, R.N. Taylor, M.M. Gorlick, D. Heimbigner, G. Johnson, N. Medvidovic, A. Quilici, and A.L. Wolf, submitted to *IEEE Expert*, Mar. 1998.

"Empirical Studies of a Prediction Model for Regression Test Selection", with M.J. Harrold, G. Rothermel and E. Weyuker, submitted to *IEEE Transactions on Software Engineering*, Feb. 1998.

"Domains of Concern in Software Architectures and Architecture Description Languages," with N. Medvidovic, *Proc. USENIX Conf. on Domain-Specific Languages*, Santa Barbara, CA, Oct. 1997, pp. 199-212.

"A Design Framework for Internet-Scale Event Observation and Notification", with A.L. Wolf, *Proceedings of the Sixth European Software Engineering Conference/ACM SIGSOFT '97 Fifth Symposium on the Foundations of Software Engineering*, Zurich, Switzerland, September 1997, pp. 344-360.

"Adequate Testing of Component-Based Software", Department of Information and Computer Science, University of California, Irvine, Technical Report 97-34, August 1997.

"Using Coverage Information to Predict the Cost-Effectiveness of Regression Testing Strategies", with E.J. Weyuker, *IEEE Transactions on Software Engineering*, March 1997, pp. 146-156.

Isaac D. Scherson

Professor

isaac@ics.uci.edu

http://www.ics.uci.edu/~shark

Parallel Computing Architectures, Operating Systems, Parallel Algorithms, Performance Evaluation

Dr. Scherson's research interests fall in the general areas of parallel computer architecture and applications of concurrent computation. His research group is currently working on four main subjects: operating systems for parallel computers, interconnection networks, performance evaluation, and parallel algorithms. The design goal of operating systems for parallel computers is to provide a level of support to the programmer similar to that provided by current uniprocessor operating systems. The programmer programs a virtual machine with as many virtual processors as necessary to exploit the inherent parallelism of the application. The operating system emulates this virtual machine, making parallel programs portable. In this context various problems are being addressed: spatial and temporal scheduling of virtual processors, Load Balancing using a novel approach dubbed Rate of Change, efficient synchronization techniques, virtual memory management and I/O issues.

The work on interconnection networks for massively parallel systems involves the development of cost-effective high performance networks capable of supporting large numbers of processing elements. Included in this study is the performance analysis of Expanded Delta Networks (EDNs) and Least Common Ancestor Networks (LCANs) under commonly occurring sets of processor to processor communication patterns. As a result of the effort, efficient off-line routing algorithms for EDNs were developed and applied to commercially available massively parallel computers. Furthermore, the study of efficient management of network functions as part of the system resources is being considered.

Current research in performance evaluation deals with the development of models and methodologies for a general supercomputer performance evaluation theory. Such methods are being developed bottom up by building on known computational models and benchmarks. A new approach to the problem allowed us to define the performance evaluation problem using a cartesian model of program execution. Concepts such as the context lattice, support and operating points of computer programs have been introduced.

Selected Works

Isaac D. Scherson and Luis Miguel Campos, Efficient Task Scheduling Heuristic for Multiprocessor Systems, Parallel and Distributed Computing and Systems, Las Vegas, USA, Oct 1998.

Fabricio Silva, Luis Miguel Campos and Isaac D. Scherson, Improvements in Gang Scheduling for Parallel Computers, Parallel Computing Worshop, Singapore, Sept 1998.

Isaac D. Scherson and Luis Miguel Campos, A Distributed Dynamic Load Balancing Strategy Based on Rate of Change, Parallel Computing Worshop, Singapore, Sept 1998.

Fabricio Silva, Luis Miguel Campos and Isaac D. Scherson, A Lower Bound for Dynamic Scheduling of Data Parallel Programs, Euro-Par 98, Southampton, UK, Sept 1998.

Christophe Rapine, Isaac D. Scherson and Denis Trystram, On Line Scheduling of Parallelizable Jobs, Euro-Par 98, Southampton, UK,Sept 1998

V. L. Reis and Isaac D. Scherson, Impacts of Network Latency on Parallel Virtual Memory Management, Symposium on Performance Evaluation of Computer and Telecommunication Systems, Reno, USA, July 1998.

V. L. Reis and Isaac D. Scherson, Swap File Organizations for Parallel Virtual Memory, Parallel and Distributed Computing and Systems, Washington DC, USA, Oct 1997, pp 209-214.

V. L. Reis, Luis Miguel Campos and Isaac D. Scherson, Parallel Virtual Memory for Time Shared Environments, Eighth Brasilian Symposium on Computer Architectures and High Performance Processing, Recife, Brazil, August 1996, pp 223-233.

R.Subramanian, Isaac D. Scherson, V. L. Reis and Luis Miguel Campos, Scheduling Computationally Intensive Data Parallel Programs, Placement Dynamique et Repartition de Charge: application aux systemes paralleles et repartis, Paris, France, July 1996, pages 39-60.

U. Krishnaswamy and Isaac D. Scherson, Micro-architecture Evaluation Using Performance Vectors, SIGMETRICS'96, May 1996.

U. Krishnaswamy and I. D. Scherson, "A Methodology for Performance Evaluation of Supercomputers," International Parallel Processing Symposium - Parallel Systems Fair, pp 46-49, April 1993.

R. Subramanian and I. D. Scherson, "An Analysis of Diffusive Load-Balancing," ACM Symposium on Parallel Algorithms and Architectures, pp. 220-225, June 1994.

I. D. Scherson and C.K. Chien, "Least Common Ancestor Networks," VLSI Design, Vo. 2, No. 4, pp. 353-364, 1995.

V. Ramakrishnan, I. D. Scherson and R. Subramanian, "Efficient Techniquesfor Fast Nested Barrier Synchronization," ACM Symposium on Parallel Algorithms and Architectures, July 1995.

Padhraic Smyth

Associate Professor
smyth@ics.uci.edu
http://www.ics.uci.edu/~smyth

Pattern Recognition, Machine Learning, Decision Theory, Exploratory Data Analysis, Knowledge Discovery and Data Minings.

Faster computers, cheaper memory, and better sensors have produced vast data sets in fields as diverse as astronomy, medicine, and finance. Unfortunately we are effectively drowning in data in terms of our ability to extract useful information from these data sets. Dr. Smyth's research focuses on the development of theories, models, and algorithms for dealing with this data avalanche. In particular, his work focuses on the fundamental problem of what can be inferred in general from a finite amount of data. This encompasses topics such as classification, prediction, image and speech recognition, time series analysis, probabilistic reasoning, decision making, knowledge discovery, and data mining.

Dr. Smyth co-authored the first journal proof that the outputs of artificial neural networks could be interpreted as probabilities. He also showed how hidden Markov models, as used in speech recognition, can reduce false alarm rates dramatically over conventional approaches when used to monitor faults in complex systems. More recently he has shown that hidden Markov models are a special case of Bayesian belief networks used in probabilistic reasoning, providing a unified framework for both fields. Other recent work has led to a novel data-driven solution based on cross-validation to the well-known problem of determing how many clusters to use in cluster analysis.

A significant component of Dr. Smyth's work is an emphasis on practical applications. This serves both to test theoretical ideas in the real-world and generate new research problems to challenge current theory. There is frequent collaboration with scientists and researchers at institutions such as the Jet Propulsion Laboratory. Recent projects have included automated recognition of small volcanoes on the surface of Venus from remotely-sensed radar images, detecting clusters of pressure patterns in the Earth's northern hemisphere in daily records since 1948, and online annotation and segmentation of engineering time-series data from the Space Shuttle and other spacecraft. Other current projects include automated image analysis tools for diagnostic medicine, the application of classification and regression algorithms to drug discovery, and the use of belief networks for fast decoding of error-correcting codes in communications systems. What is surprising is that the theory behind all of these applications is remarkably similar. Conversely, theoretical results on how to turn raw data into useful information can have tremendous practical impact.

Selected Works

P. Smyth, "Hidden Markov models for fault detection in dynamic systems," *Pattern Recognition*, vol.27, no.1, pp 149-164, 1994.

C. Glymour, D. Madigan, D. Pregibon, and P. Smyth, "Statistics and data mining," *Communications of the ACM*, 39(11), 35-41, November 1996.

P. Smyth, D. Heckerman, and M.I. Jordan, "Probabilistic independence networks for hidden Markov probability models," *Neural Computation*, 9(2), 227-269, 1997.

M. Burl, L. Asker, P. Smyth, et al., "Learning to recognize volcanos on Venus" *Machine Learning*, 30(2-3), 165--194, 1998.

P. Smyth, K. Ide, and M. Ghil, "Multiple Regimes in Northern Hemisphere Height Fields via Mixture Model Clustering," *Journal of Atmospheric Science*, to appear.

Tatsuya Suda

Professor
suda@ics.uci.edu
http://www.ics.uci.edu/~suda

Computer Networks, High-Speed Multimedia Networks, Distributed Systems, Object Oriented Communication Systems, Performance Evaluation

Prof. Suda has been actively involved in research in the area of computer networks and distributed systems. His research goal is to design, analyze and implement communication systems that allow high speed transport of multimedia information between end-users. He attacks both networking issues (to allow high speed information transfer through a network) and end-system issues (to allow end-systems to process information in a flexible manner at a speed compatible with that of the networks). His research methodology includes both theoretical performance modeling, analysis, and simulation, an empirical design, implementation, and experimentation.

Prof. Suda's research collectively addresses a complete view of a communications network, from the network substrate (ATM networks, high speed LANs/MANs), end-to-end protocols (flexible and adaptive protocol support) and communication subsystem architectures, to distributed applications.

Prof. Suda's recent research projects include (1) Video Multicast and Quality of Service, (2) Wireless/Mobile Networks, (3) Object-Oriented Frameworks for High Speed Networks, and (4) High Speed Asynchronous Transfer Mode (ATM) Networks.

Selected Works

B. J. Vickers, M. Lee and T. Suda, "Feedback Control Mechanisms for Real-Time Multipoint Video Services," the IEEE JSAC, Vol.15, No.3, April 1997.

C. Oliveira, J. B. Kim and T. Suda, "An Adaptive Bandwidth Reservation Scheme for High-Speed Multimedia Wireless Networks," to appear in the IEEE JSAC.

L. Chen, and T. Suda, "Designing Mobile Computing Systems Using Distributed Objects," the IEEE Communications Magazine, Vol. 35, No. 2, Feb., 1997.

D. C. Schmidt and T. Suda, "A Framework for Measuring the Performance of Alternative Process Architectures for Parallel Communication Subsystems," to appear in The State-of-the-art in Performance Modeling and Simulation, Kallol Bagchi, Ed., Gordon and Breach Publishers.

D. C. Schmidt and T. Suda, "Measuring the Performance of Parallel Message-based Process Architectures," Proc. of the IEEE Infocom, '95.

T. Suda, "Asynchronous Transfer Mode (ATM) Networks," to appear in the Encyclopedia of Electrical and Electronics Engineering, Wiley and Sons, planned to be published in 1998.

L. Chen, B. J. Vickers, J. Bae Kim, T. Suda and E. Lesnansky, "ATM and Satellite Distribution of Multimedia Educational Courseware," *Proc. of IEEE ICC '96.*

T. Takine, T. Suda and T. Hasegawa, "Cell Loss and Output Process Analysis of a Finite-Buffer Discrete-Time ATM Queueing System with Correlated Arrivals," *the IEEE Trans. on Commun.*, March, 1995.

Richard N. Taylor

Professor
taylor@ics.uci.edu
http://www.ics.uci.edu/~taylor

Software Engineering: Architectures, Hyperware, Workflow/Process Technology, Environments

Dr. Taylor's research is focused on the design and evolution of complex systems and processes. His approach has three major components: hyperware, software architecture, and workflow/process technology. The hyperware project is directed at enabling rich linkages between heterogeneous information sources, such as are necessary to support design rationale, and includes the evolution and enhancement of core WWW technologies. The software architecture project is focused on means for describing, implementing, and evolving highly flexible and distributed applications, particularly those which have graphical user interfaces. The architecture work emphasizes software reuse and has resulted in creation of a new architectural style. The workflow/process technology is directed at providing Intranet/Internet-based support to teams of engineers engaged in cooperative tasks. The support is keyed on project control and coordination. An entirely Java-based prototype system, Endeavors, is an outgrowth of this work. All three technologies are being integrated and brought to bear in the context of DARPA's Evolutionary Design of Complex Systems project.

Taylor was a 1985 recipient of an NSF Presidential Young Investigator Award. He was Program Co-Chair of the 1997 International Conference on Software Engineering (ICSE 97) and was the national chairman of ACM's Special Interest Group on Software Engineering, SIGSOFT, from 1989-1993. Taylor serves as the Director of IRUS, an alliance between the university and California industry, designed to promote cooperative problem solving and technology transition. In 1998 Taylor was named an ACM Fellow.

Selected Works

Roy Fielding, E. James Whitehead, Jr., Kenneth Anderson, Peyman Oreizy, Gregory Bolcer, and Richard Taylor. "Support for the Virtual Enterprise: Web-based Development of Complex Information Products." Communications of the ACM. August 1998.

G. Bolcer and R. Taylor. "Endeavors: A Process System Integration Infrastructure." *Proceedings of the 4th International Conference on the Software Process*, Brighton, England, pp. 76-85. December 1996.

Peyman Oreizy, Nenad Medvidovic, and Richard N. Taylor. "Architecture-based runtime software evolution." *Proceedings of the 1998 International Conference on Software Engineering*, Kyoto, Japan, pp. 177-186, April 1998.

Nenad Medvidovic and Richard N. Taylor. "A Framework for Classifying and Comparing Architecture Description Languages." *Proceedings of the Joint Fifth ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE) and European Software Engineering Conference (ESEC)*, October, 1997, pp. 60-76.

K. Anderson, R. Taylor, and E. J. Whitehead, Jr. "Chimera: Hypertext for heterogeneous software environments." *Proceedings of the ACM European Conference on Hypermedia Technology*, Edinburgh. September 1994.

Nalini Venkatasubramanian

Assistant Professor
nalini@ics.uci.edu
http://www.ics.uci.edu/~nalini

Concurrent/distributed systems, multimedia systems and applications, resource management in distributed systems, distributed object technologies, Quality of Service and customizable middleware, formal reasoning of distributed systems.

Prof. Venkatasubramanian's research focuses on enabling effective management and utilization of resources in the evolving global information infrastructure. Global information systems consist of asynchronous, autonomous components that are open and distributed. Applications that execute in this environment, e.g. multimedia (MM), have QoS (Quality of Service) parameters that define the extent to which performance specifications such as responsiveness, reliability, availability and security may be violated. Furthermore, service providers that deploy services for wide-area applications require cost-effective utilization of resources. Varying requirements posed by applications, customers, and service providers makes the task of resource management in the evolving global information infrastructure complicated and challenging - one with significant commercial impact as well.

Venkatasubramanian's research addresses issues in the development of a flexible, component based middleware infrastructure for global distributed computing with services that provide reliability, availability, security and QoS. Using Actors, a model of concurrent active objects, she develops meta-architectures that permits customization of policies for placement, scheduling, synchronization and management of components.

The research also involves the development of theoretical foundations, i.e. a semantic model for representing components in open distributed systems, techniques for reasoning about their interaction and ensuring safe composability of services in the presence of QoS requirements. Venkatasubramanian has worked on policies for cost-effective load management in distributed multimedia servers. Her current work includes techniques for the safe integration of mechanisms for load balancing, fault tolerance and end-to-end QoS management.

The development of a flexible and safe middleware framework will provide a platform for prototyping and building applications over the global infrastructure. Network-centric computing and mobile environments are two interesting applications that Nalini intends to explore. A *network-centric model* consists of a customizable global networking backbone that other components can dynamically attach to or detach from. Mobile environments allow applications to execute anywhere in the global infrastructure. Mobile applications also require support for the creation, execution and migration of agents and mechanisms for the composition of migratory agents with other system customizations. Other applications of interest include distributed transactional and fault-tolerant systems. The customizable framework developed can be used for the dynamic installation of policies to satisfy security, mobility, integrity and fault-tolerance requirements of applications.

Nalini has also worked on economic models used to charge for QoS in a distributed multimedia environment and would like to pursue this as well. She is also interested in applications of distributed computing and multimedia systems to art and the system level issues that arise from requirements in this domain.

Selected Works

Nalini Venkatasubramanian & Klara Nahrstedt. *An Integrated Metric for Video QoS.* Proceedings of ACM Multimedia '97, Seattle, Washington, November 1997, pp. 371-381.

Shangping Ren, Nalini Venkatasubramanian and Gul Agha. "Formalizing QoS Constraints Using Actors". *Proceedings of Second IFIP International Conference on Formal Methods for Open Object Based Distributed Systems, FMOODS'97*. July 1997. Also post-conference book, Editors Bowman and Derrick. Published by Chapman and Hall, pp. 139-157.

Nalini Venkatasubramanian and Srinivas Ramanathan. "Load Management for Distributed Video Servers", *Proceedings of the International Conference on Distributed Computing Systems (ICDCS '97)*, May 1997, pp. 528-536.

Nalini Venkatasubramanian and Carolyn Talcott. "MetaArchitectures for Resource Management in Open Distributed Systems", *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC '95)*, Aug. 1995, pp. 144-153.

DISTINGUISHED SPEAKER SERIES

1997-1998

Daryl Pregibon, AT&T Laboratories
"Mega-monitoring: Data Warehousing and Data Mining in Telecommunications"
October 20, 1997

Leslie Valiant, Harvard University
"Cognitive Computation"
November 17, 1997

Donald A. Norman, Hewlett-Packard Laboratories
"Taming technology: Why having the best technology is irrelevent,
or Where Edison went wrong."
December 1, 1997

Patrick M. Hanrahan, Stanford University
"Digital Lights, Cameras and Materials"
January 14, 1998

Niklaus Wirth, ETH Zurich, Switzerland
"Forgotten Hints on Programming Language Design"
March 2, 1998

Colloquia

1997-1998

Frank M. Shipman, III
"Making Networked Information Useful and Usable"
July 7, 1997

Martin Gitsels, Siemens Corporate Research
"Internet-Telephony based on the H.323 Standard"
September 22, 1997

Ben Shneiderman, University of Maryland
"The Eyes Have It: Understanding Information Visualization"
October 28, 1997

Prof. Dr. Laszlo Boszormenyi, University of Klagenfurt (Austria)
"Parallel, Persistent and Polymorphic Object-Sets"
November 18, 1997

Ugo Montanari, SRI International
"Constraint Programming for Coordination for Constraint Programming"
November 19, 1997

Bonnie Nardi
"The Activity Checklist: A Tool for Representing the "Space" of Context"
November 24, 1997

Prof. Bill Lin, University of California, San Diego
"Design of Embedded Systems"
December 5, 1997

Associate Prof. Bonnie John, CMU
"Cognitive Modeling in Human-Computer Interaction"
December 10, 1997

David Zicarelli
"Making Sound with Visual Computer Tools"
January 14, 1998

Dr. Robert P. Grafton, Program Director, National Science Foundation
"NSF-CISE 98: The Reorganization -- Programs, Structure, People and Thrusts"
January 27, 1998

Dr. Craig Benham, Mount Sinai School of Medicine
"Computational Analysis of DNA Structural Transitions"
January 29, 1998

Loren Terveen, AT & T Labs
"An Introduction to AT&T Labs"
November 22, 1996

Loren Terveen, AT & T Labs
"WebCite: Finding and Visualizing Structure in Hyper-Linked Collections"
January 30, 1998

Hsinchun Chen, University of Arizona
"Trailblazing Path to Semantic Interoperability in Digital Library Research"
February 2, 1998

Wanda Pratt, Stanford University
"Dynamic Categorization: A Method for Decreasing Information Overload"
February 12, 1998

Robert B. Allen, Bellcore
"Navigating Digital Libraries and Multimedia Archives"
February 16, 1998

Jason Cong, UCLA
"VLSI Interconnect Layout Optimization in Deep Submicron Designs"
February 20, 1998

Dietlind L. Gerloff, UC San Francisco
"Protein Structure and Function Prediction through Evolutionary Sequence Analysis -- Results and ProposedApplications in the Post-Genomics Era"
February 23, 1998

Christian F. Tschudin, University of Zurich
"The Geneva Messengers"
February 26, 1998

Niklaus Wirth, ETH Zurich, Switzerland
"Hardware Compilation: The Translation of Programs into Circuits"
March 4, 1998

Sudarshan S. Chawathe, Stanford University
"Managing Change in Autonomous Databases"
March 5, 1998

Nick Koudas, University of Toronto
"Fast Algorithms for Spatial and Multidimensional Joins"
March 6, 1998

Nalini Venkatasubramanian, University of Illinois at Urbana-Champaign
"An Adaptive Resource Management Architecture for Global Distributed Computing"
March 9, 1998

Sharad Mehrotra, University of Illinois at Urbana-Champaign
"Multimedia Analysis and Retrieval System (MARS) Project"
March 10, 1998

Forrest Brewer, University of California, Santa Barbara
"Production Based Synthesis Package for Finite State Machines"
March 13, 1998

Kenneth Y. Yun, Assistant Professor, University of California, San Diego
"The Design and Verification of a High-performance Low-control-overhead Asynchronous Differential Equation Solver"
March 20, 1998

Alexander V. Veidenbaum, Dept. of Elec. Engr. & Comp Science, Univ of Illinois at Chicago
"Instruction Cache Prefetching Using Multilevel Branch Prediction"
March 30, 1998

Pedro M. Domingos, Professor, Technical University of Lisbon
"A Process-Oriented Heuristic for Model Selection"
April 3, 1998

Dr. Dr. Norbert A. Streitz
GMD-German National Research Center for Information Technology
IPSI - Integrated Publication and Information Systems Institute, Darmstadt, Germany
"Cooperative Buildings: Extending the scope of CSCW beyond desktops"
April 24, 1998

Dr. Michael Turmon, Jet Propulsion Laboratory, Pasadena, CA
"Integrating Multiple Images for Segmentations"
April 24, 1998

Carl Kesselman, Research Associate Processor, Department of Computer Science
Project Leader, USC/Information Science Institute
"The Globus Project: building a computational grid"
April 27, 1998

Steve Seiden, Technische Universitaet Graz, Institut fuer Mathematik B
"On the Power of Barely Random Online Algorithms"
April 30, 1998

Dr.Barry M. Pangrle, Synopsys Inc.
"Protocol Compiler"
May 20, 1998

Robert J. Glushko, Director, Information Engineering, Veo Systems
Program Manager for eCoNet
"The XML Revolution"
May 28, 1998

Sam Roweis, Princeton University
"EM Algorithms for PCA and SPCA"
June 5, 1998

Recent Books Authored by ICS Faculty

Memory Issues in Embedded Systems-on-Chip: Optimizations and Exploration

Written by: Preeti Ranjan Panda, Nikil Dutt and Alexandru Nicolau
Kluwer Academic Publishers, 1998

As embedded systems realize the convergence of computers, communications, and multimedia into consumer products, system designers will need rapid exploration tools to evaluate candidate architectures for Systems-on-a-Chip (SOC) solutions. Such exploration tools will facilitate the introduction of new products customized for the market under increasingly shorter time frames. This book focuses on optimizations for, and exploration of the memory subsystem for processor-core-based embedded systems.

Traditionally, the memory subsystem has been a major bottleneck in the design of high-performance processor-based systems; due to the rapidly increasing gap between memory and processor performance, this memory subsystem bottleneck becomes even more critical for memory-intensive embedded system applications that process large volumes of data under demanding throughput, cost and power constraints. This book covers techniques for optimization of system-level memory requirements, and exploration of candidate memory architectures for implementing processor-core-based embedded systems.

Principles of Digital Design

Written by Daniel D. Gajski
Prentice Hall, 1997

This is an introductory text for computer science and electrical engineering curricula. It introduces the basics of design process, CAD tools, component libraries and design techniques starting with number representation, Boolean algebra, and an axiomatic approach to design. It treats in detail logic synthesis and optimization, and sequential synthesis and optimization; it also introduces combinatorial and sequential register-transfer libraries and shows how to build standard and custom microchips starting from ordinary algorithms. The book also demonstrates how to build CISC and RISC processors and large systems. The book features step-by-step procedures in each chapter, comprehensive examples that demonstrate designer's options and choices and excellent color illustrations that enhance learning and material retention.

Lecture Notes in Computer Science, vol. 1075, Combinatorial Pattern Matching, Proceedings 1996

Edited by D. Hirschberg and G. Myers
Berlin, Germany: Springer-Verlag, 1996

Combinatorial Pattern Matching addresses issues of searching and matching strings and more complicated patterns such as trees, regular expressions, graphs, point sets, and arrays. The goal is to derive non-trivial combinatorial properties for such structures and then exploit these properties in order to achieve improved performance for the corresponding computational problems.

Advanced Topics in Dataflow and Multithreading

Edited by Guang Gao, Lubomir Bic, and Jean-LucGaudiot
IEEE Computer Science Press, 1995

Dataflow ideas have become intertwined with the main stream of computer system thought in areas as diverse as compiler construction and database access. This book provides, in a single volume, reports from many of the world's projects engaged in the current evolution and application of dataflow concepts. Their subject matter emphasizes the broad reach of dataflow principles in program representation from language design to processor architecture and compiler optimization techniques.

A basic question in computer architecture has been how to build effective largescale parallel computers, with performance and programmability being central issues. The field continues in a state of flux. The data parallel model, although specialized, is applied with success to significant scientific computations.

At the same time, many computer scientists proclaim the advantages of the shared memory model even though its support for concurrent processing fails to satisfy basic principles of modular program construction. A general semantic model of program execution is needed that can serve as a standard target for the compilation of high-level languages. Dataflow concepts have shown how functional programming ideas may be an important universal semantic model of computation. The reports published here are stepping stones along the way.

Parallel Language and Compiler Research in Japan

Edited by Lubomir Bic, Alex Nicolau, and Mitisuhisa Sato
Kluwer Publishers, 1995

The contributions of Japanese scientists, engineers, and corporations in "hardware-oriented" fields such as electronics miniaturization, video and audio recording, optics and photography, and manufacturing quality are well-known internationally because of the many products exported world-wide that inncorporate these advances. However, many other areas of Japanese science and technology are not so well known. For example, Japanese innovations and advances in software technology are reported in scattered forums, with relatively little independent assessment or uniformity of presentation. Few Japanese software products are available internationally. As a result of these factors, the international technical community is only marginally aware of Japanese software research efforts, and perhaps underestimates their potential importance.

The present volume offers the international community an opportunity to learn in-depth about key Japenese research efforts in the particular software domains of parallel programming languages and parallelizing compilers. These are important topics that strongly bear on the effectiveness and affordability of high performance computing systems. The chapters of this book convey a comprehensive and current depiction of leading edge research efforts in Japan that focus on parallel software design, development, and optimization that could be obtained only through direct and personal interaction with the researchers themselves.

Readings in Human-Computer Interaction: Toward the Year 2000
Written and edited by Ronald M. Baecker, Jonathan Grudin,
William A. S. Buxton, and Saul Greenberg
San Mateo, CA: Morgan Kaufmann, 1995

This is the second edition of a highly-rated, widely-read collection of readings together with detailed overviews and literature reviews spanning this rapidly-growing field. Human-computer interaction is now considered to be a core area within computer science, while also being a field drawing on professionals from other scientific and design disciplines. The book begins with an historical and intellectual overview of the field, followed by chapters on design and evaluation, software development contexts and tools, visual and auditory channels, input modalities, and human information processing. Research frontiers receiving expanded coverage in this edition include groupware and computer-supported cooperative work, tailorable and intelligent interfaces, hypertext, multimedia, the Internet, World Wide Web, and ubiquitous computing.

Co-Synthesis of Hardware and Software for Digital Embedded System
Written by Rajesh Kumar Gupta
Boston, MA: Kluwer Academic Publishers, 1995
The Kluwer International Series In Engineering and Computer Science, Volume 329

Co-Synthesis of Hardware and Software for Digital Embedded Systems, with a Foreword written by Giovanni Micheli, presents techniques that are useful in building complex embedded systems. These techniques provide a competitive advantage over purely hardware or software implementations of time-constrained embedded systems.

Recent advances in chip-level synthesis have made it possible to synthesize application-specific circuits under strict timing constraints. This work advances the state of the art by formulating the problem of system synthesis using both application-specific as well as reprogrammable components, such as off-the-shelf processors. Timing constraints are used to determine what part of the system functionality must be delegated to dedicated application-specific hardware while the rest is delegated to software that runs on the processor. This co-synthesis of hardware and software from behavorial specifications makes it possible to realize real-time embedded systems using off-the-shelf parts and a relatively small amount of application-specific circuitry that can be mapped to semi-custom VLSI such as gate arrays. The ability to perform detailed analysis of timing performance provides the opportunity of improving the system definition by creating better phototypes.

Co-Synthesis of Hardware and Software for Digital Embedded Systems is of interest to CAD researchers and developers who want to branch off into the expanding field of hardware/software co-design, as well as to digital system designers who are interested in the present power and limitationsof CAD techniques and their likely evolution.

Interconnection Networks for High-Performance Parallel Computers

Edited by Isaac D. Scherson and Abdou S.Youssef
Los Alamitos, CA: IEEE Computer Society Press, Revised 1995

Since communication overhead is one of the most important factors affecting the performance of parallel computer systems, the study of interconnection networks is one of the most popular research areas in parallel processing. This book, a collection of 70 papers, addresses the basic problems encountered in the design analysis, use, and reliability of interconnection networks.

The book also looks into partitioning, discusses time sharing of resources to increase system throughput, details the communication requirements of different applications, and examines fault tolerance and the issues of survivability achieved by hardware and software techniques. It concludes with a discussion of the relatively new area of fault diagnosis and details important research results on newly implemented methods.

Advances in Knowledge Discovery and Data Mining

Edited by Usama M. Fayyad, Gregory Piatetsky-Shapiro,
Padhraic Smyth, and Ramasamy Uthurusamy
Menlo Park, CA: AAAI Press, 1995

During the last decade, we have seen an explosive growth in our capabilities to both generate and collect data. Advances in data collection, widespread use of bar codes for most commercial products, and the computerization of many business and government transactions have flooded us with information, and generated an urgent need for new techniques and tools that can intelligently and automatically assist us in transforming this data into useful knowledge. This book examines and describes many such new techniques and tools, in the emerging field of data mining and knowledge discovery in databases (KDD).

The chapters of this book span fundamental issues of knowledge discovery, classification and clustering, trend and deviation analysis, dependency derivation, integrated discovery systems, augmented database systems, and application case studies.

Active Contracts and Grants

The following list includes the name of the Principal Investigator, the Funding Source and the Title of the Contract or Grant.

Corporate Funded

M. Ackerman, California Micro, Quality Systems Inc
Cooperative Information Sharing Interfaces

R. Dechter, California Micro, Rockwell/Amada
Diagnosis and Planning Using Network-Based Algorithms

M. Dillencourt/S. Irani, California Micro, Kofax Image Products
Pattern Recognition in Business Documents

N. Dutt/F. Kurdahi, California Micro, Viewlogic Systems Inc.
High-Level Synthesis for Deep-Submicron Designs

D. Gajski, Hitachi
Codesign System for Telecommunication Applications

D. Gajski, Hitachi
Codesign Methodology and Verification

D. Gajski, Rockwell
System Design and Exploration

D. Gajski, California Micro, Rockwell Semiconductor Systems
Codesign Visualizer

D. Gajski, Semiconductor Research Corporation - Year 10,
System Level Tools

D. Gajski, Toshiba
Tools for Super Design Technology

R. Gupta, California Micro, Escalade Corp.
Behavioral Reuse for Silicon Building Blocks Using Presynthesis Optimizations

R. Gupta, California Micro, Synopsis Corporation
Efficient Hardware Modeling and Optimizations for Architectural Adaptation

M. Pazzani, California Micro, Touchstone Software Corp.
Knowledge Discovery in Problem Reports

M. Pazzani, California Micro, Daimler Benz
Adaptive Navigation in Information Space

D. Redmiles, California Micro, Rockwell
Applying Design Critics to Software Refquirements Engineering

D. Richardson, California Micro, Hughes
An Integrated Toolset for Specifying and Testing Complex Software-Intensive Systems

D. Richardson, California Micro, Microelectronics and Computer Technology Corporation
Quest Project

D. Rosenblum, California Micro, Northrop
Specifying and Checking Integrity Constraints in Distributed Avionics Systems

P. Smyth, Smith Kline Beecham
Exploring QSAR Data Using Probabilistic Data Mining

T. Suda, California Micro, Hitachi/SMC
Supporting Multimedia Traffic in Heterogeneous Network Environments

T. Suda, California Micro, Hitachi/Canon
Scalable Multimedia Services in High-Speed Network Environments

R. Taylor, California Micro, Hughes
Employing Connector Technologies in Software Architectures

R. Taylor, California Micro, Sun Microsystems
Process/Workflow Technology in Support of Computer-Mediated Learning

R. Taylor, California Micro, Sun Microsystems
Process/Workflow Technology in Support of Computer-ModulatedLearning

Federal Funded

M. Ackerman, National Science Foundation
CAREER: Augmenting Expertise Networks

M. Ackerman, Navy
Review of Collective Action Tools

L. Bic, Department of Education
Graduate Fellowships in Biomedical Computing

L. Bic/M. Dillencourt, National Science Foundation
MESSENGERS: A Coordination Paradigm for Distributed Computing

R. Dechter, National Science Foundation
Tractable Reasoning

N. Dutt/A. Nicolau, National Science Foundation
Design Space Exploration for Memory-Intensive Embedded Systems

D. Eppstein, National Science Foundation

NSF Young Investigator: Algorithms for Molecular Biology, Optimal Triangulation, Minimum Spanning Trees, and Geometric Optimization

M. Franz, National Science Foundation

Dynamic Optimization of Software Component Systems

R. Gupta, National Science Foundation

Architecture Evaluation and Synthesis Techniques for Embedded Systems

R. Gupta/A. Nicolau, Defense Advanced Research Projects Agency

Coordinated Compile-Time and Runtime Reconfiguration with Safe Execution and Continuous Validation of Adaptive Computing Systems

S. Irani, National Science Foundation

Competitive Analysis of Problems in Resource Allocation

R. Lathrop, National Science Foundation

CAREER: Intelligent Systems and Advanced Computation in Molecular Biology

R. Lathrop, Department of Energy

Intelligent Systems for Molecular Biology Conference to be held June 28 - July 1, 1998

A. Nicolau, DARPA

PROMIS: A Unified Compilation Approach to Fine and Coarse Grain Parallelization and Resource Management

A. Nicolau, Office of Naval Research

Unified Compilers for Instruction Level Parallelism

A. Nicolau, Office of Naval Research

Integrated Resource Allocation and Memory Management for DSP and ASIP Processors

M. Pazzani, DOED

Graduate Assistance in Areas of National Need (GAANN) - UCI Information and Computer Science

M. Pazzani, NSF

From Computer Data to Human Knowledge: A Cognitive Approach to Knowledge Discovery and Data Mining

D. Redmiles, National Science Foundation

CAREER: Improving the Design of Interactive Software

D. Richardson, Air Force Rome Air Development Center

Perpetual Testing

D. Richardson, National Science Foundation

A Formal Architecture-Based Approach to Software Integration Testing

D. Richardson, National Science Foundation
U.S. - Italy Workshop on the Role of Software Architecture in Testing and Analysis:
Marsala, Italy, July 1-3, 1998

D. Rosenblum, National Science Foundation
CAREER: Mechanisms for Ensuring the Integrity of Distributed Object Systems

D. Rosenblum, Air Force Office of Scientific Research
Specification and Dynamic Checking of Composition Constraints in Distributed Computing

I. Scherson, NASA
A Data-Parallel Computing Environment (Shark OS)

P. Smyth, AFOSR/Cal-Tech
Turbo Decoding of High Performance Error-Correcting Codes via Belief Propogation Algorithms

P. Smyth, National Science Foundation
Probabilistic Knowledge Discovery and Data Mining: An Integrated Approach at the Interface of Computer Science and Statistics

P. Smyth/M. Pazzani/D. Kibler, NSF
SGER: An Online Repository of Large Data Sets for Data Mining Research and Experimentation

T. Standish, DOED
Graduate Assistance in Areas of National Need (GAANN) - UCI Information and Computer Science

T. Suda, National Science Foundation
Adaptive Multicast Transport of Realtime Video ATM Networks

R. Taylor/D. Redmiles, ARPA
Open Technology for Software Evolution: Hyperware, Architecture and Process

THE UNDERGRADUATE PROGRAM

Objectives for the ICS major: Creating the leaders of tomorrow

The ICS department has several objectives for its majors. Computers and computer software are ubiquituous. Their performance and utilization can drastically affect peoples lives. Medical records, medical diagnoses, bank accounts, machine control, credit records, etc, are but a few of examples of where computer science impacts all of us. It is imperative that ICS students hold the highest standards of technical competency, social responsibility and individual integrity.

A major in ICS provides the serious student the opportunity to study in depth the underlying principles, current practice, and probable future trends of computer science. Just as it important for the student to acquire a broad and basic education in the major segments of modern academic study, it is essential that the student obtain an educational foundation that will permit him or her to continue to learn and keep up with the expanding field of computer science. This focus on foundations for lifelong learning is the primary guideline for the ICS curriculum. The curriculum and its implementation in specific course offerings permits the student to study basic concepts and practices of data organization, algorithmic design, hardware and software systems organization, software system design and construction, theoretical models and analysis, networks and distributed computing, artificial intelligence, and the nature of the personal, social, and organizational impact of computers.

Students may elect to complete one or more specializations within the major in Information and Computer Science. These specializations are:

a) Artificial Intelligence, which prepares students in both the theory and implementation of artificial intelligence methods, covering such topics as search, expert systems, neural nets, reasoning, constraint satisfaction, and machine learning;

b) Computer Systems, in which students acquire in-depth knowledge on the design and development of software and hardware computing systems for a number of application domains;

c) Implementation and Analysis of Algorithms, which provides students with a course of study that includes the understanding of the theoretical basis of algorithms and their analysis, as well as providing a significant practical programming experience;

d) Information Systems, in which students study the aspects of management and organizational behavior necessary to understand how computing systems are built, adopted, and maintained in companies and organizations;

e) Networks and Distributed Systems, which enables students to focus their studies on the rapidly expanding subarea of computer science that deals with networking architectures, communication protocols, programming paradigms, distributed algorithms, and distributed applications;

f) Software Systems, which enables students to focus their studies on topics related to the analysis, design, development, and evolution of large-scale software systems.

In the process of mastering the ICS curriculum, the student encounters a variety of programming languages, operating systems, support tools such as graphics packages, and hardware systems. The student also learns to use various current systems aimed at improving personal productivity. These skills are primarily aimed at providing an educational basis for professional work and advanced graduate study in computer science and sophisticated applications of computers. While many ICS graduates are successfully pursuing careers in a variety of fields, including applications development, the curriculum does not provide intensive work in computer applications. Students whose interests are primarily applications-oriented are encouraged to pursue in-depth study in another field, combined with a basic education or minor in ICS.

STUDENT PROFILE

Although most UCI students are from Orange and Los Angeles Counties, as the reputation of the campus has grown, the number of students from other parts of California and elsewhere has increased to a current level of 27%. Opportunities for computing professionals in this area are extensive and many software development, computer manufacturing, and information technology consulting firms recruit at UCI. Since the demand for qualified computer professionals is likely to persist, programs to fulfill the aspirations of students and meet society's needs are in high demand.

THE MASTERS PROGRAM

The objective of the terminal M.S. degree is to prepare students for a variety of advanced careers in the computing industry such as software development, information technology consulting, application specific integrated circuit design, network design and configuration, embedded systems design, algorithms development, data mining etc. The current ICS graduate curriculum and M.S. degree requirements include advanced courses in these areas that enhance the career options of students that graduate with an M.S. degree. 10-15 students per year will be admitted to the program.

The Orange County area alone has over 500 companies directly involved in the computer software area which are predicted to experience a long-term 10% per annum growth rate. Similarly, the computing hardware, embedded systems, and the information technology consulting fields are experiencing unprecedented growth. This industry trend is by no means limited to the Orange County region. Indeed, approximately half of the companies that recruit our Information and Computer Science majors are not from the Orange County region, and represent firms from Silicon Valley (such as Sun and Tandem Computers), the Pacific Northwest (such as Microsoft) and other states in the U.S. (such as Motorola in Illinois and Texas). These firms consistently express a strong interest in students earning an M.S. degree. While the bachelor's degree will prepare a student for entry level positions, the M.S. degree provides the in-depth training needed for more advanced computing positions, particularly in smaller companies.

THE DOCTORAL PROGRAM

The Doctoral Program in Information and Computer Science is a research-based curriculum in which students work together with faculty to solve advanced problems in computer science. The program is designed to prepare professors and researchers for positions in universities, industry, and government.

The graduate program leads to a Ph.D. degree in Information and Computer Science with a concentration in one of the following six areas: Artificial Intelligence (AI); Computer Systems Design (CSD); Computer Systems and Networks (CSN); Computers, Organizations, Policy, and Society (CORPS); Software (SW); and Algorithms and Data Structures (Theory). Additionally, an area of specialization other than one of these concentrations may be chosen with the approval of the graduate advisor.

The program is research oriented and encourages students to work together with faculty to solve advanced problems in computer science. The program is designed for full-time study, and can be completed in five to six years depending upon the focus of research. Students enrolled in the Ph.D. program must maintain satisfactory academic progress.

Quality of our Graduate Students: The students who have earned doctorates in ICS hold faculty positions in major universities all over the country, responsible research positions in major non-profit and industrial labs, and are owners of successful high-tech companies.

The graduate program in ICS attracts a large number of highly qualified applicants. We select a small portion of these students whose research interests and ability align with those of the faculty.

Doctorates of Philosophy Conferred

The following list includes the name of the recipient, the year the degree was granted, and the recipient's advisor.

William Howden, 1973
Advisor: Julian Feldman

Marsha Hopwood, 1974
Advisor: Fred Tonge

Donald Loomis, 1974
Advisor: Julian Feldman

Philip Merlin, 1974
Advisor: David Farber

Harry Wechsler, 1975
Advisor: Jack Sklansky

Richard Burton, 1976
Advisor: John Brown

Ronald Colman, 1976
Advisor: Fred Tonge

Arthur Duncan, 1976
Advisor: Thomas Standish

Steven Levin, 1976
Advisor: Peter Freeman

Lawrence Rowe, 1976
Advisor: Fred Tonge

Ram Singhania, 1976
Advisor: Fred Tonge

Patrick Hanratty, 1977
Advisor: Thomas Standish

Kenneth Larson, 1977
Advisor: David Farber

Lubomir Bic, 1978
Advisor: Kim Gostelow

Gregory Hopwood, 1978
Advisor: Julian Feldman

Dennis Kibler, 1978
Advisor: Thomas Standish

David Feign, 1980
Advisor: Thomas Standish

Dov Harel, 1980
Advisor: George Lueker

James Neighbors, 1980
Advisor: Peter Freeman

Wilfred Plouffe, 1980
Advisor: Kim Gostelow

Walter Scacchi, 1981
Advisor: Rob Kling

Robert Thomas, 1981
Advisor: Kim Gostelow

Stephen Fickas, 1982
Advisor: Dennis Kibler

David Keirsey, 1982
Advisor: James Meehan

David Keirsey, 1982
Advisor: James Meehan

Paul Mockapetris, 1982
Advisor: Thomas Standish

David Smith, 1982
Advisor: Thomas Standish

Allan Terry, 1982
Advisor: James Meehan

Robert Bechtel, 1983
Advisor: James Meehan

John Conery, 1983
Advisor: Dennis Kibler

Steven Hampson, 1983
Advisor: Dennis Kibler

Joseph Minne, 1983
Advisor: Lubomir Bic

Joseph Minne, 1983
Advisor: Lubomir Bic

Randell Flint, 1984
Advisor: Nancy Leveson

Leslie Gasser, 1984
Advisor: Rob Kling

Paul Morris, 1984
Advisor: Dennis Kibler

Bruce Porter, 1984
Advisor: Dennis Kibler

Marshall Rose, 1984
Advisor: Rami Razouk

Gene Fisher, 1985
Advisor: Thomas Standish

Martin Katz, 1985
Advisor: Dennis Volper

Francis Murgolo, 1985
Advisor: George Lueker

Ruben Prieto-Diaz, 1985
Advisor: Peter Freeman

Stephen Whitehill, 1985
Advisor: Thomas Standish

Michael Friedman, 1986
Advisor: Nancy Leveson

Gabriel Goren 1986
Advisor: John Leslie King

Lawrence Larmore, 1986
Advisor: Daniel Hirschberg

Timothy Morgan, 1986
Advisor: Rami Razouk

Stephen Willson, 1986
Advisor: Richard Taylor

Rika Yoshii, 1986
Advisor: Thomas Standish

Douglas Fisher, 1987
Advisor: Dennis Kilber

Robert Hartmann, 1987
Advisor: Lubomir Bic

Robert Rittenhouse, 1987
Advisor: Rob Kling

Jeffrey Schlimmer, 1987
Advisor: Richard Granger

Richard Sidwell, 1987
Advisor: Rami Razouk

Guillermo Arango, 1988
Advisor: Peter Freeman

James Hester, 1988
Advisor: Dan Hirschberg

Matthew Jaffe, 1988
Advisor: Nancy Leveson

Debra Jusak, 1988
Advisor: Nancy Leveson

Raymond Klefstad, 1988
Advisor: Richard Taylor

Julio Leite, 1988
Advisor: Peter Freeman

Terry Mellon, 1988
Advisor: John King

Steven Reuman, 1988
Advisor: Donald Hoffman

Kadri Krause, 1988
Advisor: George Lueker

Craig Lee, 1988
Advisor: Lubomir Bic

Tim Shimmeal, 1989
Advisor: Nancy Leveson

Kurt Eiselt, 1989
Advisor: Richard Granger

Randolph Jones, 1989
Advisor: Patrick Langley

Donald Rose, 1989
Advisor: Patrick Langley

Michal Young, 1989
Advisor: Richard Taylor

Jonathan Gilbert, 1989
Advisor: Lubomir Bic

Christopher Pidgeon, 1989
Advisor: Peter Freeman

Bernd Nordhausen, 1989
Advisor: Patrick Langley

Bonnie Melhart, 1990
Advisor: Nancy Leveson

Rogers Hall, 1990
Advisor: Dennis Kibler

John Gennari, 1990
Advisor: Patrick Langley

Mariko Molodowitch, 1990
Advisor: George Lueker

Cheng Ng, 1990
Advisor: Daniel Hirschberg

Ira Baxter, 1990
Advisor: Peter Freeman

Jack Beusmans, 1990
Advisor: Donald Hoffman

David Aha, 1990
Advisor: Dennis Kibler

Etienne Wenger, 1990
Advisor: Dennis Kibler

Jose Ambros-Ingerson, 1990
Advisor: Richard Granger

John Roy, 1991
Advisor: Lubomir Bic

Joe Lis, 1991
Advisor: Daniel Gajski

Debra Lelewer, 1991
Advisor: Daniel Hirschberg

Yellamraju Srinivas, 1991
Advisor: Peter Freeman

Karen Ruhleder, 1991
Advisor: John King

Stephen Cha, 1991
Advisor: Nancy Leveson

Wang-Chan Wong, 1991
Advisor: Lubomir Bic

Phil Anton, 1991
Advisor: Rick Granger

Adam Porter, 1991
Advisor: Rick Selby

Wayne Iba, 1991
Advisor: Pat Langley

Jaimie Bae, 1991
Advisor: Tatsuya Suda

Jim Kipps, 1991
Advisor: Daniel Gajski

Ki C. Kim, 1992
Advisor: Alex Nicolau

Xiping Song, 1992
Advisor: Leon J. Osterweil

Allen Wu, 1992
Advisor: Daniel Gajski

Michael Greg James, 1992
Advisors: Richard Selby and Richard Taylor

Tracy Maples, 1992
Advisor: Tatsuya Suda

Elke Rundensteiner, 1992
Advisor: Lubomir Bic

Hari Asuri, 1993
Advisor: George Lueker

David Levine, 1993
Advisor: Richard N. Taylor

Hung Khei Huang, 1993
Advisor: Tatsuya Suda

David Ruby, 1993
Advisor: Dennis Kibler

Yousri El Fatah, 1993
Advisor: Paul O'Rorke

Dennis Troup, 1993
Advisor: Richard N. Taylor

R. Kent Madsen, 1993
Advisor: Richard Selby

Jim Wogulis, 1994
Advisor: Michael Pazzani

Sanjiv Narayan, 1994
Advisor: Daniel D. Gajski

Patrick Young, 1994
Advisor: Richard N. Taylor

Frank Vahid, 1994
Advisor: Daniel D. Gajski

Mats Heimdahl, 1994
Advisor: Nancy Leveson

Hai-Geng Wang, 1994
Advisor: Alexandru Nicolau

Douglas Schmidt, 1994
Advisors: Tatsuya Suda and Richard Selby

Lynn Stauffer, 1994
Advisor: Daniel Hirschberg

Viraphol Chaiyakul, 1994
Advisor: Daniel D. Gajski

Loganath Ramachandran, 1994
Advisor: Daniel D. Gajski

Robert L. Coultrip, 1995
Advisor: Richard Granger

Tedd Hadley, 1995
Advisor: Daniel D. Gajski

Karen Wieckert, 1995
Advisor: John King

Jonathan P. Allen, 1995
Advisor: Rob Kling

Marc K. Albert, 1995
Advisor: Donald Hoffman

David Schulenburg, 1995
Advisor: Michael Pazzani

Meng-Lai Monica Yin, 1995
Advisor: Douglas Blough

Chi-Kai Chien, 1995
Advisor: Isaac Scherson

Gie Gong, 1995
Advisor: Daniel Gajski

Pradip K. Jha, 1995
Advisor: Nikil Dutt

Raghu Subramanian, 1995
Advisor: Isaac Scherson

Umesh Krishnaswamy, 1995
Advisor: Isaac Scherson

Kamal Ali, 1995
Advisor: Michael Pazzani

Jon D. Reese, 1995
Advisor: John King

Rebecca E. Grinter, 1996
Advisor: Jonathan Grudin

John Tillquist, 1996
Advisor: Robert Kling

Patrick Murphy, 1996
Advisor: Dennis Kibler

Holly Hildreth, 1996
Advisor: Richard Taylor

Roger Ang, 1996
Advisor: Nikil Dutt

John Self, 1996
Advisor: Richard Taylor

Cliff Brunk, 1996
Advisor: Michael Pazzani

Veronica L. Reis, 1996
Advisor: Isaac Scherson

C. Michael Davis, 1996
Advisor: Richard Granger

Owen O'Malley, 1996
Advisor: Debra Richardson

Kenneth M. Anderson, 1997
Advisor: Richard N. Taylor

Steven S. Seiden, 1997
CoAdvisors: Sandra S. Irani &
Daniel Hirschberg

Roberta E. Lamb, 1997
Advisor: Rob Kling

Munehiro Fukuda, 1997
Advisor: Lubomir Bic

Vitus J. Leung, 1997
Advisor: Sandra S. Irani

Pedro M. Domingos, 1997
Advisor: Dennis F. Kibler

Piew Datta, 1997
Advisor: Dennis F. Kibler

Hadar Ziv, 1997
Advisor: Debra J. Richardson
Nancy Sue Eickelmann, 1997
Advisor: Debra J. Richardson

Erica Hsiao-Ping Juan, 1997
Advisor: Daniel D. Gajski

Dan Frost, 1997
Advisor: Rina Dechter

Chih-wen Hsueh, 1997
Advisor: Kwei-Jay Lin

Karl Kilborn, 1997
Advisor: Richard H. Granger

Min Xu, 1997
Advisor: Daniel Gajski

Preeti Panda, 1998
Advisor: Nikil Dutt

Joseph Hummel, 1998
Advisor: Alex Nicolau

Christopher J. Merz, 1998
Advisor: Michael J. Pazzani

Eddie Schwalb, 1998
Advisor: Rina Dechter

David Kolson, 1998
Advisor: Alex Nicolau

Steven Novack, 1998
Advisor: Alex Nicolau

Jonathan Kent Martin, 1998
Advisor: Daniel Hirschberg

Mark J. Rentmeesters, 1998
Advisor: W. Kevin Tsai

Gregory A. Bolcer, 1998
Advisor: Richard N. Taylor

Brett Vickers, 1998
Advisor: Tatsuya Suda